移动终端开发技术
课程设计报告

题　　目： 音乐播放器

指导教师：曹洪新

班　　级：2017 移动通信技术一班

姓名学号：杨中月　201725150114

燕琪　201725150109

葛猜玉　201725150130

电子信息与工程学院

2019 年　1 月 2　日

目　　录

1. 序言

1.1 选题背景

当今社会的生活节奏越来越快，人们对手机的要求也越来越高，由于手机市场发展迅速，使得手机操作系统也出现了不同各类，现在的市场上主要有三个手机操作系统，Windows mobile，symbian，以及谷歌的Android 操作系统,其中占有开放源代码优势的Android 系统有最大的发展前景。那么能否在手机上拥有自己编写的个性音乐播放器呢？能的，谷歌 Android 系统就能做到。本文的音乐播放器就是基于谷歌 Android 手机平台的播放器。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。播放模式也比较完善，有单曲，顺序，循环，随机播放等模式

1.2 选题目的及意义

本设计实现的主要功能是播放 MP3 格式的音乐文件，并且能够控制音乐播放、暂停、停止、上一曲和下一曲，音量调节，播放列表和歌曲文件的操作等多种控制功能，界面简明，操作简单，锻炼

程序编写能力，熟悉安卓的设计流程。

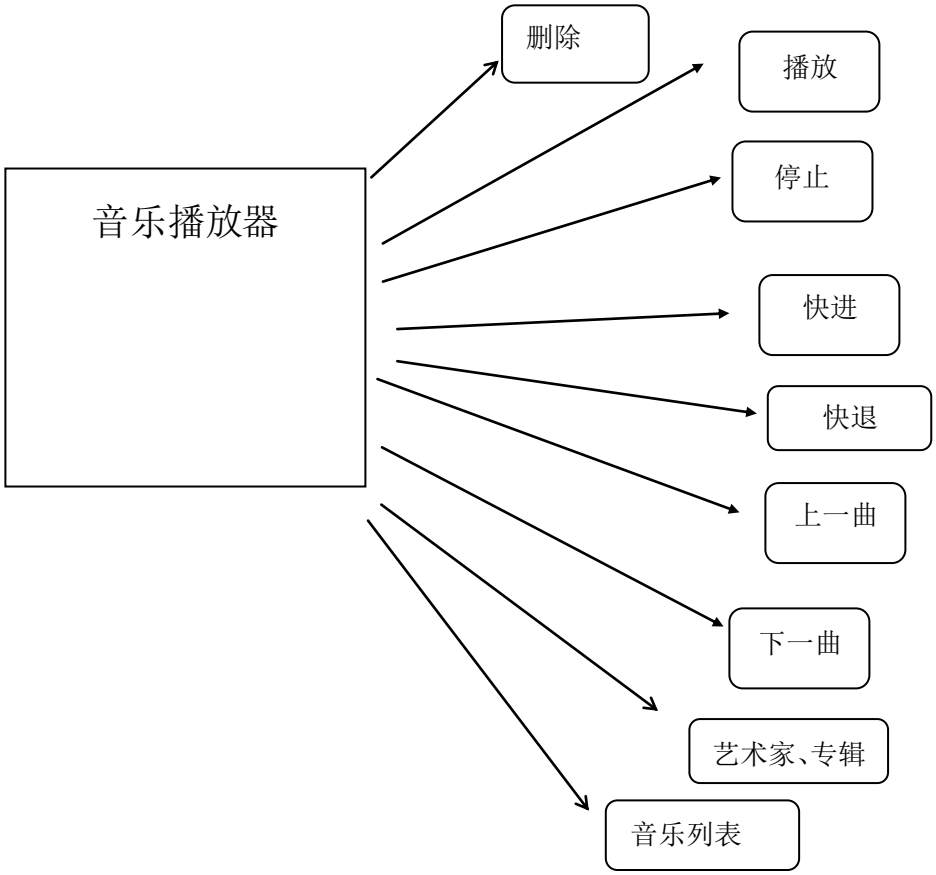
本设计是基于 android 手机平台的音乐播放器，使手机拥有个性的多媒体播放器，使手机更具娱乐性。让我们更加熟练 android 的技术和它的特点。

2. 项目需求分析

2.1 项目功能需求

2.1.1 播放器的基本控制需求

根据设计的目标，可获得本设计的基本需求，如下图所示：



播放器基本控制图

播放：进行音乐播放

停止：暂停正在播放的音乐

快进：可以拖动进度进行快进

快退：拖动音乐进度条退回

上一曲：切换到上一首歌曲

下一曲：切换到下一首歌曲

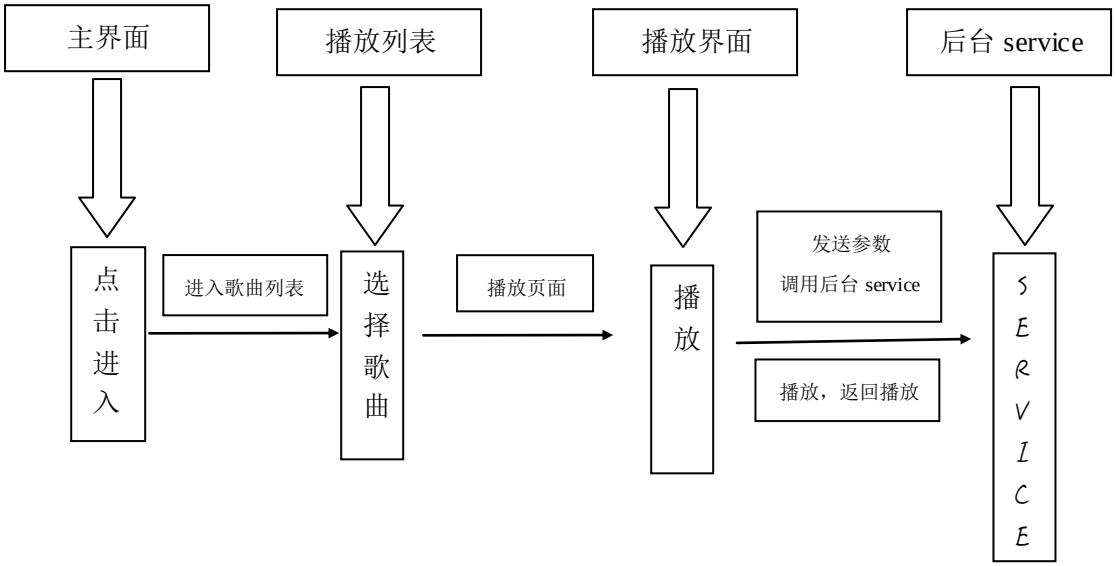
音乐列表：显示所存储的本地音乐

艺术家：根据不同歌唱家分类

专辑：每个歌曲所属的不同专辑类型

删除：删除所选中的音乐

2.1.2 功能(顺序图)操作



2.2 项目性能需求

此音乐播放器需要有的功能为 音乐播放、暂停、快进、快退、下/上一曲、音乐显示列表。

且此 APP 要适合安卓平台运行。界面应适合大多数用户的审美，如简介、美观、大方。

2.3 其他需求

运行环境需求：

操作系统：windows xp 或以上

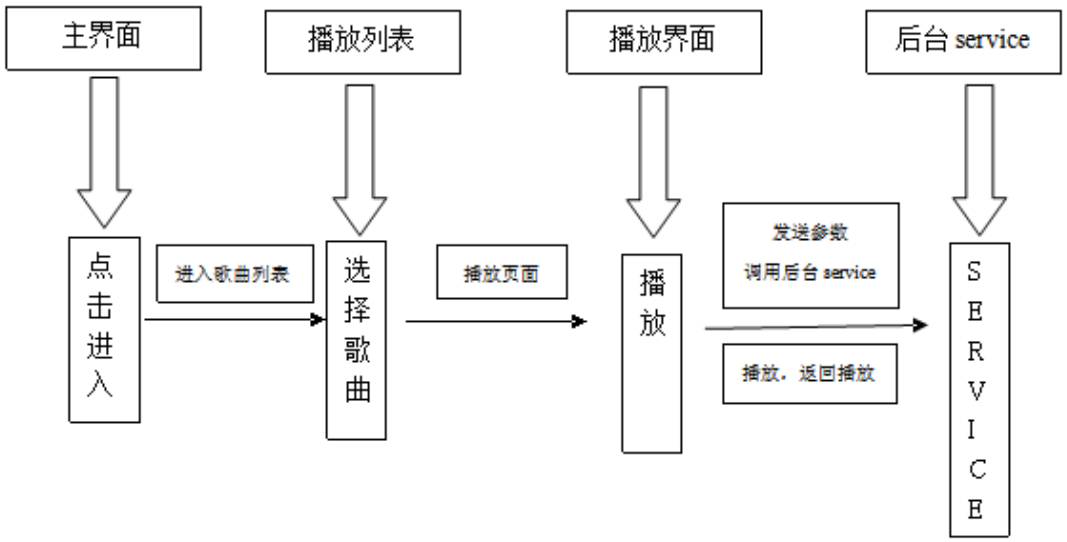
开发环境：含 adt 插件的 eclipse 以及 Android Studio

3．项目概要设计

3.1 系统功能结构设计

音乐播放器的基本功能：歌手名称、专辑、音乐播放、暂停、快进、快退、上一首、下一首。

3.2 系统整体架构设计


主界面
播放列表
播放界面
后台 service
点击进入
进入歌曲列表
选择歌曲
播放页面
播放
发送参数
调用后台 service
播放，返回播放
SERVICE

4．项目详细设计

4.1 音乐列表

4.1.1 功能描述

1. 在本地存储里面可以寻找到已经下载完成的音乐。

2. 显示存储中的音乐。

3. 具有播放、暂停歌曲功能。此功能通过点击播放、暂停按钮进行实现。

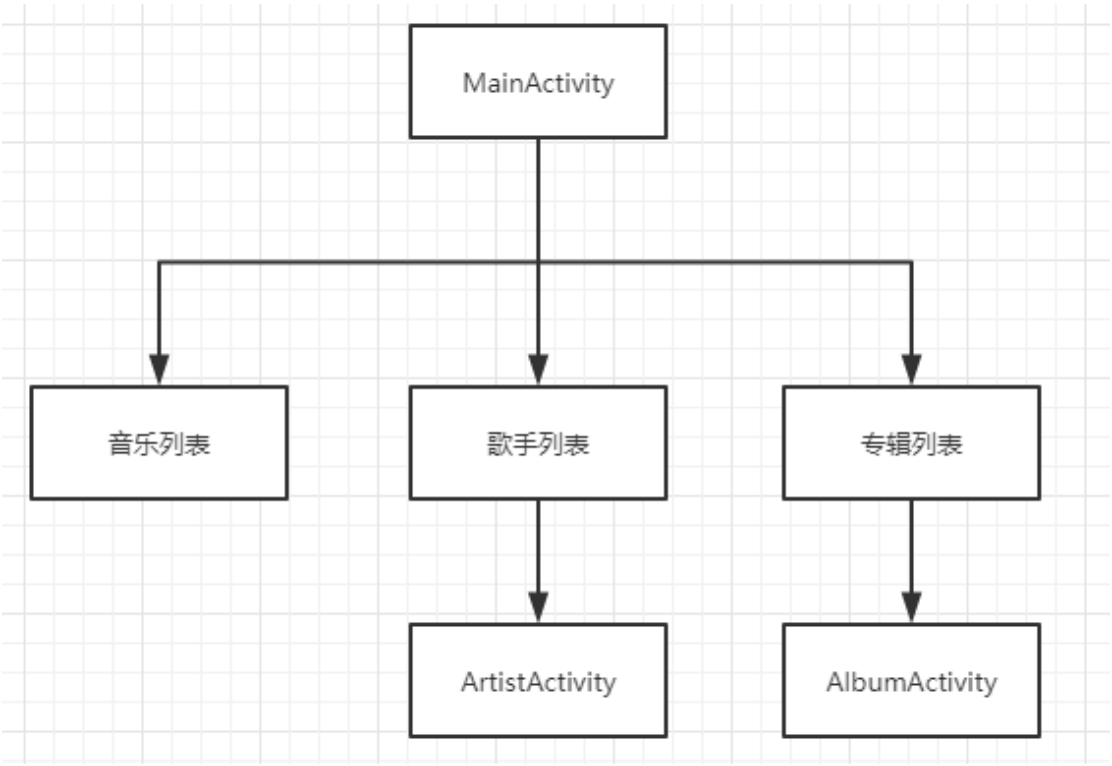
4. 具有切换上一首、下一首功能。此功能通过点击切换音乐按钮实现。

5. 具有快进、快退功能。此功能通过拖动音乐进度条实现。

4.1.2 性能描述

该程序反应时间快，运行精度高，失误较少。

4.1.3 算法描述/逻辑流程



4.1.4 关键函数设计

```
/*播放选中的音乐*/
private void playMusic(int position){
    Intent intent = new Intent(AlbumActivity.this,MusicActivity.class);
    intent.putExtra("_ids", _ids);
    intent.putExtra("_titles", _titles);
    intent.putExtra("position", position);
    startActivity(intent);
    finish();
}

/*从列表中删除选中的音乐*/
private void deleteMusic(int position){

this.getContentResolver().delete(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
            MediaStore.Audio.Media._ID + "=" + _ids[position],
            null);
```

```
}

/*从 sdcard 中删除选中的音乐*/
private void deleteMusicFile(int position){
   File file = new File(_path[pos]);
   file.delete();
}
class ListItemClickListener implements OnItemClickListener{

   public void onItemClick(AdapterView<?> arg0, View view, int position,
long id) {
       // TODO Auto-generated method stub
       playMusic(position);
   }

}
       //设置艺术家姓名
       TextView artist =
(TextView)convertView.findViewById(R.id.artist);
       artist.setText(artists[position]);

       //设置列表项图标
       ImageView Artistsitem =
(ImageView)convertView.findViewById(R.id.Artistsitem);
       Artistsitem.setImageResource(R.drawable.artist);
       return convertView;
   }

}

public class MusicActivity extends Activity {

   private int[] _ids;
   private int position;
```

```
    private String _titles[] = null;
    private Uri uri;
    private ImageButton playBtn = null;//播放、暂停
    //private Button stopBtn = null;//停止
    private ImageButton latestBtn = null;//上一首
    private ImageButton nextBtn = null;//下一首
    private ImageButton forwardBtn = null;//快进
    private ImageButton rewindBtn = null;//快退
    private TextView lrcText = null;//歌词文本
    private TextView playtime = null;//已播放时间
    private TextView durationTime = null;//歌曲时间
    private SeekBar seekbar = null;//歌曲进度
    private SeekBar soundBar = null;//音量调节
    private Handler handler = null;//用于进度条
    private Handler fHandler = null;//用于快进
    private int currentPosition;//当前播放位置
    private int duration;
    private DBHelper dbHelper = null;
    private TextView name = null;
    private GestureDetector gestureDetector;
protected void onCreate(Bundle savedInstanceState) {
    //调用父类 oncreate 方法，并创建界面
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main1);

    //从 intent 中取得传递来的数据
    Intent intent = this.getIntent();
    Bundle bundle = intent.getExtras();
    _ids = bundle.getIntArray("_ids");
    position = bundle.getInt("position");
    _titles = bundle.getStringArray("_titles");

    //初始化各个控件
    lrcText = (TextView)findViewById(R.id.lrc);
```

```
    name = (TextView)findViewById(R.id.name);
    playtime = (TextView)findViewById(R.id.playtime);//已经播放的时间
    durationTime = (TextView)findViewById(R.id.duration);
    //获取媒体音量
    mAudioManager = (AudioManager)
getSystemService(Context.AUDIO_SERVICE);
    maxVolume =
mAudioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC);//获得最
大音量
    currentVolume =
mAudioManager.getStreamVolume(AudioManager.STREAM_MUSIC);//获得当前
音量
    //TODO 手势识别
    gestureDetector=new GestureDetector(new
ChangeGestureDetector(this));

    //设置播放按钮
    playBtn = (ImageButton)findViewById(R.id.playBtn);
    playBtn.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v) {
            //flag 表示现在歌曲的播放状态
            switch (flag) {
                case STATE_PLAY:
                    pause();
                    break;

                case STATE_PAUSE:
                    play();
                    break;
            }
        }
    });
```

```
/**
 * 音乐播放
 */
private void play(){
    //开始播放，将 flag 的状态改变
    flag = STATE_PLAY;
    //将播放按键的背景图片换为暂停图片
    playBtn.setBackgroundResource(R.drawable.pause_selecor);
    //启动歌曲播放 Service
    Intent intent = new Intent(MusicActivity.this,MusicService.class);
    intent.putExtra("op",MUSIC_PLAY);
    startService(intent);
}

/**
 * 音乐暂停
 */
private void pause(){
    flag = STATE_PAUSE;
    playBtn.setBackgroundResource(R.drawable.play_selecor);
    Intent intent = new Intent(MusicActivity.this,MusicService.class);
    intent.putExtra("op",MUSIC_PAUSE);
    startService(intent);
}

/**
 * 音乐停止
 */
private void stop(){
    unregisterReceiver(musicReceiver);
    Intent intent = new Intent(MusicActivity.this,MusicService.class);
    intent.putExtra("op", MUSIC_STOP);
    startService(intent);
}
```

```
/**
 * 用户拖动进度条
 */
private void seekbar_change(int progress){
    Intent intent = new Intent(MusicActivity.this,MusicService.class);
    intent.putExtra("op", PROGRESS_CHANGE);
    intent.putExtra("progress", progress);
    startService(intent);
}

/**
 * 快退
 */
private void rewind(){
    Intent intent = new Intent(MusicActivity.this,MusicService.class);
    intent.putExtra("op", MUSIC_REWIND);
    startService(intent);
}

/**
 * 快进
 */
private void forward(){
    Intent intent = new Intent(MusicActivity.this,MusicService.class);
    intent.putExtra("op", MUSIC_FORWARD);
    startService(intent);
}

/**
 * 上一首
 */

public void latestOne(){
```

```
    if (position==0){
        position = _ids.length-1;
    } else if (position>0){
        position--;
    }
    stop();
    setup();
    play();
}

/**
 * 下一首
 */
public void nextOne(){
    if (_ids.length==1){
        position = position;
        Intent intent = new
Intent(MusicActivity.this,MusicService.class);
        intent.putExtra("length", 1);
        startService(intent);
        play();
        return;

    } else if (position == _ids.length-1){
        position = 0;
    } else if (position < _ids.length-1){
        position++;
    }
    stop();
    setup();
    play();
}
/**
     * 开始、暂停、停止
```

```
        */
        int op = intent.getIntExtra("op", -1);
        if (op!=-1){
            switch (op) {
                case MUSIC_PLAY://播放
                    if(!mp.isPlaying()){
                        play();
                    }
                    break;
                case MUSIC_PAUSE://暂停
                    if (mp.isPlaying()){
                        pause();
                    }
                    break;
                case MUSIC_STOP://停止
                    stop();
                    break;
                case PROGRESS_CHANGE://改变歌曲进度
                    currentTime = intent.getExtras().getInt("progress");
                    mp.seekTo(currentTime);

                    break;
                case MUSIC_REWIND://快退
                    rewind();
                    break;
                case MUSIC_FORWARD://快进
                    forward();
                    break;
            }
        }

    }
    /**
        * 数据库操作
```

```
     * @param pos
     */
    private void DBOperate(int pos){
        dbHelper = new DBHelper(this, "music.db", null, 2);
        Cursor c = dbHelper.query(pos);
        Date currentTime = new Date();
        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
        String dateString = formatter.format(currentTime);
//      if (c==null||c.getCount()==0){//如果查询结果为空
//          ContentValues values = new ContentValues();
//          values.put("music_id", pos);
//          values.put("clicks", 1);
//          values.put("latest", dateString);
//          dbHelper.insert(values);
//      } else{
//          c.moveToNext();
//          int clicks = c.getInt(2);
//          clicks++;
//          ContentValues values = new ContentValues();
//          values.put("clicks", clicks);
//          values.put("latest", dateString);
//          dbHelper.update(values, pos);
//      }
//      if (c!=null){
//          c.close();
//          c = null;
//      }
//      if (dbHelper!=null){
//          dbHelper.close();
//          dbHelper = null;
//      }
    }
```

4.2 音乐播放控制

4.2.1 功能描述

说明该程序应具有的功能，可采用 IPO 图（即输入一处理一输出图）的形式。

播放：

停止：

快进：

快退：

上一曲：

下一曲：

4.2.2 性能描述

说明对该模块的全部性能要求，包括对精度、灵活性和时间特性的要求。

4.2.3 算法描述/逻辑流程

```
durationTime.setText(toTime(duration));


    } else if (action.equals(MUSIC_NEXT)){
        //下一首
        nextOne();
    } else if (action.equals(MUSIC_UPDATE)){
        //刷新
        position = intent.getExtras().getInt("position");
        //refreshView();
```

```
            //name.setText(_titles[position]);
            setup();
        }
    }
};
```

进行示例：播放音乐
进行者：用户进行
目标：选中音乐列表中的音乐进行正常播放
有无条件：播放器正在运行且需用户主动点击
运行效果：用户点击某音乐后此播放器播放选中歌曲

进行示例：暂停
进行者：用户进行
目标：暂停所播放的音乐
有无条件：用户主动点击暂停按钮
运行效果：用户点击后音乐暂停播放

进行示例：快进、快退
进行者：用户进行
目标:音乐进度跳转
有无条件：需拖动进度条
运行效果：音乐播放用户点击到的片段

进行示例：上一首、下一首
进行者：用户
目标：切换到上/下一首音乐
有无条件：点击切换按钮
运行效果：音乐切换

5．实训总结

通过对本次安卓作业——音乐播放器软件的开发，让我们更好的学习了安卓这门课程，加深了对于安卓项目的了解，也通过做安卓项目明白软件应符合大众需求，而不是少数人的需求。还有，功能应该有侧重点，如这次音乐播放器软件的功能重点就是：播放、暂停、快进、快退、上一首、下一首。所以真个软件的功能都是围绕着这几点进行的，里面还有一些其余小的功能，如根据不同艺术家进行歌曲分类，根据不同专辑分类，声音大小的调控等。

通过这一次的安卓作业项目，也让我们明白了安卓开发的难度与深度，体会到了老师的艰辛与不易。以后我们一定会更加努力的学习。