happymusic项目文档

技术小组 se3-se1612

小组成员 阳攀平

指导教师 龚伟

起止时间 2019年07月 ～ 2019年09月

https://github.com/yangpanping/happymusic

重庆师范大学软件工程系

目录

[摘要](#_top)............................................................................................................................ [2](#_top)

1. [涉众分析](#_第一章 涉众分析报告)......................................................................................................4
2. [愿景文档](#_第二章 愿景文档).......................................................................................................6
3. [用况模型](#_第三章 用况模型).......................................................................................................13

3.1[主要用况](#_3.1Music _Player的主要用况).................................................................................................13

3.2[用况-music play](#_3.2用况描述— 播放音乐).......................................................................................13

3.2.1[用况图](#_3.2.1用况图)............................................................................................13

3.2.2前置条件........................................................................................14

3.2.3[基本流](#_3.2.3基本流)............................................................................................14

3.2.4[备选流](#_3.2.4备选流)............................................................................................15

3.2.5[子流](#_3.2.5子流)................................................................................................16

3.2.6[后置条件](#_3.2.6后置条件)........................................................................................16

3.2.7[特殊需求](#_3.2.7特殊需求)........................................................................................16

3.3[被包含的用况-登录](#_3.3被包含的用例 -登录).................................................................................16

3.3.1[简要描述](#_3.3.1简要描述).........................................................................................16

3.3.2[用况图](#_3.3.2用况图).............................................................................................17

3.3.3[前置条件](#_3.3.3前置条件).........................................................................................17

3.3.4[基本流](#_3.3.4基本流).............................................................................................17

3.4[注册](#_3.4注册)..........................................................................................................18

1. [通信图](#_第四章 通信图).............................................................................................................19
2. [类图](#_第五章 类图).................................................................................................................20
3. [顺序图](#_第六章 顺序图)..............................................................................................................21
4. [活动图](#_第七章 活动图)...............................................................................................................22
5. [架构设计](#_第八章 架构设计)............................................................................................................23

8.1[系统架构](#_8.1系统架构)......................................................................................................23

8.2[音频解码-Mediacode](#_8.2音频解码-Mediacode)...................................................................................24

8.3[服务器](#_8.3服务器)...........................................................................................................24

8.4[服务器的交互设计](#_8.4服务器的交互设计)......................................................................................24

8.5[客服端的交互设计](#_8.5客服端的交互设计).......................................................................................24

1. [数据库](#_第九章 数据库).................................................................................................................25
2. [操作说明](#_第十章 操作说明).............................................................................................................26

[运行界面](#_运行界面_x0001__x0001_)...........................................................................................................................29

[术语表](#_术语表)................................................................................................................................32

[参考文献 ..........................................................................................................................33](#_参考文献)

摘要

音乐无国界，可以使人们有一种精神的交流，所以人们都喜欢听听音乐。市面上的音乐播放器有很多，但却很杂乱，各种音乐都有。现在快节奏的生活，人们没有那么多的精力再去挑选优质的音乐，所以开发的一款简洁却有由网友合力推荐的高品质音乐。

关键词：推荐，播放，音乐

# 第一章 涉众分析报告

|  |  |
| --- | --- |
| 系统涉众 | 项目涉众 |
| 用户 | 项目管理层 |
| 管理人员 | 开发人员 |

系统涉众：

1.直接使用系统的人。

2.利用系统使用者成果的人。

3.系统的部署与运营影响的人。

项目涉众：

1.受预算和日程直接影响的人。

2.受产品/系统/解决方案的开放方式直接影响的人。

3.将参与系统的推广销售安装或维护的人员。

|  |  |
| --- | --- |
| 涉众角色 | 解释 |
| 代表用户 | 对音乐有一定追求的人 |
| 普通用户 | 一般使用音乐的人 |
| 构想者 | 对项目提出想法的人 |
| 开发人员 | 对项目进行开发的人 |
| 监察者 | 权威机构 |
| 服务者 | 对系统进行维护更新的人 |

|  |  |
| --- | --- |
| 涉众类型 | 用户 |
| 涉众代表 | 听歌者 |
| 特点 | 项目的使用者 |
| 职责 | 参与项目的讨论，反馈用户的需求 |
| 成功标准 | 可使用，可靠性 |
| 参与 | 不参与系统的开发 |
| 可交付工作 | 无 |

|  |  |
| --- | --- |
| 涉众类型 | 发起人 |
| 涉众代表 | 公司高层 |
| 特点 | 对项目提供财力支持 |
| 职责 | 对项目有一定的期望 |
| 成功标准 | 达到期望 |
| 参与 | 不参与 |
| 可交付工作 | 无 |
| 意见/问题 | 无 |

|  |  |
| --- | --- |
| 涉众类型 | 开发者 |
| 涉众代表 | 开发人员 |
| 特点 | 无 |
| 职责 | 开发项目 |
| 成功标准 | 完成用户和发起人的需求和期望 |
| 参与 | 参与项目的建设 |
| 可交付工作 | 有 |
| 意见/问题 | 无 |

用户分类（按用户的喜好分类）

|  |  |
| --- | --- |
| 用户分类 | 音乐爱好者 |
| 特征 | 对音乐有一定的追求，是潜在的大量用户 |
| 能力 | 有一定的网络能力 |
| 成功标准 | 可靠性，低成本的附加功能 |
| 参与者 | 歌迷,运营者, 审核者，监察者 |

|  |  |
| --- | --- |
| 用户分类 | 普通使用者 |
| 特征 | 对于音乐没有很高的追求的人，是潜在的用户 |
| 能力 | 有一定的手机应用能力 |
| 成功标准 | 可靠性 |
| 参与者 | 用户,运营者, 审核者，监察者 |

|  |  |
| --- | --- |
| 用户分类 | 没时间了解 |
| 特征 | 喜欢听歌，时间很少 |
| 能力 | 有一定的应用能力 |
| 成功标准 | 可靠性 |
| 参与者 | 用户，运营者,监察者 |

# 第二章 愿景文档

•愿景文档的目的

愿望的憧憬，是项目人员发自内心对产品未来的一种远期追求，需要回答产品存在的意义以及未来的前景。它回答了“追求什么”这个问题，给了项目人员一定的动力和憧憬。有了愿景文档，即使是一个简短甚至不完全的文档，都有助于确保所有项目人员朝着同一个目标工作。团队有了共同的目标和剧本，或者还可以有共同的心理，否则，团队的工作目标将是未知的或者相互冲突的，很快就会产生混乱。

•产品综述

•业务范围

音乐播放器覆盖范围很广，无论任何年龄段，任何职业，都会有听音乐的需求。该音乐播放器提供了很多的功能，包括在线播放、分享以及下载，用户也可以通过评论相互交流。

•用户描述

2.1用户/市场统计

随着越来越信息化的社会，像MP3，DVD等相关播放音乐的产品越来越不能满足人们的需要，基于此，我们对用户/市场的统计大概如下：

#这是一款新研发的音乐播放器，能满足用户随时随地的听歌。

#该产品是基于比较大众化的产品，比较大众化的东西往往能让更多的人使用

#新产品可能一开始并不能吸引大量的用户，但是会有很多潜在的用户，新用户会带动潜在的用户使用该产品

在产品的演进过程中，我们期望为产品增加更多方便快捷的功能以吸引更多的用户。

2.2用户剖析

\*喜欢追求新事物的人，有一定的计算机技术，比较喜欢把新研发出来的东西弄透彻，要求产品有一定的可靠性、可维护性

\*对听音乐有需求的人，能灵活的使用手机，只要求满足她们播放音乐的需求，要求产品的可靠性

\*对不同音乐播放器进行考察的人，有一定的计算机能力

2.3用户环境

用户的环境是任意的，该产品应用于移动端的用户，运用于Android、iOS系统的手机上，以后会向其他平台拓展。

1.个人应用，需要手机或者个人电脑

2.目前支持Windows及安卓平台，未来将支持ios等其他平台

2.4关键用户需求

2.4.1从音乐播放器的使用者的角度

①播放歌曲

②下载歌曲

③导入本地歌曲

④歌曲查找

⑤歌手写真及自定义背景图片

⑥自建歌单

⑦歌曲评论

⑧收藏歌曲

2.5替代和竞争对手

①MP3

强项：小巧，便于随身携带。

弱项：无法存储大量音乐。

②DVD

强项：可以边看MV边听。

弱项：不方便携带，存储的音乐有限。

•产品简介

3.1产品前景

音乐播放器是一款可以方便，随时随地播放音乐的应用。随着社会的快速发展，生活节奏越来越快，人们压力也随之增加，人们听音乐来缓解压力。但音乐可能会杂乱无章，所以我们向用户提供高品质的音乐

3.2产品定位陈述

|  |  |
| --- | --- |
| 为了 | 对听歌有需求的用户，希望能听到优质的 |
| 我们小组 | 基于现在音乐播放器比较杂乱，各种音乐混合 |
| 音乐播放器 | 是为了让人们身心愉悦，得到放松的软件 |
| 音乐播放器 | 为用户提供方便快捷的听歌方式 |
| 不像 | 比之QQ音乐/酷狗音乐更加简洁 |
| 我们的产品 | 更简洁，为无 时间的人提供高品质音乐 |

3.3能力总结

3.4假定和依赖关系

（待讨论......）

•特性属性

（1）：播放音乐

|  |  |
| --- | --- |
| 属性 | 描述 |
| 状态 | 批准的 |
| 优先级、效益 | 关键的 |
| 工作量 | 中 |
| 风险 | 低 |
| 稳定性 | 高 |
| 目标发布 | 版本1.0 |
| 分配给 |  |
| 原因 |  |

（2）：歌曲搜索

|  |  |
| --- | --- |
| 属性 | 描述 |
| 状态 | 批准的 |
| 优先级、效益 | 关键的 |
| 工作量 | 中 |
| 风险 | 中 |
| 稳定性 | 高 |
| 目标发布 | 版本1.0 |
| 分配给 |  |
| 原因 |  |

（3）：歌曲评论

|  |  |
| --- | --- |
| 属性 | 描述 |
| 状态 | 批准的 |
| 优先级、效益 | 有用的 |
| 工作量 | 高 |
| 风险 | 高 |
| 稳定性 | 中 |
| 目标发布 | 版本2.0 |
| 分配给 |  |
| 原因 |  |

（4）：下载

|  |  |
| --- | --- |
| 属性 | 描述 |
| 状态 | 批准的 |
| 优先级、效益 | 重要的 |
| 工作量 | 高 |
| 风险 | 高 |
| 稳定性 | 中 |
| 目标发布 | 版本2.0 |
| 分配给 |  |
| 原因 |  |

（5）：用户信息管理

|  |  |
| --- | --- |
| 属性 | 描述 |
| 状态 | 建议的 |
| 优先级、效益 | 重要的 |
| 工作量 | 高 |
| 风险 | 高 |
| 稳定性 | 高 |
| 目标发布 | 版本2.0 |
| 分配给 |  |
| 原因 |  |

（6）：本地导入

|  |  |
| --- | --- |
| 属性 | 描述 |
| 状态 | 批准的 |
| 优先级、效益 | 重要的 |
| 工作量 | 低 |
| 风险 | 低 |
| 稳定性 | 高 |
| 目标发布 | 版本1.0 |
| 分配给 |  |
| 原因 |  |

•产品特性

1.特性1

用户信息管理：用户的账号密码管理，更改用户的基本信息，新建用户及老用户的注销管理。

2.特性2

播放歌曲：除提供基本的播放之外，还提供多种播放模式选择。支持多种格式的音乐播放。

3.特性3

喜欢歌单：用户自己的喜好或者歌曲风格自己建立播放歌单。

4.特性4

歌曲检索：用户通过输入关键字进行检索相关歌曲，达到快速查找歌曲。

6.特性6

下载歌曲：用户可下载喜欢的歌曲到自己的设备。

•其他产品需求

6.1可应用标准

产品符合的标准：

平台标准：安卓

法律标准：FDA，FCC

质量和安全标准：UL，ISO，CMM

通信标准：TCP/IP，ISDN

6.2系统需求

支持的主机操作系统：android

内存：500MB

运行环境：

网络架构：完全支持TCP/IP协议

6.3许可证、安全及安装

许可证：BSD许可证

安全及安装：相关事项有待系统开发......

6.4性能需求

数据精确性：保证数据的准确率为百分百。

系统响应时间：系统对大部分操作系统的响应时间在1～5秒。

适应性：满足运行环境在允许操作系统之间的安全转换和其他应用软件的独立运行要求。

稳定性：系统具备长期持续工作的能力，确保用户可以随时使用。

安全性：身份认证机制和密码更改模块，用户权限可以通过网站管理员更改。

可靠性：采用良好的操作系统和数据库，确保数据的一致性和完整性，并使系统免受病毒的感染，确保信息来源安全可靠。

•文档需求

7.1 用户手册

无

7.2在线帮助

无

7.3安装指南、配置和自述文件

无

7.4标记和打包

无

•词汇表

# 第三章 用况模型

### 3.1Music \_Player的主要用况



### 3.2用况描述— 播放音乐

简要描述

该用况描述了听歌者如何使用音乐播放器播放音乐.

##### 3.2.1用况图



3.2.2前置条件：

网络连接有效，系统准备就绪。

##### 3.2.3基本流

**{打开终端}**

1.当参与者**用户**打开音乐播放器**终端**，用况开始启动。

**{显示主页面}**

2.系统显示本地歌单。

**{选择分类}**

3.参与选择分类。

4.执行“选择类别（Choose Category）”。

**{选择歌曲}**

5.用户选择歌曲。

**{加入播放列表}**

6.系统将歌曲加入**播放列表**。

**{高亮歌曲}**

7.系统高亮当前播放歌曲。

**{显示播放界面}**

8.系统显示播放的顺序、上一曲、下一曲、暂停，播放的顺序有随机播放，单曲循环，顺序播放。

**{获取信息}**

9.系统读入歌曲的歌手，时长显示到播放界面。

{**解码音频}**

10.系统获取歌曲的音频并使用**解码器**对**音频**进行解码，将解码的数据传到**终端**。

**{播放歌曲}**

11.系统开始播放歌曲并同步歌词和**进度条**。

12.用户选择顺序播放。

**{切换歌曲}**

13.歌曲的**进度条**到达结束位置时，系统切换到**播放列表**的下一首。

14.用户选择上一曲下一曲。

15.系统结束当前歌曲，并播放上一曲或者下一曲歌曲，重复7。

{**暂停歌曲}**

16.参与者用户选择暂停歌曲。

17.系统暂停当前播放歌曲，并记录歌曲播放的**进度条**。

**{终止用况}**

18.当参与者退出终端，系统结束播放音乐，用况终止。

##### 3.2.4备选流

5.1播放异常

5.1.1没有播放权限

在**{加入播放列表}**处，如果用户选择的歌曲没有播放权限，则

1.系统显示没有播放权限

2.回到{**选择歌曲}**处

5.1.2没有歌词、专辑或歌手信息

在**{获取信息}**处，如果用户点击的歌曲没有歌词、专辑或歌手信息，则

1.系统显示播放的歌曲没有歌词等信息。

2.回到**{解码音频}**处。

5.2网络异常

5.2.1无法加载信息

在**{显示主页面}**和**{终止用况}**之间，如果网络连接无效，则

1.系统显示您的网络出错了哦！

2.事件流恢复到中断的点。

5.3 选择歌单里的歌曲

5.3.1 选择歌单里的歌曲

在**{选择歌曲}**处，参与者歌迷选择歌单里面的歌曲时，则

1.系统将**播放列表**清空。

2.系统将歌迷选择的歌曲所处的**歌单**里的歌曲全部加入播放列表。

3.回到**{高亮歌曲}**处。

##### 3.2.5子流

6.1选择类别（Choose Category）

1.系统显示分类列表。

2.参与者用户选择按歌手分类。

3.系统读取并显示歌手列表。

4.参与者用户选择歌手。

5.系统读取并显示歌手专辑。

6.参与者用户选择专辑。

7.系统读取并显示专辑的歌曲列表。

8.事件流恢复到下一步。

##### 3.2.6后置条件

无

##### 3.2.7特殊需求

时间约束

每首歌在缓冲的时候不得超过三秒

### 3.3被包含的用例 -登录

##### 3.3.1简要描述

该用况是一个被包含的用况，描述了用户登录系统

##### 3.3.2用况图



##### 3.3.3前置条件

系统已启动并可以和终端通过网络连通

##### 3.3.4基本流

**{登录}**

1.用户打开客服端

**{输入信息}**

2.输入id和密码

**{开始登录}**

3.点击登录

**{获取权限}**

4.软件获取用户操作

**{数据匹配}**

5.进入数据库匹配

6.软件返回匹配情况

（1）匹配成功，用户进入界面

（2）匹配失败，用户可能是忘记密码或者未注册用户

7.用况终止

### 3.4注册

**基本流**

1.用户启动客服端

**{获取注册提示}**

2.点击注册按钮

**{开始注册}**

3.软件获取操作

**{进入注册}**

4.进入注册页面

**{填写信息}**

5.填写用户基本信息

**{填写id和密码}**

6.填写id和密码

**{提交信息}**

7.提交注册信息

**{保存并于数据库匹配}**

8.软件将信息保存到数据库

（1）用户与数据库已有的冲突，将反馈用户，重新填写冲突信息选项

（2）数据库无该组数据，将保存数据返回注册页面

**{成功}**

9.提示注册成功

**{返回}**

10.返回登录界面

11.用况终止

# 第四章 通信图



# 第五章 类图



# 第六章 顺序图



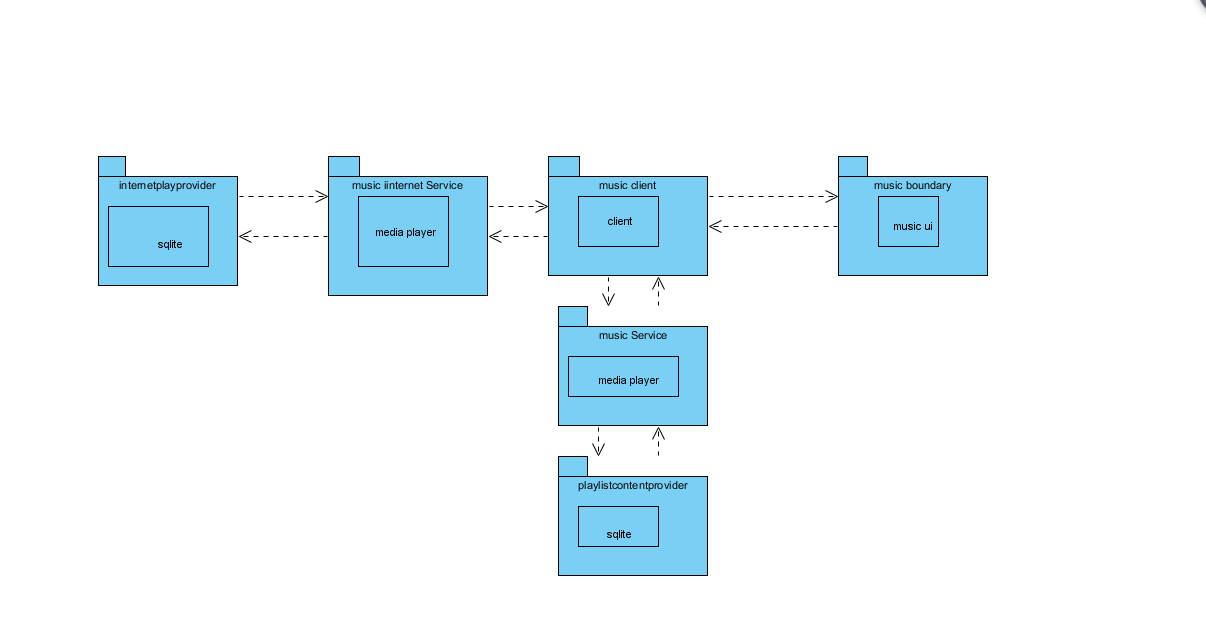
# 第七章 活动图



# 第八章 架构设计

### 8.1系统架构

本音乐软件采用的c/s架构，设计了两个服务器，一个为本地音乐播放，一个为网络搜索播放服务器。如图为音乐播放器的架构图，主要分为客服端层，媒体播放层，数据层。



### 8.2音频解码-Mediacode

本Internetplayprovider主要提供用户网络播放音频。当客服端对music Internet Service进行请求时，music Internet Service 会查找Internetplayprovider存储的信息，例如歌手id，歌名title等，解码使用的是Android自带的Mediacode，安卓系统各种音频解码都支持，要使用可以调用接口

### 8.3服务器

Bmob云端服务器，是比较稳定的服务器，对用户的id＆密码进行存储。请求采用的HTTP协议进行网络传输。Connet类主要用于对服务器的连接。

用户数据的读写操作（set（）/get（）），set（）主要用于用户的写操作，用户进行id＆password的写入。get（）主要读取用户的id＆password。

### 8.4服务器的交互设计

1. 启动服务器：创建InternetService，调用listen（）函数开启监听客服端。
2. 当监听到新请求时，InternetService用connet（）连接进行数据连接
3. 当服务器接收到客服端的请求时，通过stream来获取客服端传来的消息，通过该消息将相对应的数据通过stream调用get（）函数写入数据传回客服端。
4. 当服务器接收到disconnected时，断开连接。

### 8.5客服端的交互设计

1. 当用户搜索在线音乐时，调用connet类进行网络连接，发送请求消息。
2. 服务器接收消息后，向客服端返回。
3. 客服端接收后，创建一个search类进行请求。
4. 服务器接收消息后，返回一个listview。
5. 客服端接收信息，创建musicplay的实例。向服务器发送播放请求。
6. 服务器返回解码音频
7. 客服端读取播放。

# 

# 

# 第九章 数据库

Music：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型 | 宽度 | NULL | 注 |
| Title | 歌名 | String | 20 | No |  |
| Singer | 歌手 | String | 30 | No |  |
| Lyric | 歌词 | String | 30 | No |  |
| ID | 序号 | Int | 10 | NO | 关键字 |

Singer：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型 | 宽度 | NULL | 注 |
| ID | 序号 | int | 10 | No | 关键字 |
| Name | 歌手姓名 | String | 30 | No |  |
| Category | 类别 | String | 30 | No |  |

User：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段含义 | 数据类型 | 宽度 | NULL | 注 |
| Objectid | 序号 | String | 10 | No | 关键字 |
| Uername | 用户名称 | String | 30 | No |  |
| user password | 用户密码 | String | 30 | No |  |

# 第十章 操作说明

Context: Music

Operation specification:getMusicDetails()

Operation intent: return MusicInformation.

Operation signature: Music::getMusicDetails() obtainMusicInformation:MusicInformation

Logic description(pre- and post-conditions):

pre: self->exists()

post:result = MusicInformation

Other operations called: none

Events transmitted to other objects:Album

Attributes set:none

Response to exceptions:none defined

Non-functional requirements: none defined

Context:Album

Operation specification:getAlbumDetails()

Operation intent: return AlbumInformation

Operation signature: Album::getAlbumDetails() obtainAlbumInformation:AlbumInformation

Logic description(pre- and post-conditions):

pre: self→exists()

post:result = AlbumInformation

Other operations called: none

Events transmitted to other objects:Singer

Attributes set:none

Response to exceptions:none defined

Non-functional requirements: none defined

Operation specification:getAlbumMusics()

Operation intent: return Musics

Operation signature: Album::getAlbumMusics() obtainMusic:Musics

Logic description(pre- and post-conditions):

pre: self→exists()

post:result = Musics

Other operations called: none

Events transmitted to other objects:MusicPlayer

Attributes set:none

Response to exceptions:none defined

Non-functional requirements: none defined

Context:Singer

Operation specification:getSingers()

Operation intent: return singers

Operation signature: Singer::getSingers() obtainSingerObjects:Singers

Logic description(pre- and post-conditions):

pre: self→exists()

post:result = Singers

Other operations called: none

Events transmitted to other objects:MusicPlayer

Attributes set:none

Response to exceptions:none defined

Non-functional requirements: none defined

Operation specification:getSingerAlbums()

Operation intent: return SingerAlbums

Operation signature: Singer::getSingerAlbums() obtainSingerAlbums:SingerAlbums

Logic description(pre- and post-conditions):

pre: self→exists()

post:result = SingerAlbums

Other operations called: none

Events transmitted to other objects:MusicPlayer

Attributes set:none

Response to exceptions:none defined

Non-functional requirements: none defined

# 运行界面Screenshot_20190923_225928_me.wcy.musicScreenshot_20190923_225931_me.wcy.music



# 术语表

|  |  |
| --- | --- |
| 名称 | 定义 |
| 歌迷 | 听歌的人 |
| 运营者 | 制作并上传歌单的人 |
| 审核者 | 对歌曲进行上传或删除的人 |
| 监察者 | 对违规言论进行删除的人 |
| 分类列表 | 对歌曲进行分类的方式 |
| 播放列表 | 播放的歌曲列表 |
| 音频 | 一种可以被识别并解码播放的文件 |
| 进度条 | 记录歌曲播放的进度 |
| 解码器 | 对音频进行识别并解码 |
| 歌单 | 存放歌曲的列表 |
| 歌单标签 | 对歌单风格进行描述的词语 |
| 终端 | 系统在设备上启动的一个服务 |

## 参考文献

Kurt Bittner,Ian Spence,Use Case Modeling[M].清华大学出版社：2003.8

Dean Leffingwell,Don Widrig,软件需求管理用例方法(第二版)[M].中国电力出版社

Simon Bennett,Steve McRobb and Ray Farmer,Object-Oriented Systems Analysis and Design(4th Edition)

Android移动开发基础案例教程