# 1 可行性分析

## 1.1技术可行性

本次系统采用b/s的架构来实现开发系统，后端的使用当下比较主流的语言java来开发，前端使用的是js和html来进行展示。主要的框架是基于java开发的springboot框架，以及前端的vue框架。springboot具有快速构建项目，对主流开发框架的无配置集成，并且项目可独立运行，不需要依赖单独的servlet容器。Vue是一个前端的构建框架，具有，响应式编程，组件化两大特点，数据和结构分离，虚拟DOM，运行速度快。

存储方面： 主要使用Mysql和redis来存储。

选择mysql的原因主要有:

性能卓越服务稳定，很少出现异常宕机  
开放源代码且无版权制约，自主性强、使用成本低。  
历史悠久、社区及用户非常活跃，遇到问题，可以很快获取到帮助。  
软件体积小，安装使用简单，并且易于维护，安装及维护成本低。  
支持多种操作系统，提供多种api几口，支持多种开发语言。

Redis是基于内存的，速度比较快，支持多种数据类型，可以用来缓存热数据，减少查询数据库的次数。提高查询数据的响应时间，保证系统和用户的良好交互体验。

## 1.2 经济可行性

本次系统主要是为用户提供音频的服务，所以存储和网络就比较重要，所以本次打算使用七牛云的对象存储来来存储具体的音频文件，使用一个云主机来进行部署。在经济的可用性是可以负担的。

## 1.3 社会因素可行性(市场和政策)

本系统主要为用户所服务，所以用户的信息安全就显得非常的重要。为了保证用户信息的安全性，主要使用token的保证用户登陆信息的安全性。并进行了必要的校验工作。后端使用了springsecuriy这个框架，来确保系统安全。

## 1.4 开发方案可行性

对待实现的功能和现有的技术进行了评估。借鉴历史的相关系统，综合评估项目中的重难点。具体包括用户的行为分析，用户的访问历史记录，用户和系统之间怎么实现良好的交互，怎么存储大量的歌曲文件，对用户评论内容的过滤，后台系统对用户的行为信息的管理，对用户行为的分析的维度，对歌曲流行趋势的分析等等技术的实现都进行了评估。对系统中可能出现的风险，以及异常也做了一些评估。

综合各个方面，确定的开发方案可实现的。

# 2 需求分析

编写目的： 随着人们生活节奏的不断加快，对生活的品质要求越来越高，如何利用碎片化的时间来最大化的愉悦身心，放松心情就显得很重要，本系统就是音频管理系统，为用户提供比较丰富的音频资源，用户也可以创建自己的歌单来收集自己喜欢的音乐，最主要的一个功能，就是根据用户听歌习惯，生成用户的听歌分析报告，让用户能实时的了解自己的听歌类型，更好的放松自己的身心。

## 2.1 任务概述

该音乐软件目前已经有完善的功能和需求，研制这个系统是为了给用户以好的生活体验，和更好的利用音频资源来分析需求。

### 2.1.1开发意图

### 2.1.2 应用目标

通过本系统，能帮助用户找到自己喜欢的音乐，并使创作者的音频能进行更好的管理和分析。帮助好的作品能让更多的人听到。

### 2.1.3 产品描述

本产品是一个b/s架构的产品，用户可以方便的使用浏览器来进行访问。存储使用的mysql数据库。并使用vue展示具体的信息，给用户提供良好的交互体验。

### 2.1.4 产品功能

#### 1 系统功能划分

系统主要分为游客，普通用户，创作者，系统管理员三个角色，

游客可以查询歌曲，听歌，但是不能评论，创建自己的歌单，不能收藏。

普通用户可以登录，注册，浏览歌曲，收藏歌单，根据自己的喜好创建自己歌单信息，浏览自己的听歌历史，评论歌曲，查看自己的听歌报告等功能。

创作者除了具有普通用户的功能之外，还可以上传自己的作品，并且可以通过登录后台，查看自己的作品的访问量，评论，以及综合评分等维度的信息，可以方便的为创作者提供有关作品比较直观的数据。便于后期的创作。

系统管理员具有最高的权限，具体的可以实现对用户信息的管理，对歌曲的管理，歌单的管理，评论的管理，收藏的管理，查看每首歌曲的一个访问情况，歌手的地域分布，用户的分布情况，对后端的接口管理，对用户的的授权管理，角色管理，角色分配，查看歌手和用户以及歌曲的分布等功能。并可以查看系统的实时的监控信息和设置系统的一些配置信息等功能。

#### 2 系统功能模块划分

##### 2.1 用户信息模块：

主要包括用户信息的增加，修改，更新和删除。普通用户可以进行注册和修改自己的信息。查看自己的浏览歌曲的历史等的信息，管理员可以对用户的基本信息进行分析，具体的包括用户的地域，以及用户的性别，出生年龄等的信息。

##### 歌曲管理模块：

对系统中的所有歌曲进行管理，包括歌曲的风格，歌曲地域分布，歌曲的增加，修改和删除，以及歌曲的文件的存储。由于歌曲的文件比较大，本系统采用存储到七牛云的第三方平台上来进行实现的。增加的了歌曲的各个维度的分类，方便用户进行方便的搜索使用。对部分的热门歌曲的显示做了一层缓存，提高用户的检索效率，有一个更好的系统体验。

歌手管理模块：

歌手信息的增删改查和统计实现，用户可以查询歌手的基本信息，包括简介，代表作，出生地等的信息。每个歌手会关联自己的音乐作品。用户可以根据歌手来搜索自己喜欢的音乐。系统后台也可以统计每个歌手的作品信息以及歌手分布等维度的统计信息。

##### 角色管理模块:

本系统是基于RABC模型来对用户进行权限管理的。角色管理模块主要是对具体的角色信息进行动态的管理，具体包括角色的创建，角色的授权，角色的查询等功能，实现不同角色的可以访问不同的资源。这里的角色授权分为两个维度，一个是菜单的授权，一个是具体的接口的授权。具有更好的灵活性。

##### 歌单管理模块:

将不同的歌曲组成歌单，进行展示，给用户提供更好的检索体验和歌曲资源，系统也会根据用户的听歌内容，每天给用户推送个性化的歌单。实现歌曲内容的最大共享。

##### 评论管理模块:

用户可以对自己喜欢的歌曲进行评论，后台集中对一首歌曲的评论进行管理，包括对评论敏感词汇的过滤，一首歌曲的评论的具体数量。用户评论的点赞数等信息。以及对歌曲评论的动态管理。

##### 接口管理模块:

一般是由管理员进行管理，可以对后台菜单进行动态的添加和授权。

##### 统计信息模块:

主要用户信息统计，歌手的信息统计，歌曲的信息统计等子功能。

##### 系统监控模块:

主要是监控系统的硬盘信息，系统的cpu的信息，系统的存储等信息。实时的查看整个系统的运行状况。

## 2.2 具体的需求分析

### 2.2.1 系统的流程图

两个图，一个是用户的登录之后的图

一个是管理员的的图

### 2.2.2 系统的数据流图

画出数据流图

### 2.2.3 数据字典

数据流的描述

数据元素词条描述

### 2.2.4 运行环境

<https://blog.csdn.net/chenguang79/article/details/2224999>

# 3 系统概要设计（画出e-r图）

## 3.1 需求规定

输入输出规定:

## 3.2 运行环境

硬件环境：

最低配置：内存：1G以上，硬盘：20GB以上空间

推荐配置：内存：1GB以上，硬盘：100GB以上空间。

软件环境：Windows10

Mysql:5.7

JDk1.8+

Springboot 2.0+

Redis5.0+

Vue 3.0

## 3.3 基本设计概念和处理流程

### 3.3.1 E-R图

用户信息的E-R 图

歌曲信息的E-R图

歌单信息的E-R图

歌曲的分类的E-R图

歌手信息的E-R图

用户和歌曲，歌单之间E-R图

歌曲，歌单，歌手之间E-R图。

### 3.3.2 模块结构

就是分为管理员模块和用户模块，各自阐述各个什么所具有的功能信息。

需要一个E-R图

### 3.3.3 功能和需求之间的关系

各项需求的实现同各种程序的分配关系。

也就是那个功能需要什么的用户进行访问，需要一个表格信息来说明。

3.3.4 人工处理过程

1 创建歌曲： 用户需要将曲库中大量的歌曲进行导入。

2 创建预设系统歌单： 需要将不同的歌单的信息导入到系统中去。

3.4 接口设计

3.4.1 用户接口

3.4.2 歌曲接口

3.4.3 歌单接口

3.4.4 歌手接口

3.4.5 评论接口

3.4.6 用户角色的接口

3.4.7 系统访问的接口

3.4.8 用户歌单推荐设计

3.4.9 用户的评论过滤的设计

3.5 系统数据结构设计

3.5.1 用户信息的数据库设计

3.5.2 歌曲的数据库设计

3.5.3 歌手的数据库设计

3.5.4 歌单的数据库设计

3.5.5 评论的数据库设计

3.5.6 歌单类别的数据库设计

3.6 系统出错设计。

3.6.1 出错信息

各种出错的情况的考虑。

4 系统详细设计

4.1 程序系统的结构

4.2 系统功能实现

数据库DB设计

用例图

用例图，软件设计的方面

5 系统的单元测试

主要类的设计

软件测试