**重庆科技学院**

**《移动应用开发》**

**大作业**

学 院:\_ 电气与信息工程学院\_ 专业班级: 计科2013-03

学生姓名: 郑敦坤 学 号: 2013443419

设计地点（单位）\_\_\_ I313\_\_\_\_\_\_ \_\_\_  
设计题目: 简单的音乐播放器

完成日期： 2016 年12月23日

指导教师评语: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

成绩（五级记分制）:\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_   
 指导教师（签字）:\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

# 摘 要

  随着生活水平的提高，娱乐已成为非常主流的话题，人们不仅需要通过音乐陶冶情操，而且越来越多的人倾向于使用音乐、视频等娱乐和放松自己，这大大促进了媒体软件的发展.本文旨在介绍android相关知识，了解音乐播放器功能的实现，掌握开发音乐播放器所需的相关知识，采用了面向对象软件工程方法，其开发主要包括应用程序界面设计和后台代码运行两个方面，实现了音乐播放器在安卓手机的应用，可以在很大程度上满足用户的需求.该系统主要具备：音乐播放控制、音乐文件控制、进度控制等功能模块。

**关键词：**音乐播放器；android；

# 1 需求分析

## 1.1 问题描述

随着Android时代的到来，应用越来越普遍，加之国民手机持有量更是爆炸式增长。移动音乐服务正是为这一形势而发展起来，本人提出一种简单的音乐播放器的应用，主要为了满足不同人群的音乐需要，并且提高优质的音乐服务，于是提出本音乐播发器的主要功能。

## 1.2 功能需求

1.实现手机SDCard音乐文件的播放、暂停、重头开始播放、停止等功能；

2.随着歌曲的播放，能在进度条上显示歌曲播放的进度；

3.能在进度条上用手指拖拽实现歌曲的快进或者后退到指定位置上播放；

4.能实现播放前一首歌曲或者下一首歌曲的切换；

5.能实现本地音乐列表显示在界面上，点击某一首歌曲，能自动播放；

6.能实现在后台播放（即界面关闭，音乐能继续播放）；

7.当有来电呼入或者呼出电话，音乐暂停。

# 2 详细设计

## 2.1模块设计

2.1.1在线播放音乐

提供音乐播放的基本功能，快进，暂停，播放，快退，同时进度滑块条需跟随滑动，用户也可以滑动进度条进行调整播放位置。

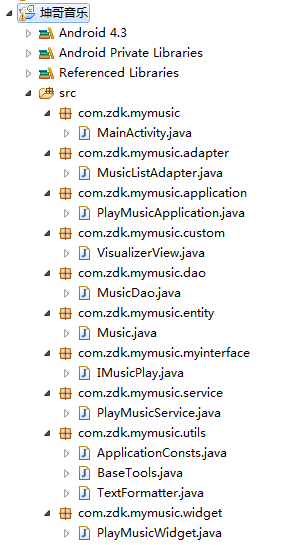
2.1.2扫描SD卡音乐

检测用户SD卡，如果没有，则弹出信息提示，并提示用户安装SD卡，如果检测有SD卡，则再遍历是否有可播文件，如果没有，则返回。如果有，则显示出全部音乐文件，并提供点击播放、暂停、上一首以及下一首功能。

2.1.3后台服务功能

当用户离开主界面时，音乐还可以继续播放，以及实现了通知栏模块，可以播放，下一首与服务同步，当用户拨打或接听电话时，音乐就会自动暂停，当挂断电话时，音乐又接着继续播放。

## 2.2类设计



## 2.3界面设计

2.3.1音乐播放界面



在该界面进行歌曲列表的查看以及播放等操作，主要代码如下

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);

setContentView(R.layout.*activity\_main*);

// 获取Application对象

Log.*d*("", "MainActivity.onCreate()");

app = (PlayMusicApplication) getApplication();

// 界面的初始化

initViews();

// 创建意图对象 显式启动

Intent intent = **new** Intent(**this**, PlayMusicService.**class**);

// 建立连接

conn = **new** InnerServiceConnection();

// 绑定service

bindService(intent, conn, *BIND\_AUTO\_CREATE*);

// 初始化广播接收者

initReceiver();

}

数据库的操作代码如下

**public** **void** update(**long** l){

ContentResolver cr = context.getContentResolver();

System.*out*

.println(MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*.toString());

Uri uri = Uri.*parse*(MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*.toString()+"/"+String.*valueOf*(l));

ContentValues values=**new** ContentValues();

values.put(Media.*TITLE*, "郑敦坤");

cr.update(uri, values, **null**, **null**);

Toast.*makeText*(context, "这是媒体文件，不允许修改歌曲信息哦", Toast.*LENGTH\_LONG*).show();

}

**public** **void** delete(**long** l){

ContentResolver cr = context.getContentResolver();

System.*out*

.println(MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*.toString());

Uri uri = Uri.*parse*(MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*.toString()+"/"+String.*valueOf*(l));

cr.delete(uri, **null**, **null**);

}

**public** ArrayList<Music> getMusicList() {

ArrayList<Music> musics = **null**;

Music music;

ContentResolver cr;

String[] projection;

Cursor cursor;

Uri uri;

uri = Media.*EXTERNAL\_CONTENT\_URI*;

projection = **new** String[] { Media.*\_ID*, Media.*DATA*, Media.*TITLE*,

Media.*DURATION*, Media.*ARTIST*, Media.*ALBUM*,Media.*ALBUM\_ID*};

cr = context.getContentResolver();

cursor = cr.query(uri, projection, **null**, **null**, **null**);

**if** (cursor != **null** && cursor.getCount() > 0) {

musics = **new** ArrayList<Music>();

**while** (cursor.moveToNext()) {

**long** long1 = cursor.getLong(cursor

.getColumnIndex(Media.*DURATION*));

**if** (long1 > 30000) {

music = **new** Music();

music.setId(cursor.getLong(cursor.getColumnIndex(Media.*\_ID*)));

music.setDuration(cursor.getLong(cursor

.getColumnIndex(Media.*DURATION*)));

Log.*i*("DURATION", long1 + "long1");

music.setData(cursor.getString(cursor

.getColumnIndex(Media.*DATA*)));

music.setTitle(cursor.getString(cursor

.getColumnIndex(Media.*TITLE*)));

music.setArtist(cursor.getString(cursor

.getColumnIndex(Media.*ARTIST*)));

music.setAlbum(cursor.getString(cursor

.getColumnIndex(Media.*ALBUM*)));

music.setAlbumId(cursor.getLong(cursor.getColumnIndex(Media.*ALBUM\_ID*)));

musics.add(music);

music = **null**;

}

}

cursor.close();

}

// 以下是Log输出获取数据的结果

Log.*d*("", "MusicDao.getMusicList(), musics = " + musics);

**if** (musics != **null**) {

**for** (Music m : musics) {

Log.*d*("", "MusicDao.getMusicList(), music = " + m.toString());

}

}

**return** musics;

}

后台服务代码主要如下

**public** **class** PlayMusicBinder **extends** Binder **implements** IMusicPlay {

@Override

**public** **void** play() {

Log.*i*("", "PlayMusicBinder---play()");

playMusic();

}

@Override

**public** **void** stop() {

Log.*i*("", "PlayMusicBinder---stop()");

pauseMusic();

}

@Override

**public** **int** getProgress() {

**return** player.getCurrentPosition();

}

@Override

**public** **void** playSeekTo(**int** progress) {

Log.*i*("", "playSeekTo---()");

currentPosition = (**int**) (musics.get(app.getCurrentMusicIndex())

.getDuration() \* progress / 100);

playMusic();

}

@Override

**public** **void** playNext() {

nextMusic();

}

@Override

**public** **void** playPrevious() {

previousMusic();

}

@Override

**public** **int** getAudioSessionId() {

**return** player.getAudioSessionId();

}

}

通知栏代码如下：

**public** **void** sendNotification() {

Music mp3InfoSend = **this**.musics.get(app.getCurrentMusicIndex());

**final** NotificationCompat.Builder builder = **new** NotificationCompat.Builder(

**this**);

// 创建一个远程的视图

RemoteViews views = **new** RemoteViews(getPackageName(),

R.layout.*custom\_layout*);

Bitmap albumBitmap = MusicDao.*getArtwork*(**this**, mp3InfoSend.getId(),

mp3InfoSend.getAlbumId(), **false**, **false**);

**if** (albumBitmap != **null**) {

views.setImageViewBitmap(R.id.*custom\_song\_icon*, albumBitmap);

} **else** {

views.setImageViewResource(R.id.*custom\_song\_icon*,

R.drawable.*app\_logo2*);

}

views.setTextViewText(R.id.*tv\_custom\_song\_name*, mp3InfoSend.getTitle());

views.setTextViewText(R.id.*tv\_custom\_song\_singer*,

mp3InfoSend.getArtist());

// 如果版本号低于（3。0），那么不显示按钮

**if** (BaseTools.*getSystemVersion*() <= 9) {

views.setViewVisibility(R.id.*ll\_custom\_button*, View.*GONE*);

} **else** {

views.setViewVisibility(R.id.*ll\_custom\_button*, View.*VISIBLE*);

**if** (isPlaying) {

views.setImageViewResource(R.id.*btn\_custom\_play*,

R.drawable.*pause2*);

} **else** {

views.setImageViewResource(R.id.*btn\_custom\_play*,

R.drawable.*play2*);

}

}

Intent intent = **new** Intent(**this**, MainActivity.**class**);

PendingIntent pi = PendingIntent.*getActivity*(**this**, 0, intent,

PendingIntent.*FLAG\_UPDATE\_CURRENT*);

// 点击的事件处理

Intent buttonIntent = **new** Intent(*ACTION\_BUTTON*);

/\* 播放/暂停 按钮 \*/

buttonIntent.putExtra(*INTENT\_BUTTONID\_TAG*, *BUTTON\_PLAY\_ID*);

PendingIntent intent\_play = PendingIntent.*getBroadcast*(**this**, 1,

buttonIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT*);

views.setOnClickPendingIntent(R.id.*btn\_custom\_play*, intent\_play);

/\* 下一首 按钮 \*/

buttonIntent.putExtra(*INTENT\_BUTTONID\_TAG*, *BUTTON\_NEXT\_ID*);

PendingIntent intent\_next = PendingIntent.*getBroadcast*(**this**, 2,

buttonIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT*);

views.setOnClickPendingIntent(R.id.*btn\_custom\_next*, intent\_next);

/\* 退出按钮 \*/

buttonIntent.putExtra(*INTENT\_BUTTONID\_TAG*, *BUTTON\_EXIT\_ID*);

PendingIntent intent\_exit = PendingIntent.*getBroadcast*(**this**, 3,

buttonIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT*);

views.setOnClickPendingIntent(R.id.*btn\_custom\_exit*, intent\_exit);

builder.setContent(views).setContentIntent(pi)

.setWhen(System.*currentTimeMillis*())// 通知产生的时间，会在通知信息里显示

.setTicker("坤哥音乐").setPriority(Notification.*PRIORITY\_MAX*)// 设置该通知优先级

.setOngoing(**true**).setSmallIcon(R.drawable.*app\_logo2*);

NotificationManager nm = (NotificationManager) getSystemService(Context.*NOTIFICATION\_SERVICE*);

nm.notify(NID\_1, builder.build());

}

总结

经过这一学期的学习，这次我负责简单音乐播放器的开发，在开发过程中遇到了许多的问题，比如经常遇到空指针异常，后台播放与通知栏不同步等问题，但经过我不懈努力，终于解决了这些问题，经过这次的大作业的练习，我深深的感觉到实践的重要性，唯有实践才能体会到更多在学习理论知识中学不到的知识。俗话说：“实践是检验真理的标准。”

涉及到的知识点：综合运用内容提供者、广播接受者、服务等知识实现音乐播放器。利用ListView以及适配器实现了显示全部的SD卡音乐的信息。

不足：没有实现菜单按钮功能。页面布局弄得不看，代码优化也弄得不好。