

# 1 引言

## 1.1 选题背景

随着智能手机的大众化，越来越多的人应用智能手机。相对的对手机软件的需求也就越来越多。当今时代手机不再只是用来打电话、发短信的工具，反而对手机的娱乐功能要求越来越高<sup>[1]</sup>。手机的一个重要的功能就是听音乐，一款好的手机音乐播放器可以更好的体现智能手机的强大。目前中国音乐播放器行业已经具备了相当的规模，并在国际市场上已有一定的竞争力。苹果手机作为当今中国智能手机市场上不可或缺的一个品牌，基于 iOS 的手机软件需求量也就相应的越来越强。市场上的音乐播放器品牌繁多，定位不一，基本满足了各层次消费者的需求，例如：酷我音乐、酷狗音乐、百度音乐、QQ 音乐等等这些大家所熟知的音乐播放器已经在苹果软件中占据了一定的地位。智能手机的应用使现代人的生活更加的方便。

伴随着苹果手机进入中国市场，相应的 iOS 技术逐渐流行起来。而对于开发苹果应用的语言——Object-C，也成为当今社会最流行的开发语言，并且保持着强劲的上升趋势，隐隐有超过 C 语言的现象<sup>[2]</sup>。对于其他的开发语言，例如 java、javascript，Object-C 以其强大的优势遥遥领先于这些其他的编程语言。本人根据当前的形式，决定引用这种这种技术来开发手机应用。

## 1.2 设计目的及意义

听音乐已经成为当今社会，人们必不可少的一项娱乐活动。根据调查所得信息，本人发现最近智能手机用户对较大的音乐播放器应用的喜爱度明显下滑，而相对的那些小巧的音乐播放器反而被越来越多的用户下载使用。苹果系统的软件具有的特点就是美观、简洁、易操作。根据这些信息，设计一款小巧的基于 iOS 的音乐播放器会很容易得到用户的青睐。同时本人从去年开始接触 iOS，在学习了几个月后，对 iOS 开发有了一些理解，所以对于这次毕业设计，我就选择了基于 iOS 系统开发一款产品。希望通过这次毕业设计来进一步巩固自己的 iOS 开发水平。使自己在以后的社会竞争中更加具有优势。

## 1.3 设计思想和目标

设计思想：

自己的这款音乐播放器设计的过程是按照一般的应用开发流程进行的。自己首先调查了当今社会，人们对音乐播放器的需求，从中得出现阶段手机用户需要的是小巧、方便的音乐播放器。从而决定自己开发一款小巧的音乐播放器。在有了用户的需求后，本人开始开发属于自己的音乐播放器。在真正设计音乐播放器的过程中，本人先决定了音乐播放器的整体框架结构，应用的图片出处等等这些开发必需的部分。于是自己借鉴酷狗音乐的框架，开发一款精简的酷狗音乐播放器。在决定了整体框架后，就是真正的开发了。自己将酷狗音乐中与播放无关的部分删除，只保留他的播放功能，歌词显示功能音乐下载功能等等。下载功能的实现本人决定针对这款音乐播放器设计出一个可供下载的网站以及服务器。

设计目标：虽说这款应用自己不打算上线，但自己依旧打算使其完美。拥有播放器的一般功能，可以根据不同的歌曲显示不同的歌词，进度条根据时间显示播放进度，根据不同的音乐信息进行选择，根据搜索到的歌曲信息，下载自己喜爱的歌曲。在完成这些功能的过程中来巩固自己学到的知识

## 2 应用分析

### 2.1 可行性分析

可行性分析也可称为可行性研究，是在网上调查的基础上并辅助其他的信息渠道，针对基于 iOS 的音乐播放器的开发是否具备必要性和可能性。对新的音乐播放器的开发从技术、操作、经济等不同方面进行分析研究，来保证投资的失误，保证音乐播放器的开发成功。就我们学的知识知道可行性分析可以在最短的时间内分析出这款基于 iOS 的音乐播放器是否可以达到预定的效果。

(1) 经济可行性分析：这款基于 iOS 的音乐播放器的开发是通过苹果系统的 Xcode 来进行的。虽然本人不可能购买苹果笔记本来开发这款产品，但现阶段人们可以在不同的电脑上运行苹果系统。那就是人们所说的黑苹果，通过一定的技术使非苹果笔记本也可运行苹果系统。开发音乐播放器所需的图片，自己可以不通过专业人士来绘制。自己通过破解现有 App store 上的在线应用，来获取自己想要的资源。并且这个过程很是简单，不存在浪费人力物力的现象。所以得出开发一款基于 iOS 的音乐播放器在经济方面是可行的。

(2) 技术可行性分析：开发基于 iOS 的音乐播放器运用到的技术是 Object-C，以及一些 iOS 开发的 UI 设计，某些类库的应用等等。就技术方面而言，自己学习 iOS 技术已经有四五个月的基础了，相对于开发一款音乐播放器而言自己的技术是完全可以胜任的。而自己在黑苹果上装的 Xcode 完全可以胜任这次开发的需求。从而分析得出开发这款音乐播放器在技术方面是可行的。

(3) 社会可行性分析：开发这款音乐播放器，本人不是为了盈利的目的不会损害到任何人的利益。这款音乐播放器虽有一部分按照酷狗音乐来完成，但自己不会发布出去，所以不会影响到酷狗公司的利益。本人开发这款音乐播放器符合国家的方针政策，不影响任何组织的利益，没有宗教方面的宣传同时不会影响社会稳定。所以可以得出本人开发基于 iOS 的音乐播放器在社会方面是可行的。

所以，综上所述本人的基于 iOS 的音乐播放器的开发是可行的。

### 2.2 需求分析

需求分析指的是在建立一个新的应用的时，就这个新的应用的目的、范围、定义及功能这些方面所要做的工作。需求分析是完成一个应用必不可少的部分。在进行一款项目开发的过程中，开发者必须确定客户的需求，在确定了客户需求后才能进行应用的开发。

根据当前社会形势，苹果系统在中国逐渐流行，单单是 iOS 方面的音乐播放器行业也已经具备了相当的规模<sup>[3]</sup>。市场上的音乐播放器品牌繁多，定位不一，基本

满足了各层次消费者的需求。最新的调查显示比较大的音乐播放器市场，并不是很乐观。而外形新颖、小巧轻便灵活的迷你音乐播放器却很受年轻人的青睐和欢迎。虽然国内的较大的音乐播放器销售状况不是很好，但是迷你音乐播放器却是个例外，市场发展前景不容小觑。基于此，设计音乐播放器重点还是在于美观的外形和其吸引人的性能，小体积却可以实现相对更大的功率，新型的迷你音乐播放器要有其他的附加功能，可以更好的吸引用户。

## 2.3 系统开发工具介绍

### 2.3.1 Objective-C 技术

Objective-C 是扩充 C 的面向对象编程语言。它主要用于 Mac OS X 和 GNUstep 这两个使用 OpenStep 标准的系统，而在 NeXTSTEP 和 OpenStep 中它更是基本语言。Objective-C 可以在 gcc 运作的系统写和编译，因为 gcc 含 Objective-C 的编译器。1980 年代初布莱德·确斯在其公司 Stepstone 发明 Objective-C。他对软件设计和编程里的真实可用度问题十分关心，目前主要是编写 iOS 操作系统应用程序的利器。近几年 Objective-C 排名大幅提升，它的流行当归功于 iPhone 的成功，因为 Objective-C 一直被用于编写 iPhone 应用程序<sup>[4]</sup>。Objective-C 是非常实际的语言。它是一个用 C 写成、很小的运行库，只会令应用程序的大小增加很小，和大部分 OO 系统使用极大的 VM 执行时间会取代了整个系统的运作相反。ObjC 写成的程序通常不会比其原始码大很多。而其函式库亦和 Smalltalk 系统要使用极大的内存来开启一个窗口的情况相反。

### 2.3.2 Xcode 开发工具

Xcode 是开发人员建立基于 iOS 的应用最快捷的方式。Xcode 是苹果公司向开发人员提供的集成开发环境。其运行于苹果公司的 Mac 操作系统下。

不管你用 C、C++、Objective-C 或 Java 编写程序，在 AppleScript 里编写脚本，还是试图从另一个奇妙的工具中转移编码，你会发现 Xcode 编译速度极快，每次操作都很快速和轻松<sup>[5]</sup>。

苹果公司为用户提供了全套免费的 Cocoa 程序开发工具。Mac OS X 一起发行，在苹果公司官方的网站下载<sup>[6]</sup>。

### 2.3.3 SQLite 简介

SQLite 是一款轻型的数据库<sup>[7]</sup>。是遵守 ACID 的关联式数据库管理系统，它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百 K 的内存就够了。它能够支持 Windows/Linux/Unix 等等主流的操作系统，同时能够跟很多程序语言相结合，比

如

Tcl、C#、PHP、Java 等，还有 ODBC 接口，同样比起 Mysql、PostgreSQL 这两款开源世界著名的数据库管理系统来讲，它的处理速度比他们都快。SQLite 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<sup>[8]</sup>。不像常见的客户-服务器范例，SQLite 引擎不是个程序与之通信的独立进程，而是连接到程序中成为它的一个主要部分。所以主要的通信协议是在编程语言内的直接 API 调用。这在消耗总量、延迟时间和整体简单性上有积极的作用。整个数据库都在宿主主机上存储在一个单一的文件中。它的简单的设计是通过在开始一个事务的时候锁定整个数据文件而完成的。

### 2.3.4 Visual Studio 简介

Visual Studio 是微软公司推出的开发环境。是目前最流行的 Windows 平台应用程序开发环境。Visual Studio 2010 同时带来了 NET Framework 4.0、Microsoft Visual Studio 2010 CTP，并且支持开发面向 Windows 7 的应用程序。除了 Microsoft SQL Server，它还支持 IBM DB2 和 Oracle 数据库。Visual Studio 中的编程语言运行于 .NET Framework 中。Framework 有助于基于 Web 和 Windows 的应用程序开发，允许来自不同语言的对象共同运行，并标准化了语言引用数据和对象的方式。Visual Studio .NET 是微软公司推出的程序开发工作。和以前的 Visual Studio 相比，Visual Studio .NET 是一个质的飞跃。C# 是 Visual Studio .NET 的重要组成部分。在 Visual Studio .NET 平台上使用 C# 语言可以开发出各种各样的应用程序<sup>[9]</sup>。在 .NET 平台中，C# 语言是一种脚本语言，即简化的 C# 语言。C# 语言已经通过国际标准组织的认证，成为一种国际标准的计算机语言。

.NET Framework 是用于生成、部署和运行应用程序的多语言环境。.NET Framework 通过公共语言运行库、.NET 框架类库、Microsoft 中间语言、JIT 即时编译等机制生成应用程序。

### 3 应用总体设计

在经过可行性分析后本章将主要介绍应用的总体设计。应用所实现的功能的强弱是衡量一款应用的标准。通过对音乐播放器的全面分析并结合调研的情况，得出了这款音乐播放器的结构图与功能模块图。

#### 3.1 音乐播放器的功能模块概况

根据需求分析的结果得到音乐播放器的基本模块包括以下几个模块

(1) . 我的音乐模块：其中主要有本地音乐、我喜欢的音乐、最近播放的音乐的几个部分组成。

(2) . 网络音乐模块： 该模块主要包括最新、歌单、排行榜、歌手等几部分构成。

(3) . 更多更能模块：该模块主要包括换肤、意见反馈、设置及部分构成。根据音乐播放器的功能分析可以画出音乐播放器的功能模块图，如图 1 所示

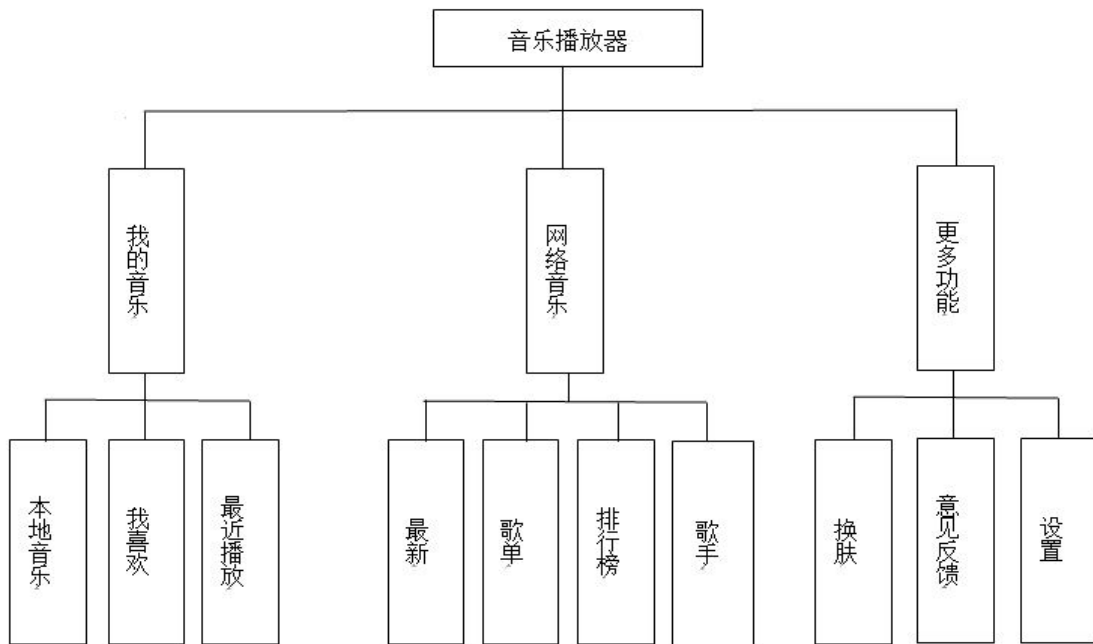


图 1 音乐播放器功能模块图

#### 3.2 音乐播放器各功能模块介绍

各功能模块介绍如下：

(1) 我的音乐模块

用户可以通过此模块来播放本地的音乐。具体播放功能包括音乐的演唱、歌

词的显示、播放进度的显示、歌曲的暂停、播放下一首等功能。用户可以通过本地音乐列表将歌曲存储成我喜欢的音乐，同时在播放的过程中会将播放过的歌曲名存储起来，对用户进行提示。

### (2)网络歌曲模块

用户可以通过本模块进行选择网络上的音乐信息，同时用户可以将网络上的歌曲下载到本地音乐中进行播放，用户根据不同的歌曲信息进行歌曲搜索，方便了用户的使用。

### (3)更多功能模块

用户可以通过此模块对本款音乐播放器进行换肤操作，更改本款音乐的皮肤。用户可以通过意见反馈页面来提出自己的意见。相应的设置模块，用户可以进行一些对本应用的操作，使本应用更加适合用户需求。



## 4 音乐播放器详细设计

### 4.1 音乐播放器界面设计

(1) 我的音乐界面如图 4-1 所示



图 4-1 我的音乐

用户可以通过此界面选择不同的信息。点击本地音乐可以进入音乐播放界面，点击我喜欢可以进入我喜欢界面，其中的歌曲是自己从本地音乐界面中选取出来的，而最近播放界面显示用户最近播放的歌曲。此界面部分相关代码如下：

```
- (void)viewDidLoad
{
    NSMutableArray *mainTableViewArray=[[NSMutableArray alloc] initWithObjects:@"本地音乐",
    @"iPod 音乐", @"我喜欢", @"本地收藏", @"网络收藏", @"最近播放", nil];
    UIView *vi=[[UIView alloc] initWithFrame:CGRectMake(0, 0, 280, 480)];
    [self.view addSubview:vi];
    [vi setBackgroundColor:[UIColor colorWithRed:58/255.0 green:58/255.0
    blue:58/255.0 alpha:1]];
    UILabel *lableOne=[[UILabel alloc] initWithFrame:CGRectMake(11, 72, 25, 120)];
    lableOne.text=@"我的音乐";
    lableOne.userInteractionEnabled=YES;
    lableOne.highlighted=YES;
    [vi addSubview:lableOne];
    UILabel *lableTwo=[[UILabel alloc] initWithFrame:CGRectMake(11, 140+57, 25, 120)];
    lableTwo.text=@"网络音乐";
```

```
labeTwo.textColor=[UIColor colorWithRed:200/255.0 green:200/255.0  
blue:200/255.0 alpha:1];  
labeTwo.userInteractionEnabled=YES;  
[vi addSubview:labeTwo];  
labeThree=[[UILabel alloc] initWithFrame:CGRectMake(11, 280+52, 25, 120)];  
labeThree.text=@"更多功能";  
labeThree.textColor=[UIColor colorWithRed:200/255.0 green:200/255.0  
blue:200/255.0 alpha:1];  
[vi addSubview:labeThree];  
UITapGestureRecognizer *tapGestureOne=[[UITapGestureRecognizer  
alloc] initWithTarget:self action:@selector(choselt:)];  
[labeOne addGestureRecognizer:tapGestureOne];  
[tapGestureOne release];  
}
```

此段代码得到具体含义是，加载初始界面并添加相关的手势来控制 mainTableViewArray 显示的内容。随后在加载 tableView 的过程中运行

UITableView 的方法即：cellForRowAtIndexPath: 和 didSelectRowAtIndexPath: 其中 cellForRowAtIndexPath 是控制每一个 cell 所显示的内容，而 didSelectRowAtIndexPath 则是在这些 cell 被点中时判断哪个被点中了，要运行什么方法。

(1) 网络音乐界面如图 4-2 所示



图 4-2 网络音乐界面

用户可以从此界面中选择网络相关的音乐。点击歌单会弹出歌单界面，其中是各种流行的单曲。点击排行榜则会弹出当前的流行歌曲排行榜，同样点击歌手则会弹出歌手界面。

此界面的代码是只要将最初的 mainTableViewArray 的显示内容改变了，并通过相应的手势来达到内容的改变。手势方法相关的代码如下：

```

-(void)choseIt:(UITapGestureRecognizer *)tap
{
    [mainTableViewArray removeAllObjects];
    CGPoint point=[tap locationInView:self.view];
    if(CGRectContainsPoint(lableOne.frame, point))
    {
        flag=0;
        [array removeAllObjects];
        mainTableViewArray=[[NSMutableArray alloc] initWithObjects:@"本地音乐",@"iPod音乐",@"我喜欢",@"本地收藏",@"网络收藏",@"最近播放", nil];
        lableTwo.highlighted=NO;
        lableThree.highlighted=NO;
        lableOne.highlighted=YES;
        lableOne.highlightedTextColor=[UIColor whiteColor];
        signImage.frame=CGRectMake(43, 68, 4, 120);
    }
    else if(CGRectContainsPoint(lableTwo.frame, point))
    {
        flag=1;
        [array removeAllObjects];
        mainTableViewArray=[[NSMutableArray alloc] initWithObjects:@"搜索",@"最新",@"歌单",@"排行榜",@"歌手",@"分类",@"电台",@"MV", nil];
        signImage.frame=CGRectMake(43, 197, 4, 120);
        lableOne.highlighted=NO;
        lableThree.highlighted=NO;
        lableTwo.highlighted=YES;
        lableTwo.highlightedTextColor=[UIColor whiteColor];
    }
    else
    {
        flag=2;
        [array removeAllObjects];
        mainTableViewArray=[[NSMutableArray alloc] initWithObjects:@"仅在wifi下联网",@"换肤",@"睡眠定时",@"wifi电脑传歌",@"91助手 / iTools传歌",@"意见反馈",@"设置",@"应用",nil];
        signImage.frame=CGRectMake(43, 332, 4, 120);
        lableTwo.highlighted=NO;
        lableOne.highlighted=NO;
        lableThree.highlighted=YES;
        lableThree.highlightedTextColor=[UIColor whiteColor];
    }
    [tableVlew reloadData];
}

```

}

在此方法中不只改变了 mainTableViewArray 的内容，同时改变了选中字体的颜色以及未选中字体的颜色。同时 signImage 的位置也发生相应的改变。

(3)更多功能界面如图 4-3 所示



图 4-3 更多功能界面

用户可以在此界面进行播放器的设置，例如点击换肤则会弹出换肤界面，在换肤界面会有三种皮肤供用户选择。点击意见反馈会弹出意见反馈界面，用户可以通过此界面对本应用提出意见，并保存到相应的数据库中。点击设置会进入设置界面用户可以进行相应的操作。

此界面的代码也与网络音乐界面的代码相同。

这三个界面内容显示在同一个 UITableView 上，其中我的音乐、网络音乐、更多功能是三个 UILabel，即使用来显示文字的。将 UILabel 的属性设置为可以与用户交互后点击这三个 UILabel 就可达到按钮的效果。将三个点击手势加到这三个 UILabel 上，设置运行方法。更改右侧的 UITableView 上显示的内容。同时更改标志图片的位置，是应用显得更加美观。

(3)本地歌曲界面如图 4-4 所示



图 4-4 本地歌曲界面

用户点击初始界面中的本地歌曲，就会弹出此界面。在这个界面中显示的是存储在本地的歌曲名称。点击任何一个歌曲名字就可进行该歌曲的播放。在下面的播放条中可以显示歌曲的总时间，歌曲名称以及播放歌曲的时间按。同时触动向左滑动的手势，将会显示出歌曲的歌词。点击暂停/开始按钮可是歌曲暂停会继续。点击下一首按钮将播放在本地列表中的下一首歌曲。

此界面的相关代码如下：

```
- (void)viewDidLoad
{
    musicPlayBar=[[UIView alloc] initWithFrame:CGRectMake(0, 410, 320, 50)];
    [faView addSubview:musicPlayBar];
    imageOfMusic=[[UIImageView alloc] initWithFrame:CGRectMake(2, 2, 46, 46)];
    imageOfMusic.image=[UIImage
imageNamed:currentMusic.smallImageOfMusic];
    imageOfMusic.userInteractionEnabled=YES;
    [musicPlayBar addSubview:imageOfMusic];
    smallProgressView=[[MyProgerss alloc] init];
    smallProgressView.frame=CGRectMake(51, 42, 250, 2);
    [musicPlayBar addSubview:smallProgressView];
    totalTime=[[UILabel alloc] initWithFrame:CGRectMake(51, 23, 80, 20)];
    totalTime.text=@"00:00";
    [musicPlayBar addSubview:totalTime];
    UIView *sepView=[[UIView alloc] initWithFrame:CGRectMake(90, 28, 1, 10)];
    [musicPlayBar addSubview:sepView];
    smallProgressView.curTime.frame=CGRectMake(100, 23, 80, 20);
```

```

    smallProgressView.curTime.text=@"00:00";
    [musicPlayBar addSubview:smallProgressView.curTime];
    startButton=[UIButton buttonWithType:UIButtonTypeCustom];
    startButton.frame=CGRectMake(235, 10, 20, 20);
    [startButton setBackgroundImage:[UIImage
imageNamed:@"playarticle_play.png"] forState:UIControlStateNormal];
    [startButton addTarget:self action:@selector(startMusic)
forControlEvents:UIControlEventTouchUpInside];
    [musicPlayBar addSubview:startButton];
    UIButton *nextButton=[UIButton buttonWithType:UIButtonTypeCustom];
    nextButton.frame=CGRectMake(275, 10, 20, 20);
    [nextButton setBackgroundImage:[UIImage imageNamed:@"play_bar_next.png"]
forState:UIControlStateNormal];
    [nextButton addTarget:self action:@selector(nextMusic)
forControlEvents:UIControlEventTouchUpInside];
    [musicPlayBar addSubview:nextButton];
    NSMutableArray *musicArray=[[NSMutableArray alloc] initWithObjects:@"一眼万年",@"夜曲",@"最炫民族风",nil];
    NSArray *musicNameArray=[[NSArray alloc] initWithObjects:@"一眼万年",@"夜曲",@"最炫民族风",nil];
    curMusicNumber=0;
    faHeaderView=[[UIImageView alloc] initWithFrame:CGRectMake(0, 0, 320, 44)];
    faHeaderView.image=[UIImage imageNamed:@"navbar_bg@2x.png"];
    faHeaderView.userInteractionEnabled=YES;
    [faView addSubview:faHeaderView];
    [faHeaderView release];
    faRightButton=[UIButton buttonWithType:UIButtonTypeCustom];
    [faRightButton setBackgroundImage:[UIImage
imageNamed:@"nav-menu-icon@2x.png"] forState:UIControlStateNormal];
    faRightButton.frame=CGRectMake(5, 5, 40, 30);
    [faRightButton addTarget:self action:@selector(changeView:)
forControlEvents:UIControlEventTouchUpInside];
    [faHeaderView addSubview:faRightButton] ;
}

```

在本段代码中，对界面上添加了音乐播放条，并添加了用来显示总时间以及播放时间的 Label，在播放条中添加开始（暂停）、下一首按钮。而显示歌曲名的 UITableView 在最初的 UITableView 的 didSelectRowAtIndexPath 方法中生成。相应的代码如下：

```

if(indexPath.row==0)
{
    UITableView *localMusicTableView=[[UITableView
alloc] initWithFrame:CGRectMake(0, 0, 320, 386)];
    localMusicTableView.delegate=self;
}

```

```
localMusicTableView.dataSource=self;
[totalBackView addSubview:localMusicTableView];
whichTableView=4;
[otherTableViewArray removeAllObjects];
[otherTableViewArray addObjectsFromArray:musicArray];
}
```

其中indexPath.row是判断点击了UITableView的哪一行，来进行下一步的操作

(4)我喜欢界面如图 4-5 所示

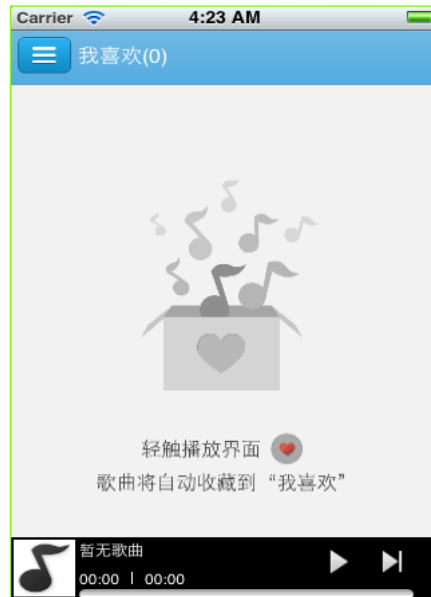


图 4-5 我喜欢界面

用户在本地歌曲界面将某首歌曲设置为我喜欢，则这首歌曲的名字会存储到我喜欢的界面中。

(5)最近播放界面如图 4-6 所示

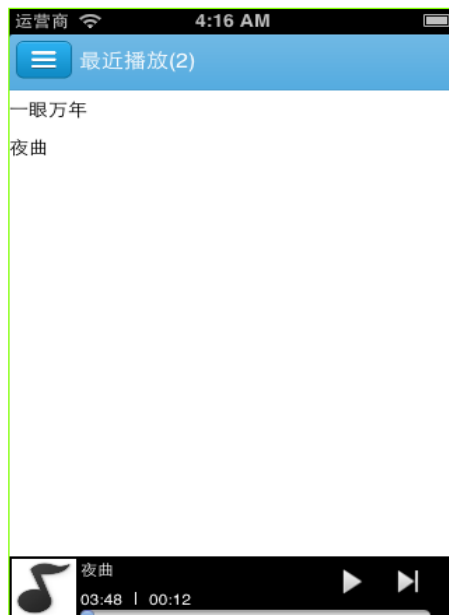


图 4-6 最近播放界面

用户点开此界面会显示出用户播放歌曲的历史记录，在不同时间播放了同一首歌曲目录会自动删除较早的记录保留最新的播放记录。

此界面的相应代码如下：

```
else if(indexPath.row==5)
{
    int j=[lastArray count];
    faHeadLable.text=[NSString stringWithFormat:@"最近播放(%d)",j];
    if(j==0)
    {
        UIImageView *im=[[UIImageView alloc] initWithFrame:CGRectMake(0, 0,
320, 386)];
        im.image=[UIImage imageNamed:@"nillist_bg.png"];
        [totalBackView addSubview:im];
        [im release];
    }
    else
    {
        UIScrollView *sc=[[UIScrollView alloc] initWithFrame:CGRectMake(0, 0,
320, 386)];
        sc.contentSize=CGSizeMake(320, 386*3);
        for(int i=0;i<j;i++)
        {
            UILabel *la=[[UILabel alloc] initWithFrame:CGRectMake(0, i*30, 320,
30)];
            la.text=[lastArray objectAtIndex:i];
            la.font=[UIFont systemFontOfSize:14];
            [sc addSubview:la];
            [la release];
        }
        [totalBackView addSubview:sc];
    }
}
}
```

在此段代码中包含了，在没有播放过音乐时显示那个界面，在播放了歌曲后显示那个界面。

(6) 歌单界面如图 4-7 排行榜界面如图 4-8 歌手界面如图 4-9



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/806222143232011005>