Rythmn音乐播放平台-软件规格说明书(SRS)

目录

[0.版本记录 2](#_Toc102078042)

[1.引言 2](#_Toc102078043)

[1.1 编写目的 2](#_Toc102078044)

[1.2 背景 2](#_Toc102078045)

[1.3 读者对象和阅读建议 2](#_Toc102078046)

[1.4 参考资料 3](#_Toc102078047)

[1.5基线 3](#_Toc102078048)

[2.总体概述 3](#_Toc102078049)

[2.1 软件概述 3](#_Toc102078050)

[2.2 软件功能 4](#_Toc102078051)

[2.3 用户类及特征 9](#_Toc102078052)

[2.4 假设和依赖关系 9](#_Toc102078053)

[3.系统功能 9](#_Toc102078054)

[3.1 功能需求 10](#_Toc102078055)

[3.2.1描述约定 12](#_Toc102078056)

[3.2.2设计和实现的约束 12](#_Toc102078057)

[4. CSCI外部接口需求 12](#_Toc102078058)

[4.1 CSCI用户接口 13](#_Toc102078059)

[4.2 硬件接口 13](#_Toc102078060)

[4.3 软件接口 13](#_Toc102078061)

[4.4 通信接口 13](#_Toc102078062)

[5.其他非功能性需求 13](#_Toc102078063)

[5.1 性能需求 13](#_Toc102078064)

[5.2 安全性需求 13](#_Toc102078065)

[6.软件质量特性 14](#_Toc102078066)

[6.1 可靠性 14](#_Toc102078067)

[6.2 易用性 14](#_Toc102078068)

[6.3 可维护性 14](#_Toc102078069)

[6.4 扩展性 14](#_Toc102078070)

[6.5 可测试性 15](#_Toc102078071)

[6.6 健壮性 15](#_Toc102078072)

[7合格性规定 15](#_Toc102078073)

[8.数据库 16](#_Toc102078074)

[9.尚未解决的问题 16](#_Toc102078075)

[10.注释 17](#_Toc102078076)

[附录 17](#_Toc102078077)

## 0.版本记录

| **版本号** | **说明** | **日期** |  |
| --- | --- | --- | --- |
| V1.0 | 初版需求 | 2022/3/31 |  |
|  |  |  |  |
|  |  |  |  |

## 1.引言

### 1.1 编写目的

该文档首先给出了整个系统的整体网络结构和功能结构的概貌，试图从总体架构上给出整个系统的轮廓，然后又对功能需求、性能需求和其它非功能性需求进行了详细的描述。其中对功能需求的描述采用了 Petri Net, ER图，UML等用例模型方式，而且还给出了非常直观的用例图。这些文字和图形都为了本文档能详细准确地描述用户的需求，同时也为用户更容易地理解这些需求的描述创造了条件。 该文档详尽说明了这一软件产品的需求和规格，这些规格说明是进行设计的基础，也是编写测试用例和进行系统测试的主要依据。同时，该文档也是用户确定软件功能需求的主要依据。

### 1.2 背景

Rythmn音乐播放平台的初版需求规格说明编写

### 1.3 读者对象和阅读建议

本文档的主要内容共分 4 部分：综合描述、系统特性、和非功能性需求和外部接口描述。综合描述部分主要对系统的整体结构进行了大致的介绍；系统特性部分对系统的功能需求进行了详细描述，是本文的主要部分；非功能性需求部分对非功能需求进行了详细的描述；外部接口需求部分对用户界面、软件接口、硬件接口和通讯接口等进行了描述。

本文档面向多种读者对象：

* 项目经理：项目经理可以根据该文档了解预期产品的功能，并据此进行系统设计、项目管理。
* 设计员：对需求进行分析，并设计出系统，包括数据库的设计。
* 程序员：配合《设计报告》，了解系统功能，编写《用户手册》。
* 测试员：根据本文档编写测试用例，并对软件产品进行功能性测试和非功能性测试。
* 用户：了解预期产品的功能和性能，并与分析人员一起对整个需求进行讨论和协商。

### 1.4 参考资料

* 软件需求规格说明书 IEEE 标准
* **国标GB**8567-2006

### 1.5基线

* 可行性分析研究报告（FAR）

## 2.总体概述

### 2.1 软件概述

#### 2.1.1 项目介绍

基于Web技术，打造一款跨平台，支持多种机器类型与操作系统的在线音乐媒体播放平台，兼具一定的社交分享功能。

#### 2.1.2 产品环境介绍

系统为 B/S 架构。以下是系统的软件环境：

* 客户端操作系统：Windows 系列操作系统，Linux的各种发行版。
* 客户端浏览器：Edge,Chrome浏览器。
* 应用服务器端操作系统：Ubuntu Server 18.04
* 应用服务器：Tomcat
* 数据库访问：Mybatis
* 数据库端操作系统：Ubuntu Server 18.04
* 数据库系统：MySQL 5.7

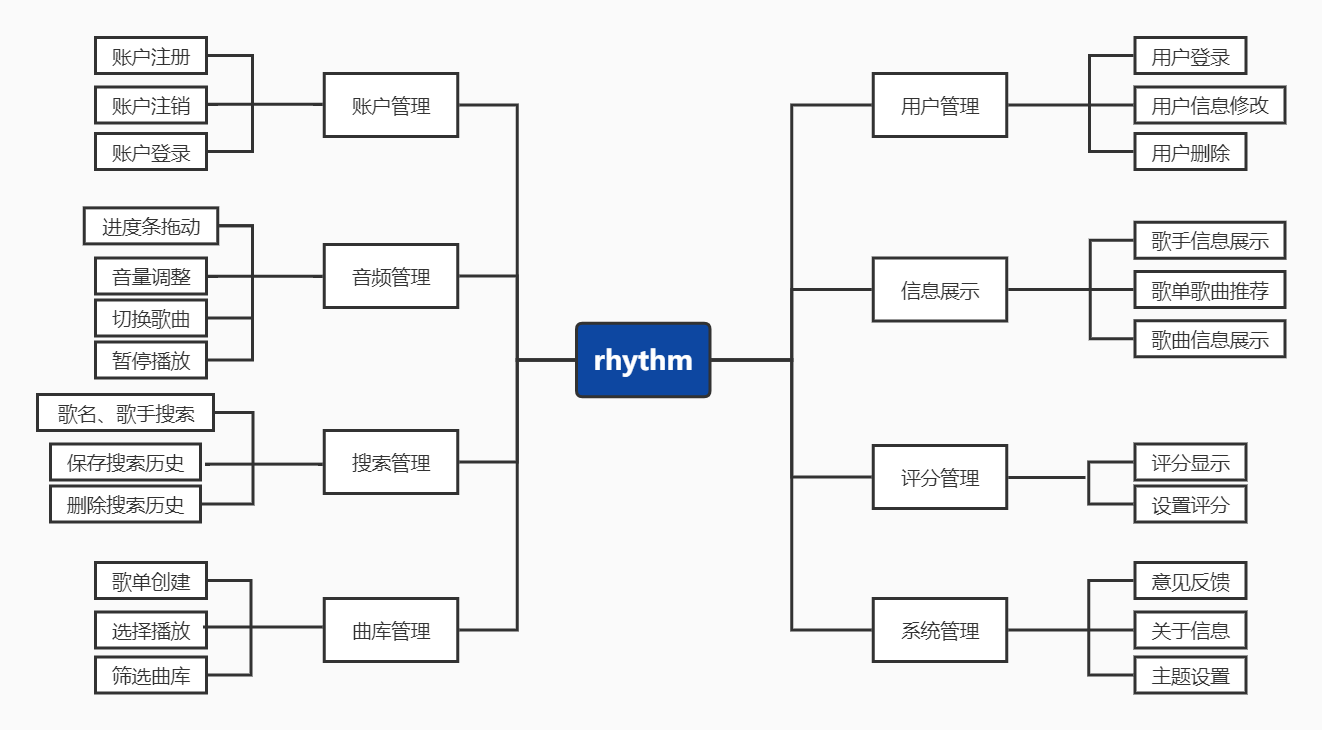
### 2.2 软件功能

#### 2.2.1 功能列表

功能简介

根据rhythm平台使用功能情况将系统主要分为以下八个功能模块：

* 账户管理
* 账户注册
* 账户注销
* 账户登录
* 音频管理
* 进度条拖动
* 音量调整
* 切换歌曲
* 暂停播放
* 搜索管理
* 歌名、歌手搜索
* 保存搜索历史
* 删除搜索历史
* 曲库管理
* 歌单创建
* 筛选曲库
* 选择播收
* 用户管理
* 用户登录
* 用户信息修改
* 用户删除
* 信息展示
* 歌手信息展示
* 歌单歌曲推荐
* 歌曲信息展示
* 评分管理
* 评分显示
* 设置评分
* 系统管理
* 意见反馈
* 关于信息
* 主题设置

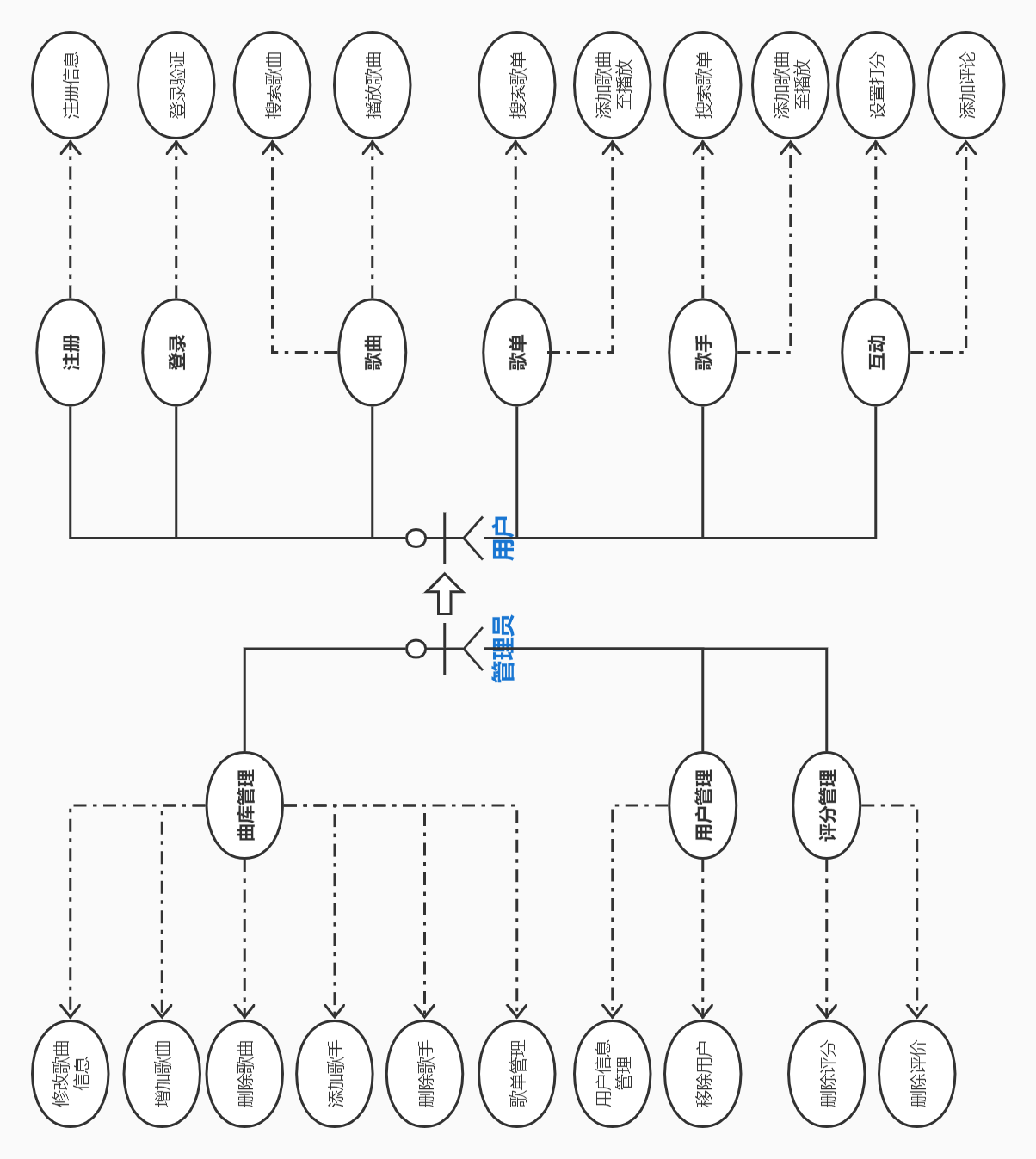


#### 2.2.2 用例图

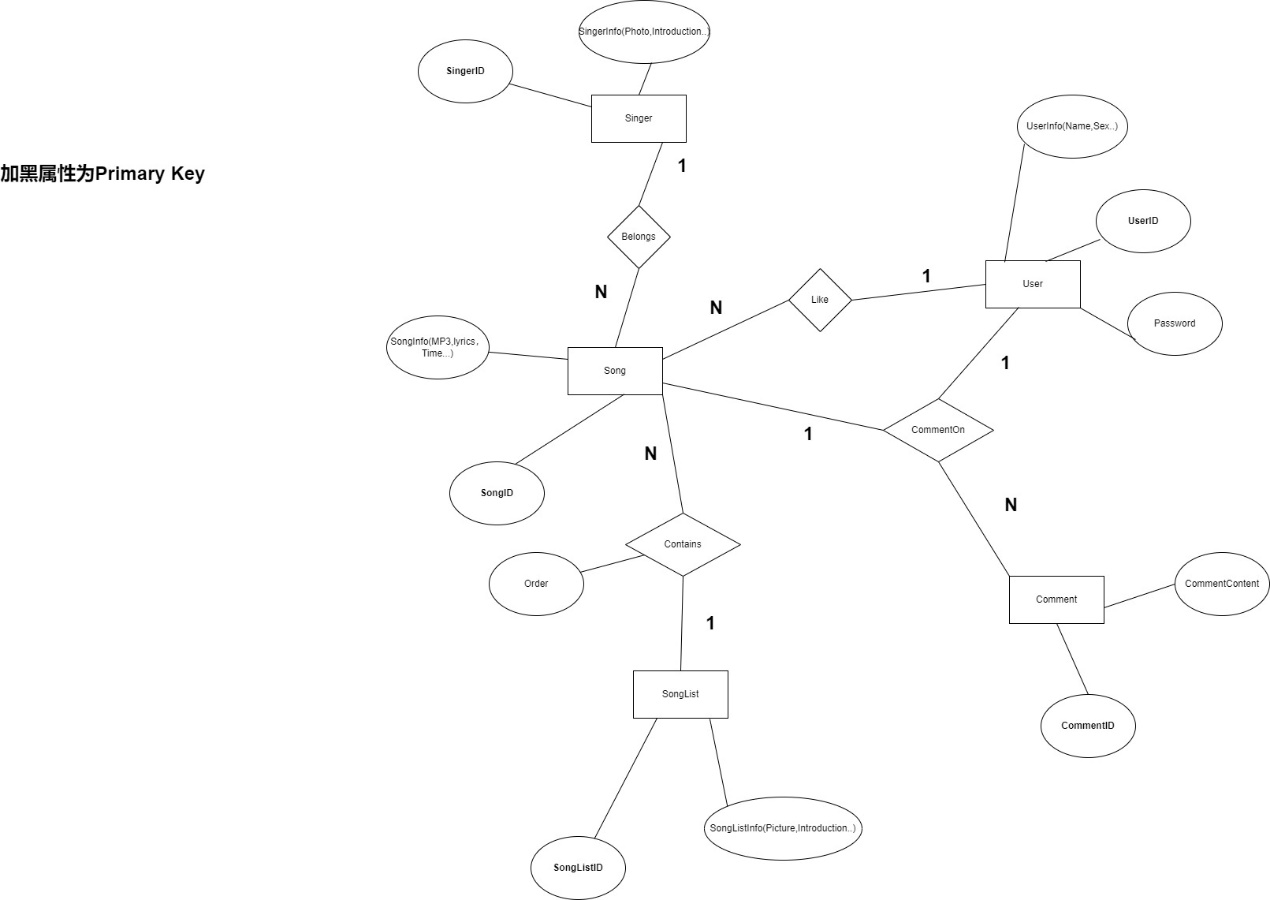
设计简介:

rhythm的主要分为两类：普通用户和管理员。

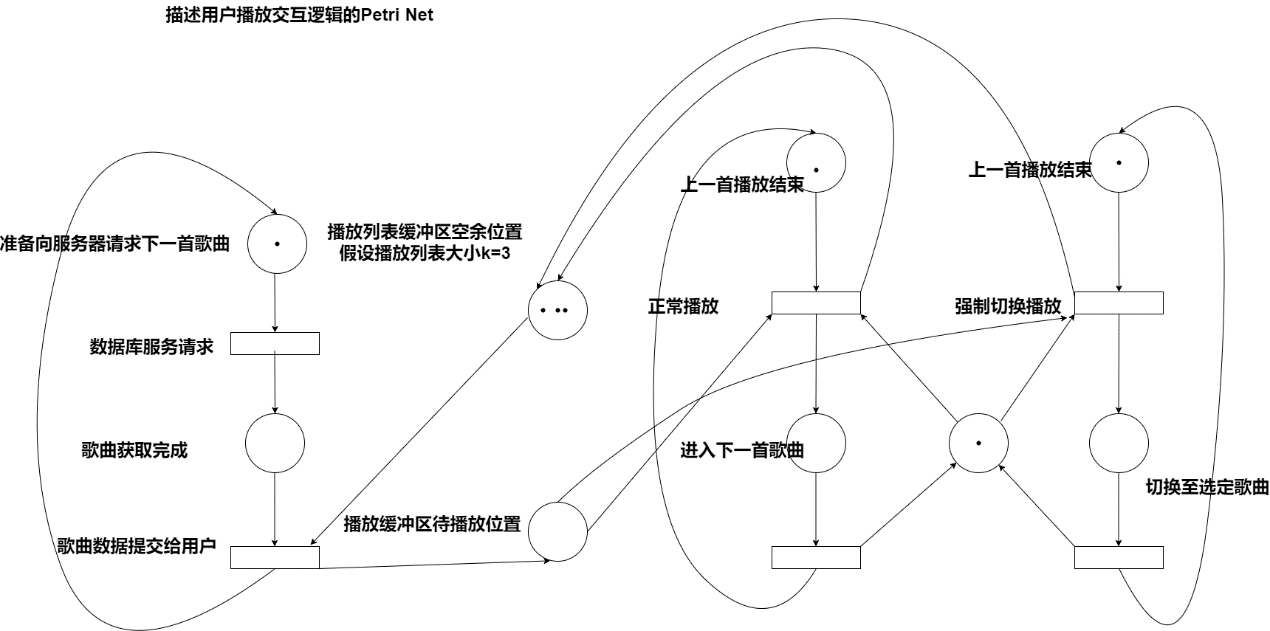
* 普通用户：
* 基本功能:
* 登录：填写登录信息，验证身份。
* 注册：填写注册信息。
* 歌曲：能对歌曲进行播放聆听和收藏歌曲到歌单、对歌曲搜索。
* 歌单：对歌单能够有搜索、评价、打分操作。
* 歌手：搜索歌手歌曲、播放歌手歌曲。
* 互动：能够评论歌曲；点赞他人的评论；对他人的评论进行回复。
* 管理员
* 管理员具有一些高权限功能:
* 歌曲管理：增加曲库中的歌曲、删除歌曲、修改歌曲的相关信息（例如歌手、歌词等）、对歌单的管理。
* 用户管理：对普通用户的信息、移除非法用户。
* 评分管理：删除用户评分、移除评论。



#### 2.2.3 复杂系统后端设计的ER图



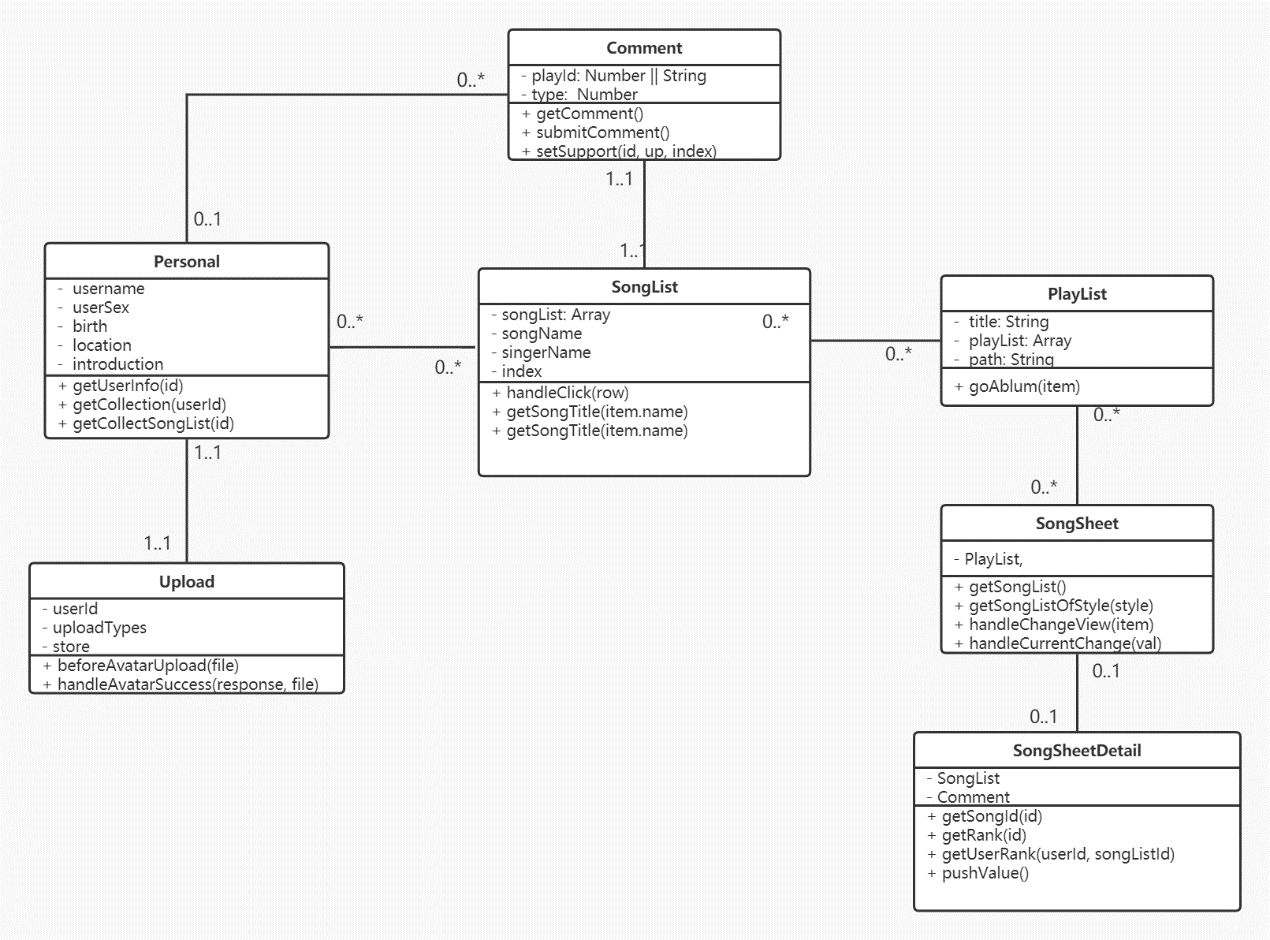
#### 2.2.4 用户播放/切换功能的交互逻辑的Petri Net描述



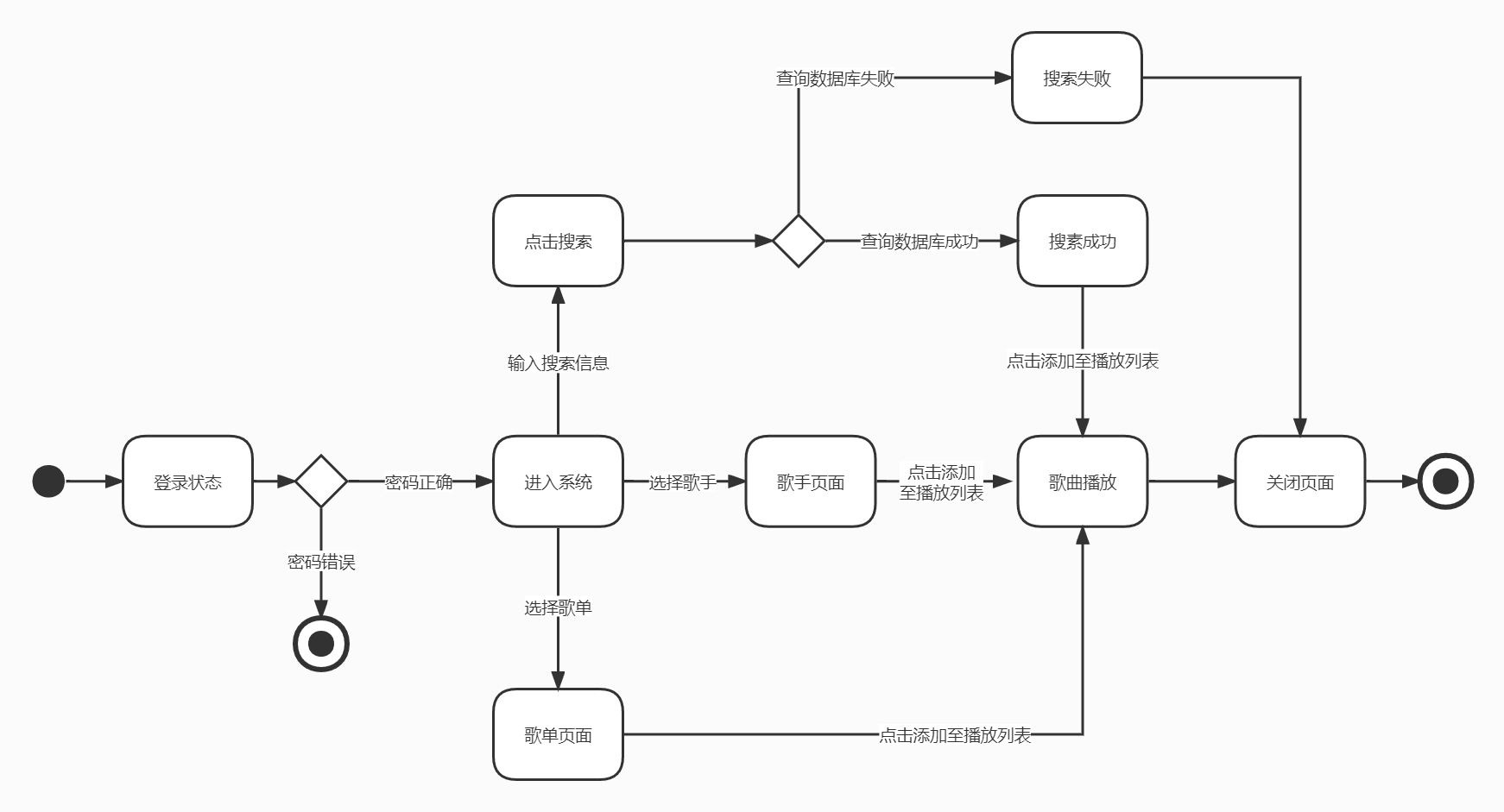
#### 2.2.5.1UML类图

基于为rhythm系统中用到的类及其类与类间的关系

UML类图中还展现类中成员变量及成员函数即每个类所进行的操作；



#### 2.5.5.2UML状态图

用户对系统进行基本操作的状态过程图，能够对平台的使用过程有更明确了解

### 2.3 用户类及特征

| **用户类** | **描述** |
| --- | --- |
| 普通用户 | 统一身份认证登录，修改个人资料，绑定/解绑用户账号，论坛操作（发帖、删帖、举报、评论、点赞、收藏），查看收藏内容，资讯操作（查看、收藏），曲库查询，搜索查询，意见反馈 |
| 平台管理员 | 管理数据库中的用户，歌手歌曲专辑，评论等数据信息。 |
| 系统管理员 | 代码维护，bug处理 |

### 2.4 假设和依赖关系

本项目是否能够成功实施，主要取决于以下的条件：

* 为了项目的开发和实施，在必要时对现有的软件开发流程进行详细的了解。
* 具有相对稳定的项目团队，不稳定的团队将影响项目的进度和质量。
* 团队成员对软件开发过程使用的性能良好的软件有较好的了解，明确其中必要的功能，为本项目提供完整的功能和性能需求资料，以便于对其进行分析，从而形成完善的软件需求。
* 团队拥有软件系统的运行必要的且能够满足系统运行条件的硬件环境和通讯环境，不合适的硬件环境和通讯环境将会影响系统的性能。
* 项目团队掌握先进的能够适用于该项目的技术，这是系统的性能是否优化和项目能否成功的保证。

## 3.系统功能

### 3.1 功能需求

#### 3.1.1 普通用户

* 注册

通过提供邮件地址，设置密码和用户名进行注册

* 注销

删除在该平台的账号信息及相关数据，不可再使用此账号登录。

* 登录

通过注册过程设置的用户民/邮箱账号，密码进行登录。非登录状态也可使用一些基本功能如歌曲播放，但无法评论，收藏歌曲等。

* 播放/暂停

选定播放或暂停当前选中歌曲

* 播放顺序与歌曲切换

通过将歌曲加入或移出’待播放’歌单，以及顺序播放，随机播放，单曲循环的三种播放模式，来确定当前歌曲播放结束后的下一首播放内容。

* 评论与评分

可对某一首歌曲进行评论，且对其他用户可见，并对专辑进行打分，并展示当前的用户平均分。

* 信息展示

在进入首页，点击进入某位歌手，某张专辑或某个歌曲的页面后，会给出相应的信息展示。

进入首页时：推荐系统会根据一定的历史记录分析推荐用户可能感兴趣的歌曲歌单，以及当前热门的歌手与歌曲。

* 音乐收藏与下载

用户可以查看自己正在借阅的图书信息和到期时间。在到期时间前用户可以选择续约一次图书。具体续约结果以山东大学图书馆系统为准。

* 音量调整

调整播放时的音量大小

* 意见反馈

对网站内容维护人员，系统开发人员提出反馈意见。

* 消息通知

定时监听是否有发送给自己的消息，有新消息要进行提醒。

#### 3.1.2 平台管理员

* 注册、登录

管理员账户唯一，由程序编写时写入数据库。必须输入正确账户密码才可登录。

* 查看编辑歌手歌曲信息

查看并编辑歌手的相关信息，以及歌手的专辑歌曲信息（即每一首歌曲都需要属于一个歌手的专辑）。

* 查看编辑歌单信息

查看并编辑歌单，该歌单可由推荐系统推送给用户。

* 查看用户信息

可查看已注册的用户相关非敏感信息，用户的收藏歌单，必要时可编辑和删除用户及相关信息。

* 数据统计

显示如用户性别比例，歌手与歌曲数据规模等有助于平台商业化发展的统计信息。

#### 3.1.3 系统管理员

* 系统维护

可通过直接修改文件，数据库内容等维护系统。

### 3.2.1描述约定

* 数字符号：阿拉伯数字（0，1,2,3,4,5,6,7,8,9）
* 区域格式：中华人民共和国
* 软件语言：简体中文

### 3.2.2设计和实现的约束

* 利用现有的前端技术（Vue3.0 + Vue-Router + Vuex + Axios + ElementPlus + Echarts）及后端（SpringBoot + Mysql），是一个不错的选择。
* 采用JS、java、HTML等语言进行开发。
* 项目开发过程中工具使用case工具（ 如Rational Rose），来支持软件开发过程中所有的技术工作及管理工作。前期我们使用前段case工具帮助应用开发的所有计划。需要的软件配置项有：
* 项目管理文档。包括：《软件项目计划》、《项目进度报告》、《项目开发总结报告》。
* 软件开发文档。包括：《需求规格说明》、《概要设计说明》、《详细设计说明》、《测试计划》、《软件测试分析报告》。
* 产品文档。包括：《用户操作手册》《演示文件》。
* 前期在数据库的建立过程中，可能出现表间关系不强导致的一些功能实现起来比较困难等问题。系统的稳定性可能也需要提高，长时间的运行也可能导致软件运行缓慢进而影响软件的工作效率。系统还可能存在一些交互和用户体验方面的问题，功能的实现也可能不那么完善。

## 4. CSCI外部接口需求

### 4.1 CSCI用户接口

本系统采用 B/S 架构，采用 Material Design 设计风格，保证各个页面的设计风格美观统一。

### 4.2 硬件接口

服务器端设置多台服务器分布式处理请求。

### 4.3 软件接口

能够调起手机浏览器，发送通知消息，访问网络和储存空间。

### 4.4 通信接口

客户端和服务器端采用 Http 协议通信。通知部分使用轮询。

## 5.其他非功能性需求

### 5.1 性能需求

* 用户提交不同请求之后，对请求的响应时间不能超过 5 秒，在此时间内将完成请求，并返回确认信息。
* 用户播放与切换的过程不能出现多次及明显卡顿。

### 5.2 安全性需求

* 权限控制

用户只能对歌曲等信息只有访问权限，平台管理员可以查看，修改或删除信息，但涉及由用户隐私及敏感信息需经过加密。

* 记录日志

本系统应该能够记录系统运行时所发生的所有错误，包括本机错误和网络错误。这些错误记录便于查找错误的原因。日志同时记录用户的关键性操作信息。

* 数据备份

定时进行数据的备份和恢复，以弥补数据的破坏和丢失。

* 重要数据加密

本系统对一些重要的数据按一定的算法进行加密，如用户口令、用户个人信息，评论信息，涉及版权的歌曲信息等。

## 6.软件质量特性

### 6.1 可靠性

要求系统能够长时间正常运行，并且支持多用户同时访问。在任何时间用户均可通过登录进入系统，进行操作。

### 6.2 易用性

具有友好易用的用户界面及简单的操作方法，与市面上大多数播放平台的操作逻辑相符合，降低用户学习成本。

### 6.3 可维护性

便于系统管理员使用简单的图形化界面对歌曲，用户等数据进行维护更新，而不必直接操作代码等复杂。

### 6.4 扩展性

主要在服务器端预留可扩展接口，便于进行推荐算法，音效处理等暂时不影响基本功能的扩展。

### 6.5 可测试性

通过Spring boot框架@test注解可以进行相对简单的单元测试，后期考虑将网站部署并邀请多为同学进行较大规模的测试，以及通过JMeter等工具进行压力测试。。

### 6.6 健壮性

系统不会产生数据的冗余、不匹配等情况。如果用户在操作过程中与系统中断，只需刷新界面，重新连接系统，系统即可恢复修改未保存的状态。

## 7合格性规定

合格性方法包括：

a、演示：运行依赖于可见的功能操作的CSCI或部分CSCI，不需要使用仪器、专用测试设备或进行事后分析；

b、测试：使用仪器或其他专用测试设备运行CSCI或部分CSCI，以便采集数据供事后分析使用；

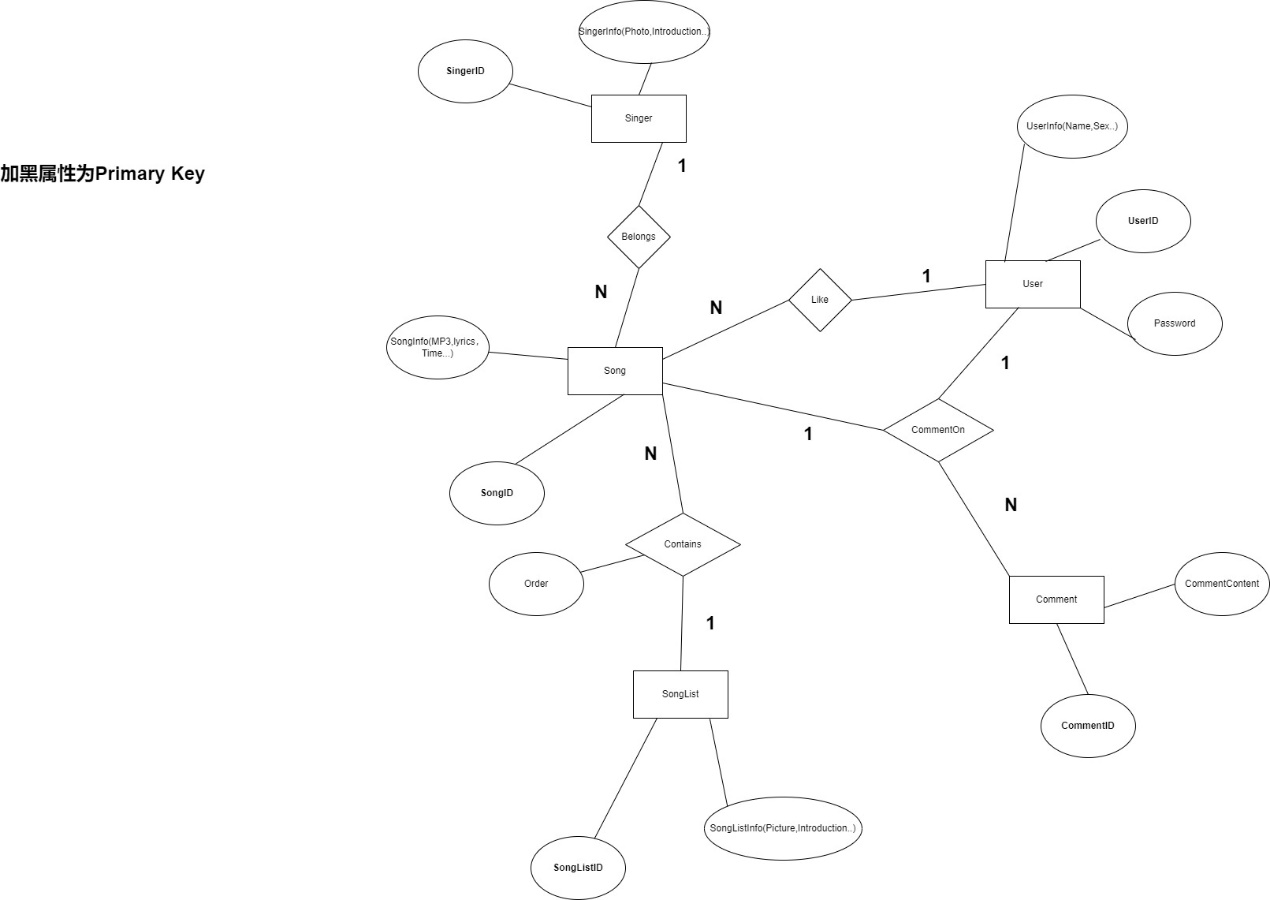
c、分析：对从其他合格性方法中获得的积累数据进行处理，例如测试结果的归约、解释或推断；

d、审查：对CSCI代码、文档等进行可视化检查；

e、特殊的合格性方法：任何应用到CSCI的特殊合格性方法，如：专用工具、技术、过程、设施、验收限制。

|  |  |  |
| --- | --- | --- |
| 需求名称 | 需求实现 | 检测方法 |
| 适应性需求 | 适应性 | b |
| 保密性需求 | 保密性 | e |
| 保密性和私密性需求 | 保密性和私密性 | e |
| CSCI环境需求 | 环境要求 | B、e |
| 计算机资源需求 | 计算机硬件 | b |
| 计算机硬件资源利用需求 |  | b |
| 计算机软件需求 |  | b |
| 计算机通信需求 |  | b |

## 8.数据库

详见具体设计说明书。这里给出ER图用于前期辅助设计：

## 9.尚未解决的问题

如何高效率的将用户特征信息的提取，歌曲特征信息的提取，推荐系统的个性化推荐结合起来，假如采用当前较为流行的在线的深度学习模型，有较大的复杂度和性能需求，而目前团队所租用的服务器暂时难以支持如此大的开销。只能将歌曲信息通过算法模型预处理，将特征向量相对静态的存储至数据库中作为单独一列使用。

## 10.注释

1. My SQL：MySQL是一个精巧的SQL数据库管理系统，而且是开源的数据管理系统，是一个真正的多用户、多线程SQL数据库服务器。MySQL 主要目标是快速、健壮和易用。
2. MFCC:一种声学/音频处理中常用的音频特征，即梅尔倒谱系数（Mel-scale FrequencyCepstral Coefficients，简称MFCC）。依据人的听觉实验结果来分析语音的频谱。
3. K-Means：一种机器学习聚类方法，这里用于歌曲风格分类的算法。

# 附录

附录可用来提供那些为便于文档维护而单独出版的信息(例如图表、分类数据)。为便于处理，附录可单独装订成册。附录应按字母顺序(A，B等)编排。