**文本, 信件

描述已自动生成**

Android**课程设计报告**

**（计算机科学与技术专业）**

**题目** 简单音乐播放器

|  |  |
| --- | --- |
| **学生姓名** | 董金伟、牛五扬、杨朋、叶龙 |
| **学 号** | 1912240124、1912240131、1912240127、1912240132 |
| **专业班级** | 计19本2班 |
| **指导教师** | 安强强 |

**信息工程学院**

**2021年12月12日**

# 目录

[目录 I](#_Toc90570278)

[1 绪论 1](#_Toc90570279)

[1.1 选题的背景 1](#_Toc90570280)

[1.2 本课题涉及内容的研究现状 1](#_Toc90570281)

[1.3 本系统要实现的基本目标和研究内容 2](#_Toc90570282)

[2 安卓平台介绍 3](#_Toc90570283)

[2.1 安卓平台介绍 3](#_Toc90570284)

[2.2 安卓结构具体分析 3](#_Toc90570285)

[2.2.1 应用程序 3](#_Toc90570286)

[2.2.2 应用程序的框架 4](#_Toc90570287)

[2.2.3 系统运行库 4](#_Toc90570288)

[2.2.4安卓运行库 4](#_Toc90570289)

[2.2.5Linux内核 5](#_Toc90570290)

[2.3 Android系统四大组件 5](#_Toc90570291)

[2.4 安卓开发环境搭建 6](#_Toc90570292)

[2.4.1 Java环境搭建 6](#_Toc90570293)

[2.4.2 安装Android Studio 7](#_Toc90570294)

[2.5 Android工程程序结构 7](#_Toc90570295)

[2.6 AndroidManif.xml文件 8](#_Toc90570296)

[2.7 安卓平台的优势 10](#_Toc90570297)

[2.7.1 开放性 10](#_Toc90570298)

[2.7.2 不受束缚 10](#_Toc90570299)

[2.7.3 丰富的硬件选择 10](#_Toc90570300)

[2.7.4 不受任何限制的开发商 10](#_Toc90570301)

[3 需求分析 11](#_Toc90570302)

[3.1 软件目标 11](#_Toc90570303)

[3.2 功能需求 11](#_Toc90570304)

[3.2.1 用例图及用例描述 11](#_Toc90570305)

[3.2.1音乐播放流程图 13](#_Toc90570306)

[3.2.2 软件规格说明 13](#_Toc90570307)

[3.2.3 运行环境 14](#_Toc90570308)

[4 项目实现 15](#_Toc90570309)

[4.1 设计思想 15](#_Toc90570310)

[4.2 系统功能结构图 16](#_Toc90570311)

[4.3主页面 16](#_Toc90570312)

[4.4 歌曲列表 17](#_Toc90570313)

[4.5 控制面板 18](#_Toc90570314)

[4.5.1 拖动条 18](#_Toc90570315)

[4.5.2 控制按钮 19](#_Toc90570316)

[4.5.3 音量控制 19](#_Toc90570317)

[4.6 功能实现 19](#_Toc90570318)

[data包 19](#_Toc90570319)

[model包 19](#_Toc90570320)

[MainActivity类 19](#_Toc90570321)

[PlayerService类 20](#_Toc90570322)

[5 测试 21](#_Toc90570323)

[6 结论 29](#_Toc90570324)

# 1 绪论

安卓（Android）是一种基于Linux内核（不包含GNU组件）的自由及开放源代码的操作系统。主要使用于移动设备，如智能手机和平板电脑，由美国Google公司和开放手机联盟领导及开发。Android操作系统最初由Andy Rubin开发，主要支持手机。2005年8月由Google收购注资。2007年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良Android系统。随后Google以Apache开源许可证的授权方式，发布了Android的源代码。第一部Android智能手机发布于2008年10月。Android逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机、智能手表等。2011年第一季度，Android在全球的市场份额首次超过塞班系统，跃居全球第一。 2013年的第四季度，Android平台手机的全球市场份额已经达到78.1%。2013年09月24日谷歌开发的操作系统Android在迎来了5岁生日，全世界采用这款系统的设备数量已经达到10亿台。

2021年5月19日凌晨消息，谷歌宣布Android 12正式到来，测试版现在即可供下载 。

## 选题的背景

随着计算机的广泛运用，手机市场的迅速发展，各种音频视频资源也在网上广为流传，这些资源看似平常，但已经渐渐成为人们生活中必不可少的一部分了。于是各种手机播放器也紧跟着发展起来，但是很多播放器一味追求外观花哨，功能庞大，对用户的手机造成了很多资源浪费，比如CPU，内存等的占用率过高，在用户需要多任务操作时，受到了不小的影响，带来了许多不便，而对于大多数普通用户，许多功能用不上,形同虚设。针对以上各种弊端,选择了开发多语种的音频视频播放器，将各种性能优化，继承播放器的常用功能，满足 般用户(如听歌，看电影)的需求，除了能播放常见格式的语音视频文件，高级功能:还能播放RMVB格式的视频文件。此外，还能支持中文、英文等语言界面。要研究各种市场上流行的手机播放器，了解它们各自的插件及编码方式，还有各种播放器播放的特别格式文件,分析各种编码的优缺点以及各种播放器本身存在的缺陷和特点，编写出功能实用，使用方便快捷的播放器。目前已经实现的功能有能播放常见音频文件的功能，如MP3，WAV,等，拥有播放菜单，能选择播放清单，具备一般播放器的功能，如快进，快退，音量调节等。

## 本课题涉及内容的研究现状

目前手机的音乐播放功能已经是大家比较关注的一个部分了,不少人在购买手机的时候都会关心手机的音乐播放的能力，这也足以看出目前大家对音乐播放功能的重视，所以一款性能良好的手机音乐播放器软件一定会受到欢迎。和传统的音乐播放器相比，手机播放音乐更加的容易携带，其他方面也都不输于传统音乐播放器所。一款专业的音乐播放软件是手机中必不可少的要素,几乎市场上所有的手机都拥有一个自带的音乐播放软件,通过软件可以对音乐欣赏中一些必要的元素进行人性化的设置，同时在操作的时候也能得到一定的便利，但是不能满足用户的所有要求，所以就出现了各种各样的手机音乐播放软件,来满足不同用户的不同要求。不同手机平台的手机，所支持的音乐播放软件也是各不相同，而现在最流行的手机平台就是google的android手机平台。Android播放器是近年来手机当中比较流行的音乐播放软件.通过结合自己所学的知识来设计和开发Android音乐播放器，可以深入学习Android多媒体框架结构和多媒体框架核心Open core,并通过对Open core的学习进一步 的认识和理解Android多媒体播放机制，而且还可以在开发软件实物的过程中，熟悉软件开发流程，掌握Android领域小应用开发的方法，学习新的计算机知识，丰富自己的知识面。Android 平台是完全开放的平台，智能手机里的各种应用软件也越来越完备，在Android平台设计和开发一款音乐播放器可以摆脱不够智能音乐播放器的因素，并且具有很广阔的前景。

## 本系统要实现的基本目标和研究内容

本文的主要内容是研究、分析和实现基于Android手机平台的音乐播放器系统，包括以下几个部分:

第1章简要介绍了Android手机平台的背景及发展情况。

第2章研究了基于Android平台的软件开发环境，主要包括Android平台的产品支持、Android平台软件的消息机制。

第3章详述进行基于Android手机平台的音乐播放器的开发的整个过程，包括对软件的需求分析、软件的结构设计。

第4章主要叙述了播放器的详细设计,包括主要模块功能的实现以和部分核心代码。

第5章是对音乐播放器的软件测试，主要是对其功能验证，确定是否符合软件规格要求，最后对全文进行一个总结。

# 安卓平台介绍

## 2.1 安卓平台介绍

Android是一.种基于Linux的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。Android操作系统最初由Andy Rubin开发，主要支持手机。2005 年8月由Google收购注资。2007 年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良Android系统。随后Google以Apache开源许可证的授权方式，发布了Android的源代码。Android 逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机等。Android的架构图如图1所示。

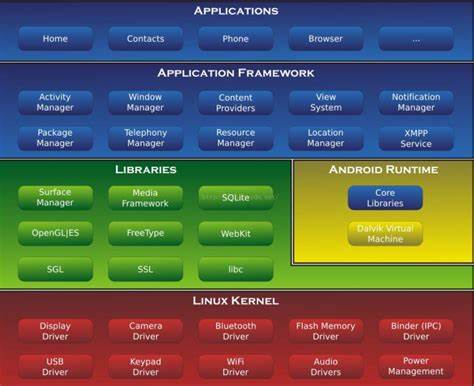


图 1 Android的架构图

Android的系统架构和其操作系统一样，采用了分层的架构。从架构图看，Android分为四个层分别是应用程序层、应用程序框架层、系统运行库层和Linux内核层。

## 安卓结构具体分析

### 2.2.1 应用程序

Android会同一系列核心应用程序包一起发布，该应用程序包包括客户端，SMS短.消息程序，日历，地图，浏览器，联系人管理程序等。所有的应用程序都是使用JAVA语言编写的。

### 2.2.2 应用程序的框架

开发人员也可以完全访问核心应用程序所使用的API框架。该应用程序的架构设计简化了组件的重用;任何一个应用程序都可以发布它的功能块并且任何其它的应用程序都可以使用其所发布的功能块(不过得遵循框架的安全性)。同样，该应用程序重用机制也使用户可以方便的替换程序组件。

隐藏在每个应用后面的是系列的服务和系统,其中包括;丰富而又可扩展的视图(Views)，可以用来构建应用程序，它包括列表(Lists)，网格(Grids)，文本框(Text boxes),按钮(Buttons)，甚至可嵌入的web浏览器。

内容提供器(Content Providers)使 得应用程序可以访问另一个应用程序的数据(如联系人数据库)，或者共享它们自己的数据。

资源管理器(Resource Manager)提供非代码资源的访问，如本地字符串，图形，和布局文件(Layout files )。

通知管理器(Notification Manager) 使得应用程序可以在状态栏中显示自定义的提.示信息。

活动管理器(Activity Manager)用来管理应用程序生命周期并提供常用的导航回退能。

### 系统运行库

Android包含 些C/C++库，这些库能被Android系统中不同的组件使用。它们通.过Android应用程序框架为开发者提供服务。以下是一些核心库:

(1)系统C库，一个从BSD继承来的标准C系统函数库Libc， 它是专门为基于Embedded Linux的设备定制的。

(2)媒体库,基 于PacketVideo OpenCORE；该库支持多种常用的音频、视频格式回放和录制，同时支持静态图像文件。编码格式包括MPEG4、H.264、MP3、AAC、AMR、JPG、PNG等。

(3) Surface Manager对显示子系统的管理，并且为多个应用程序提供了2D和3D图层的无缝融合。

(4) LibWebCore，个 最新的Web浏览器引擎用，支持Android浏览器和一个可嵌入Web视图。

### 2.2.4安卓运行库

Android包含一个核心库的集合,提供大部分在Java编程语言核心类库中可用的功能。每一个android应用程序是Dalvik虚拟机中的实例，运行在自己的进程中。Dalvik虚拟机被设计成，在一个设备可以高效地运行多个虚拟机。Dalvik虚拟机可执行文件格式是dex,dex格式是专为Dalvik设计的一一种压缩格式，适合内存和处理器速度有限的系统。

Dalvik虚拟机依赖于Linux内核提供基本功能，如线程和底层管理。

### 2.2.5Linux内核

Android基于Linux version 2.6 内核来提供系统的核心服务，例如:安全机制，内存管理，进程管理，网络堆栈和驱动模块。该内核还扮演着一个虚拟的中间层，该层介于硬件和其余的软件堆栈。

## 2.3 Android系统四大组件

在Android系统中，为我们提供了四大基本组件，每个组件是一个视图，但是有些是可见的，有些是不可见的视图，像Activity，ContentProvider 是可见的视图，Service和BroadcastReceiver是不可见的，它们只在系统的后台运行。下面分别介绍四大组件:

(1) Activity简介

在Android系统中Activity 提供可视化的用户界面，一个Android应用通常由多个Activity组成。多个Activity组成了Activity 栈(Stack)，当前活动的Activity处于栈顶。Activity有自己的生命周期，由Android系统来控制。

(2) Service简介

顾名思义Service就是运行在后台的一种服务程序，一般很少和用户交互，因此没有可视化界面定义一个Service 类比较简单，只要继承Service 类,实现其生命周期中的方法就可以了。一个定义好的Service必须在AndroidManifest.xml配置文件注册，通过<service>元素声明才能使用。Service有自己的生命周期，我们可以调用startService()启动一个 Service或者bindService()方法来绑定一个存在的Service。

(3) Broadcast Receiver简介

Broadcast Receiver顾名思义广播接收器，它和事件处理机制类似，只不过事件处理机制是程序组件级别的(例如，某个按钮的单击事件)，而广播事件处理机制是系统级别的。到目前为止我们可以使用Intent 来启动一个程序组件，我们还可以通过使用sendBroadcast()方法来发起一个系统级别的事件广播来传递消息。我们可以在你的应用程序中实现Broadcast Receiver 来监听和响应这些广播的Intent。 Intent(意图)主要是解决.Android应用的各项组件之间的通讯。

Intent负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述，Android则根据此Intent的描述，负责找到对应的组件，将Intent 传递给调用的组件，并完成组件的调用。

因此，Intent在这里起着一个媒体中介的作用，专门提供组件互相调用的相关信息，实现调用者与被调用者之间的解耦。

例如，在一个联系人维护的应用中，当我们在一个联系人列表屏幕(假设对应的Activity为ListActivity)上，点击某个联系人后，希望能够跳出此联系人的详细信息屏幕(假设对应的Activity 为detailActivity)。

为了实现这个目的，ListActivity 需要构造一个 Intent， 这个Intent 用于告诉系统，我们要做“查看”动作，此动作对应的查看对象是“某联系人”，然后调用startActivity(Intent intent)， 将构造的Intent 传入，系统会根据此Intent中的描述，到ManiFest中找到满足此Intent要求的Activity, 系统会调用找到的Activity, 即为detailActivity, 最终传入Intent，detail Activity则会根据此Intent 中的描述，执行相应的操作。

(4) Content Provider简介

Content Provider 用来保存和检索数据，并且使应用程序之间相互访问数据成为可能。它是跨应用程序共享数据的唯一方法。Android为常用的数据类型(如:音视频、图片和联系方式等)提供了大量的Content Provider.它们被定义在Android.provider包下面。通过这样定义好的Content Provider我们可以方便的进行数据操作。当然我们必须拥有适当的权限。我们也可以自己来定Content Provider 共享我们的数据，方便用户的访问。

## 2.4 安卓开发环境搭建

### 2.4.1 Java环境搭建

首先我们需要下载java开发工具包JDK，下载地址：https://www.oracle.com/

在下载页面中你需要选择接受许可，并根据自己的系统选择对应的版本，本文以 Window 64位系统为例：

下载后JDK的安装根据提示进行，还有安装JDK的时候也会安装JRE，一并安装就可以了。

安装JDK，安装过程中可以自定义安装目录等信息，例如我们选择安装目录为 C:\Program Files (x86)\Java\jdk11.0.1。

配置环境变量

1.右击“我的电脑”→“属性”→“高级系统设置”→“高级”→“环境变量”；

2.选择"高级"选项卡，点击"环境变量"；

3. 新建“JAVA\_HOME”系统变量（点击“系统变量”下方的“新建”按钮，填写变量名与变量值，点击“确定”）

4. 同上，新建“CLASSPATH”系统变量，变量值为“.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;”。

5. 双击“系统变量”下的“Path”变量进行编辑。（此时可以看到JAVA\_HOME已经存在于系统变量中），（有的电脑"Path"也写作“PATH”）

在"系统变量"中设置3项属性，JAVA\_HOME,PATH,CLASSPATH(大小写无所谓),若已存在则点击"编辑"，不存在则点击"新建"。

变量设置参数如下：

变量名：JAVA\_HOME

变量值：C:\Program Files (x86)\Java\jdk1.8.0\_91        // 要根据自己的实际路径配置

变量名：CLASSPATH

量值：.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;         //记得前面有个"."

变量名：Path

变量值：%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;

注意：如果使用1.5以上版本的JDK，不用设置CLASSPATH环境变量，也可以正常编译和运行Java程序。

通过控制台测试JDK是否安装成功

1、同时按键盘上“win”、“R”两个键打开运行，输入“cmd”确定打开控制台。

2、键入命令: java version、java、javac 几个命令，如果可以执行，说明环境变量配置成功；

### 2.4.2 安装Android Studio

Android Studio 是开发 Android 应用程序的官方 IDE，基于 Intellij IDEA。可以从官网Android Studio下载下载最新版本的 Android Studio。Windows 上安装 Android Studio 的新人，找到名为 androidstudiobundle135.17407740windows.exe 文件下载，并通过 Android Studio 向导指南运行。

安装：

确保在安装 Android Studio 之前，已经安装好 Java JDK。

一旦启动 Android Studio 安装，需要在 Android Studio 安装器中设置 JDK5 或以后版本路径。

Android SDK 初始化 JDK。检查创建应用程序所需的组件，选中 "Android Studio", "Android SDK", "Android 虚拟机"和"外观(Intel chip)"。需要指定本机上的 Android Studio 和 Android SDK 的路径。指定 Android 模拟器默认需要的 ram 空间为512M。最后，解压 SDK 软件包到本地机器，这将持续一段时间并占用2626M 的硬盘空间。完成上面的步骤，将看到结束按钮，并可以在欢迎界面中打开 Android Studio 项目，通过调用开始新的 Android Studio 项目来开始 Android 应用程序开发。在新的安装页面，要求填写应用程序名称，包名信息和项目路径。 输入应用程序名称之后，开始选择应用程序运行的环境参数需要指定最小 SDK。安装的下一步需要选择移动设备的活动，为应用程序指定默认布局。最后阶段打开开发工具来编写应用程序代码。

## 2.5 Android工程程序结构

Android工程基本结构目录包括: src(源代码)、gen(Android系统自动生成的常量)、Res(资源文件):主要存放程序界面的布局配置(.xml)文件和图片资源。AndroidMainfes.xml是四大组件的驱动配置文件，有它的存在，Android 程序的四大组件才可以正常的跳转。Android 资源的类型和布局。

图形用户界面, 文本, 应用程序

描述已自动生成

图21 项目结构图

## 2.6 AndroidManif.xml文件

Android系统的任何一个程序启动都需要AndroidManifest.xml来启动程序，任何一个新建的工程目录都会自动生成一个AndroidManifest.xml 文件，此配置文件是整个程序能够正常运行的核心，它里面包含了Android SDK的版本，程序运行的默认Activity,当程序中的任何一个组件触发事件后，系统都会自动的去AndroidManifest.xml中寻找标识来作出相应的操作。任何一个 AndroidManifest.Xml文件的格式都应为:

<?xml version="1.0" encoding="utf8"?>

<manifest >

<usespermissionsdk23

android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>

<application>

<service>

</service>

<activity>

<intentfilter>

<action />

<category />

</intentfilter>

</activity>

</application>

</manifest>

Android系统有一个编码的规定，支持的编码是"utf8"，当程序中出现乱码是，我们须将其编码格式转化为"utf8".在AndroidManifest.xml 中会在第一行 声明句: <?xmlversion="1.0" encoding= "utf8"?>。它是该系统支持的版本和编码。在第二行定义了我们需使用的架构，该架构来自xmIns:Androidttp://schemas.Android.com/apk/res/ Android。

Xmlns(XML namespace)是一 一个 命名空间，它就像一条河 的源头一样能让我们找到系统架构，每一个配置文件必须有这个xmlns，否则程序中的所使用的所有资源将无法找到。

我们定义Android:icon="@drawable/icon"Android:abel="@string/app\_ name"， Android: icon 是程序在手机中显示的图片，我们运用@drawable/icon表示了对res/drawable 目录下的icon.png 的引用。@string/app\_ name表示了对res/value目录下string.Xml中定义的app name 参数的引用。

因为系统首先要启动的Activity 是MainActivity， 所以文件如下:

<activity Android:name=".MainActivity">

<intentfilter>

<action Andro id:name="Android. intent. action.MAIN"/>

<category Android:name="Android. intent.category.LAUNCHER" />

</intentfilter>

</activity>

在intent filter选项中，有action和category等等属性，我们需要定义action的值为Android.intent.action.MAIN，category 的值为Android.intent.category.LAUNCHER，它们是系统的值，程序在解析到这些常用后就可以确定首先要启动的Activity。 当我们需要通过一个组件触发另一个组件时，我们将设定action和category的其它属性值，系统会根据我们设定的属性值去AndroidManifest.Xml 的application 中寻找标识，任何一个Activity的action 和category 是唯一， 所以程序在任何时候都能唯一启动一个Activity，这样，不同的Activity之间的跳转就实现了。

在AndroidManifest.Xml中有一个根结点<usessdk>，它是我们需要定义的ADT版本，例如我们将其值设定为Android:minSdk Version="10"，则表示此Android工程所用的版本为2.3，Android 还有1.1、1.5、1.6、2.0、2.1、2.2、2.3、3.0、4.0这几个版本。

如果我们需要对网络进行访问，比如打开一个网页,刚我还需要设置一个网络访问权限usespermission,将其的值设置为Android:name="Android.permission.INTERNET,我们就可以访问网络了。当然usespermission还有很多其它的权限，例如GPS定位权限浏览器数据存取权限com.Android.browser.permission.READ HISTORY\_BOOKMARKS等等。

当程序的任何一个组件要用到时，都必须在AndroidManifest.Xml文件中声明。四大组件要使用前都需要声明，其声明方式如下:

<service></service>

<provider></provider>

<activity></activity>

<receiver></receiver>

其中声明provider时还需要声明其授权authorities。每个组件里面还有很多的属性，会根据程序的具体需要定义不同的属性。过滤器intent filter有两种启动方式来启动界面。

(1) 显示方式

显示方式是在代码中的直接启动组件，启动时会声明Intent，在它的构造方法Intent(组件1,组件2)中直接指定一个组件的名字，然后调用方法startIntent(intent)以启动指定的组件。

(2) 隐式方式

隐式方式是通过过滤器来过滤完成的,启动的组件必须满足action和category的条件才能够启动该组件。这些属性是在AndroidManifest.Xml文件中声明的。两种启动方式都可以通过程序setAction(ACTION常量)和startActivity( intent)或配置文件两种方式来实现组件的跳转。

## 2.7 安卓平台的优势

### 2.7.1 开放性

在优势方面，Android 平台首先就是其开发性，开发的平台允许任何移动终端厂商加入到Android联盟中来。显著的开放性可以使其拥有更多的开发者，随着用户和应用的日益丰富，一个崭新的平台也将很快走向成熟。

开发性对于Android的发展而言，有利于积累人气,这里的人气包括消费者和厂商，而对于消费者来讲，最大的受益正是丰富的软件资源。开放的平台也会带来更大竞争，如此一来，消费者将可以用更低的价位购得心仪的手机。

### 2.7.2 不受束缚

在过去很长的一段时间，特别是在欧美地区，手机应用往往受到运营商制约，使用什么功能接入什么网络，几乎都受到运营商的控制。从去年iPhone上市，用户可以更加方便地连接网络，运营商的制约减少。随着EDGE、HSDPA这些2G至3G移动网络的逐步过渡和提升，手机随意接入网络已不是运营商口中的笑谈，当你可以通过手机IM软件方便地进行即时聊天时，再回想不久前天价的彩信和图铃下载业务，是不是像噩梦一样？互联网巨头Google推动的Android终端天生就有网络特色，将让用户离互联网更近。丰富的硬件

### 2.7.3 丰富的硬件选择

这一点还是与Android平台的开放性相关，由于Android的开放性，众多的厂商会推出千奇百怪，功能特色各具的多种产品。功能上的差异和特色，却不会影响到数据同步、甚至软件的兼容，好比你从诺基亚Symbian风格手机一下改用苹果iPhone，同时还可将Symbian中优秀的软件带到iPhone上使用、联系人等资料更是可以方便地转移，是不是非常方便呢？

### 2.7.4 不受任何限制的开发商

Android平台提供给第三方开发商一个十分宽泛、自由的环境，不会受到各种条条框框的阻扰，可想而知，会有多少新颖别致的软件会诞生。但也有其两面性，血腥、暴力、情色方面的程序和游戏如可控制正是留给Android难题之一。

# 3 需求分析

软件需求分析是软件生存期中重要的一步，也是决定性的一步。需求分析是软件定义时期的最后一个阶段，它的基本任务是准确地回答“系统必须做什么”这个问题，深入描述软件的功能和性能的需求。在这一节中主要就是围绕“Android开发平台的音乐播放器系统必须做什么”这个问题来分析说明的。

## 3.1 软件目标

本系统属于用户常用的音乐播放器。可以方便用户平时对音乐文件的操作。本系统应达到以下目标:

1. 软件采用人机交互的方式，界面美观友好，操作灵活、方便、富有个性化。
2. 能够对音乐文件进行有效的管理。
3. 系统应最大限度地实现易维护性和易操作性。

## 3.2 功能需求

### 3.2.1 用例图及用例描述

功能需求定义了开发人员必须实现的软件功能,使得用户能完成他们的任务，从而满足了用户的业务需求。用户需求文档描述了用户使用产品必须要完成的任务。针对音乐播放器分别对音乐播放功能进行详细的调研和分析，总结出如用户需求信息，其用例图如图所示。

文本, 白板

描述已自动生成

图3-1 播放器页面布局

音乐播放器的用例描述如下：

1. 用例名称：播放

参与者：用户

目标：使得用户可以播放在播放列表中选中的歌曲

前置条件：播放器正在运行

基本事件流: 1.用户单击“播放"按钮，

2.播放器将播放列表中的当前的歌曲

1. 用例名称：暂停

参与者：用户

目标：使得用户可以暂停正在播放的歌曲

前置条件：歌曲正在播放且末停止和暂停

基本事件流：1.用户单击“暂停”按钮

2.播放器将暂停当前的歌曲

1. 用例名称：上一首，下一首

参与者：用户

目标：使得用户可以听上一首或下一首歌曲

前置条件：歌曲正在播放或暂停

基本事件流：1.用户单击“上一首或下一首”按钮

2.播放器将播放上一首或下一首歌曲

1. 用例名称：播放列表

参与者：用户

目标：使得用户可以进入播放清单

前置条件：程序在运行

基本事件流：1.用户单击“清单”按钮

2.播放器播放歌曲

### 3.2.1音乐播放流程图

主页面列表

程序退出

点击歌曲名

播放音乐

点击退出

程序启动

是

是

否

否

## 3.2.2 软件规格说明

随着手机的功能的不断增加，音乐播放功能已经是时“下各大手机厂商手机方案中的标配了。那么我们要做出怎样的一款手机才能满足消费者的需求呢？仅仅能够播放MP3音乐文件是远远不够的。通过我们对市场的调研，总结出以下设计思路，这些思路应该是我们MP3播放器的亮点:绚丽的界面，强大的功能，简单的操作。根据以上的设计思路，我们做出了以下详细的定义：

1. 播放音乐文件。.
2. 播放文件时实现暂停，停止。
3. 播放文件时支持前一首或后一首音乐文件与当前音乐文件的切换。
4. 播放文件时支持进度条的显示。
5. 播放音乐文件时，支持背景动画。
6. 主播放界面支持播放方式的选择。
7. 主播放界面支持文件序号及文件名称的显示。
8. 支持文件列表。
9. 支持歌词文件同步。

## 3.2.3 运行环境

操作系统：Android手机基于Linux操作系统。

支持环境：Android 11 版本。

开发环境：Android Studio 2020.3.1.25windows。

# 4 项目实现

## 4.1 设计思想

随着智能手机的普及，用户需要各种各样的应用软件，音乐播放器肯定是必不可少的。随着计算机的广泛运用，手机市场的迅速发展，各种音频资源也在网上广为流传，这些资源卡是平常，但已经渐渐成为人们生活中必不可少的一部分了。于是各种手机播放器也紧跟着发展起来，但是很多播放器一味最求外观花俏，功能庞大，对用户的手机造成很多资源浪费，比如CPU，内存等的占用率过高，在用户需要多任务操作时，受到不小的影响，带来了许多不便，而对于大多数普通用户，许多功能用不上，形同虚设。针对以上各种弊端，选择了开发多语种的音频播放器，将各种性能优化，继承播放器的常用功能，满足一般用户听歌的需求。

## 4.2 系统功能结构图

图形用户界面, 图表, 树状图

描述已自动生成

## 4.3主页面

Android的每一个可视化界面，都有其的唯一的布局配置文件，该文件里面有各种布局方式，和各种资源文件如图像，文字，颜色的引用，程序在运行时，可以通过代码对各配置文件进行读取。这样就可以形成不同的可视化界面和炫丽的效果。播放器主界面是一个Activity， Android 工程在每个activity 启动的时候会首先执行Oncreate()方法，代码如下：

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

该方法主要执行界面的初始化操作。 进入主界面如图所示，主界面采用相对布局（RelativeLayout）。

电脑萤幕画面

描述已自动生成

图4-1 音乐播放器主页面

## 4.4 歌曲列表

歌曲显示以列表方式显示，显示时第一行是歌曲名，第二行是歌手名，并且使用线性布局。android:orientation="vertical"指定线性布局方向为垂直。android:width="match\_parent"指定线性布局的容器宽度为撑满整个屏幕宽度,android:height="wrap\_content"指定线性布局的容器高度为：根据容器内容高度大小来填充屏幕高度。最终使用ListView显示。

图图形用户界面, 文本

描述已自动生成4-2 音乐列表

## 4.5 控制面板

先使用线性布局LinearLayout，将拖动条，播放状态，控制按钮音量控制垂直摆放。然后使用相对布局RelativeLayout将其他按钮摆放进去。效果如图4-3所示

手机屏幕的截图

中度可信度描述已自动生成

图4-3 控制面板

### 4.5.1 拖动条

使用Android的基本UI控件SeekBar即可实现。之后使用相对布局RelativeLayout，设置TextView将音乐起始的时间与结束时间显示出来。其中时间的显出格式为“--:--”。

在拖动条下方插入一个TextView，显示歌曲的播放状态，默认显示未在播放。当点击歌曲后，更改为歌曲名，下方的TextView显示歌手名和专辑名称。

### 4.5.2 控制按钮

控制按钮使用相对布局RelativeLayout，每个按钮使用ImageButton控件来实现，每个按钮使用素材里的图片，设置layout参数时，首先使用layout\_margin="10dp"固定上一曲的位置，其余的可以layout\_toEndOf="@+id/imgBtnStop"的方式，按顺序依次摆放即可。

### 4.5.3 音量控制

使用相对布局，先使用ImageButton设置一个喇叭的图标，实现点击静音，再次点击恢复音量。图标后使用SeekBar拖动条来控制音量，之后再使用一个ImageButton设置一个控制播放模式的按钮。

## 4.6 功能实现

### data包

Music类中包含歌曲的名称，歌手，路径，时长等属性及相关获取方法，MusicList类提供了对列表的get方法。

### model包

实现了读取configuration.cfg中的主题，以及将主题保存到configuation.cfg中。

### MainActivity类

获取音乐文件，睡眠模式，播放模式，设置主题。

#### 获取音乐文件

使用data包中的两个类，将音乐文件的数据放到一个list中。使用MediaStore.Audio.AudioColumns分别获取音乐的TITLE，ARTIST，DURATION，DATA，ALBUM，ALBUM\_ID。然后遍历数据，并将Music类对象加载到ArrayList中。如果艺术家的值是"<unknown>"那么就将他设置为"未知艺术家"。

#### 睡眠模式

先用getLayoutInflater().inflate方法获取布局，用来初始化一个View类对象，通过View类的findViewById方法获取到组件对象，根据当前的睡眠状态来确定Switch的状态，最多60分钟

#### 播放模式

每个播放模式都有一个状态代码，不同代码表示不同的模式。使用switch-case来判段选择的模式，并执行相应的代码。如使用playmode = PlayerService.MODE\_SINGLE\_CYCLE;将播放模式设置为循环播放。

#### 设置主题

判断用户点击的主题是否和设定好的字符串匹配，如果匹配执行setBackgroundResource()方法，将相应的图片设置为背景。

### PlayerService类

提供了播放控制命令，播放状态。

#### 播放控制命令

使用不同的数字表示按下按钮的按钮，使用switch-case判断用户按下的哪一个按钮，然后执行对应的命令即可。

#### 播放状态

play()方法控制音乐的播放，播放前需要判断歌曲的状态，如果上一首歌曲时播放状态的话，需要将上一首歌曲暂停后才可以播放这首歌曲，播放之前需要载入新的音乐以及更新UI显示的内容。

pause()方法控制音乐的暂停，如果音乐正在播放的话，那就暂停音乐，否则啥都不做。

stop()方法控制停止播放音乐。

moveNext()和movePrevious()方法分别控制下一曲和上一曲，执行前，需要判断遍及，如果时列表底端或者列表顶端，就无法切换并给用户相应的提示。如果不处于列表的边界，执行相应的++或—操作，将歌曲的索引移动到下一个或上一个。

# 5 测试

1. 安装后打开软件，软件可以正常启动，并且可以获取到歌曲列表。

图形用户界面, 文本, 应用程序

描述已自动生成

图5-1 软件主页面

1. 点击列表中的歌曲，播放音乐。点击后可以正常播放。

图形用户界面, 文本, 应用程序

描述已自动生成

图5-2 播放测试

1. 进度条测试。拖动进度条后，歌曲可以正常播放，并且跳转到了相应的位置。

图形用户界面, 文本, 应用程序

描述已自动生成

图5-3 拖动进度条

1. 控制按钮测试。点击对应的按钮后，均可实现对应的操作。

图形用户界面, 文本, 应用程序

描述已自动生成

图5-4 下一曲测试

图形用户界面, 文本, 应用程序

描述已自动生成

图5-5 上一曲测试

图形用户界面, 文本, 应用程序

描述已自动生成

图5-6 暂停测试

图形用户界面, 文本, 应用程序

描述已自动生成

图5-7 停止测试

1. 音量滑动条测试。

图形用户界面, 文本, 应用程序

描述已自动生成

图5-8 音量条测试

1. 菜单各个功能测试。

文本

描述已自动生成

图5-9 主题测试

手机屏幕的截图

描述已自动生成

图5-10 睡眠测试

图片包含 文本

描述已自动生成

图5-11 播放模式测试

# 6 结论

通过对Android手机平台的音乐播放器软件的开发，使我对Android音乐播放器系统的整体设计有一个深入的了解，对整个流程也会有一个清晰的认识。开发Android音乐播放器，要抓住开发的核心部分，音乐播放器大体s由播放主界面、播放列表组成。这两个个模块完成，音乐播放器就能初具规模。而其它的功能都是在这两个模块的基础上去补充添加的，这是功能的确是必不可少的功能，否则就不具备音乐播放器的基本功能了。

音乐播放器软件实现了播放器的基本功能:播放，暂停、上/下一首、歌曲列表浏览等功能。运用的系统平台是现今最热门的Android 系统，是一个开源的基于Linux 系统的手机平台。是以Java语言编写实现了音乐播放器的系统编程。

本次程序设计，对播放器的系统结构框架做了精心的构思与设计，采用主要开发工具是Android Studio Java语言的支持，和Android SDK 11版本的全面结合，才使整个音乐播放器系统得以实现。在开发过程中我学到了一些经验：系统分析的好坏将决定着的系统开发成功与否，一份好分析 设计将是成功开发主要因素。我们在着手开发之前不要急于编程,先应有较长的时间去把分析做好，写出相关的开发文档等。然后再开始编写程序代码，这样做到每写一步伐码心底有数。

这个音乐播放器的开发已经结束，尽管是在整个开发过程中，走了不少弯路，犯了许多错误，但是吃一堑长一智，以后的工作中应该有了经验教训。总之，经过这次的实验Android系统技术知识的学习和对音乐播放器的构思与设计，使我对Android系统有了一定的了解，让我对Android系统上的开发流程和开发模式有了深入的了解，同时也对整个软件产品的实现流程有了一个深入的了解，这对我以后从事软件开发有很大的帮助。