

# 系统设计说明书

---

## 一、引言

---

### 1.1 编写目的

随着科技的发展与进步，人们的日常生活逐渐走向了快节奏，而快节奏的生活容易给人们产生疲劳、压力以及不适感。人们会去寻求一些释放压力的空间去放松，其中听歌和闲聊是比较大的一部分，因此，我们希望建立一个可以与世界各地的网友一起听歌闲聊的网站来让人们拥有一个新的用于放松的途径。

### 1.2 项目背景

- 软件系统名称
  - OMCP（在线音乐聊天平台）
- 提出者
  - Tom & Jerry Team
- 开发者
  - Tom & Jerry Team
- 用户
  - All Friends

### 1.3 实现功能

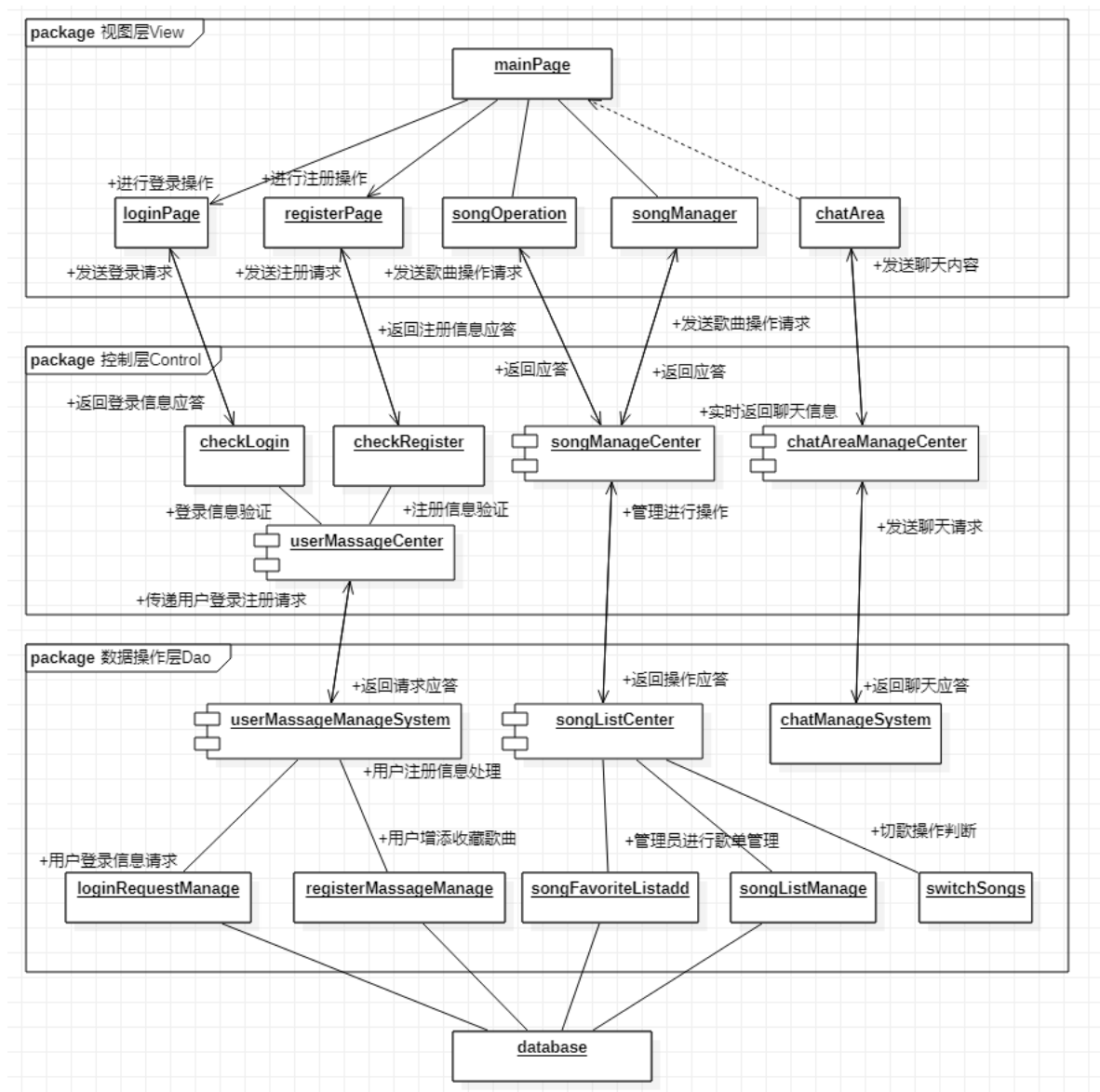
我们发现市面上已经有非常多的听歌平台，而这些平台也拥有一定的社交功能，比如网易云音乐的每一首歌都有对应的歌曲评论，大家可以一起讨论。最近的腾讯QQ手机版，新增了一起听歌的功能，它能主动点一首歌，并邀请好友同时听这一首歌。有了这两个“巨头”的思路引导，我们很快就想到，能否去做一个既可以一起听一首歌，又可以同时聊天的平台。因此，我们最终确定下来这个平台需要实现的功能：在线的网友同时听一首歌，并可以进行实时聊天。

## 二、系统总体设计

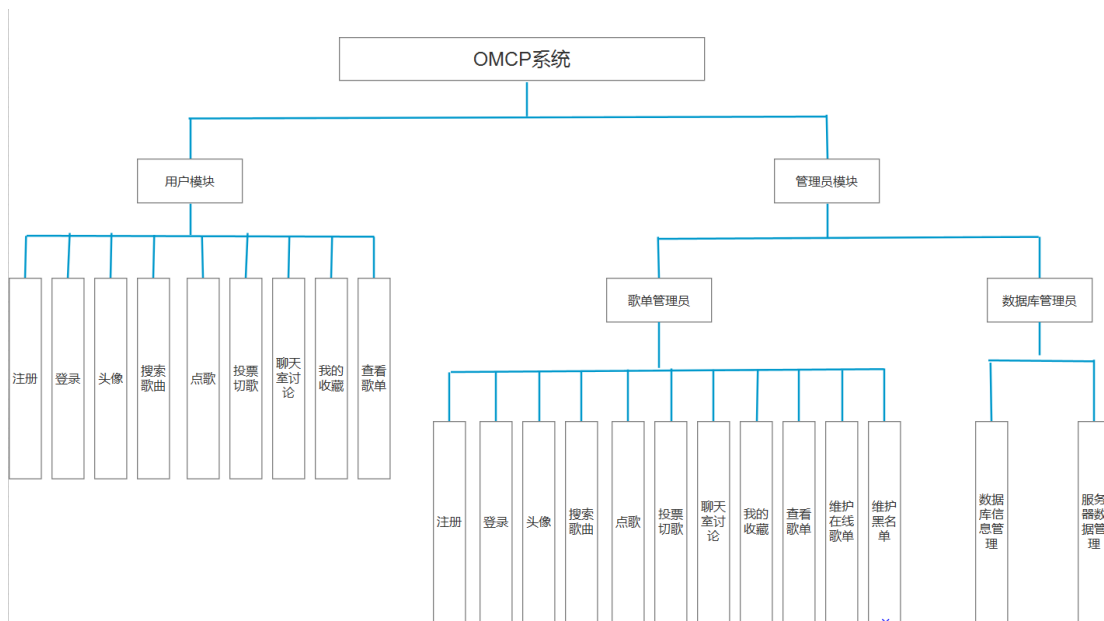
---

### 2.1 体系结构设计

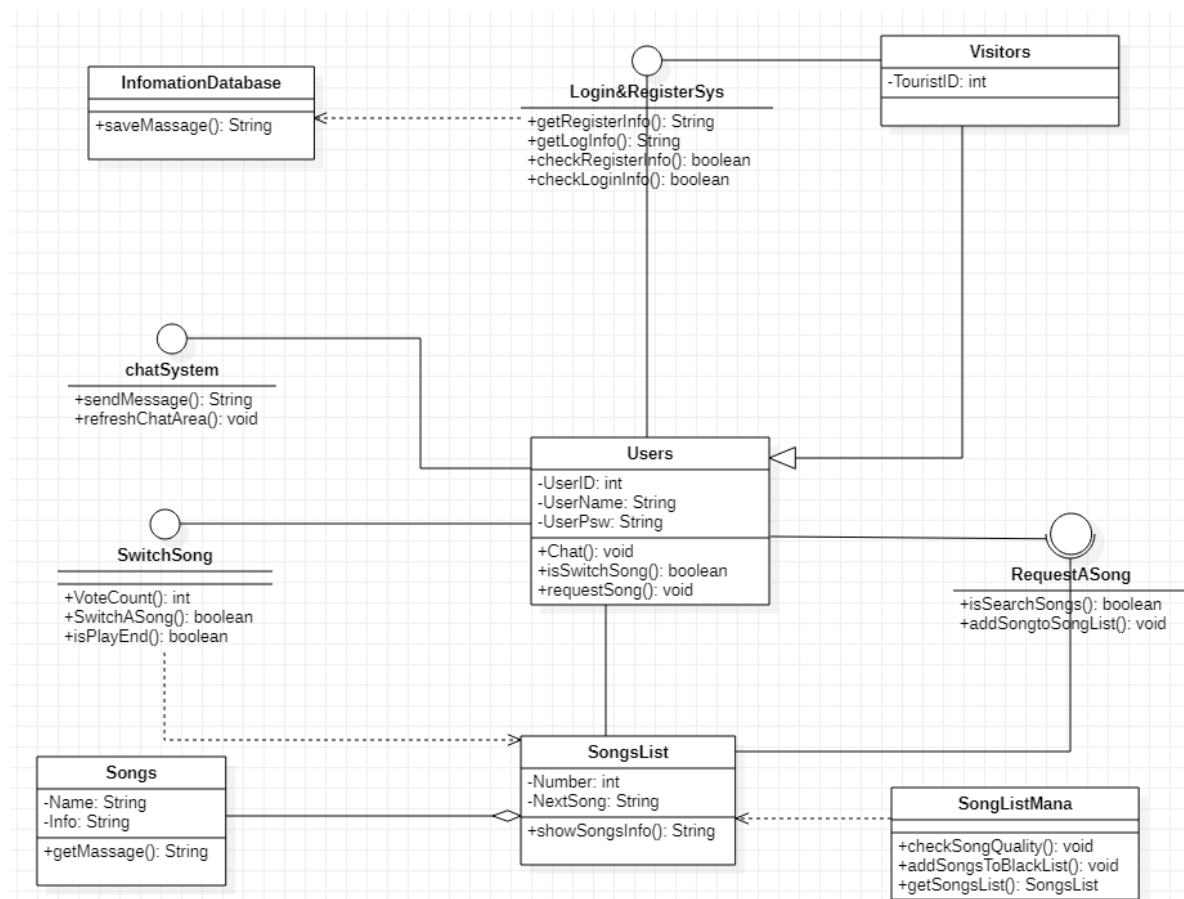
OMCP系统由用户模块和管理员模块组成，用户模块由注册、登录、头像、搜索歌曲、点歌、投票切歌、聊天室讨论，我的收藏、查看歌单 9 个功能组成，管理员模块分为歌单管理员模块和数据库管理员模块，歌单管理员拥有用户的所有功能以外还由维护在线歌单、维护黑名单功能组成；数据库管理员由数据库信息管理以及服务器数据管理两个功能组成。用户可通过个人信息功能进行修改头像、查看我的收藏和管理我的收藏。歌单管理员可通过歌单管理按钮进入歌单管理页面，进行在线歌单以及黑名单的维护。数据库管理员则是在后台进行数据库以及服务器数据的维护和管理。



## 2.2 功能模块层次图



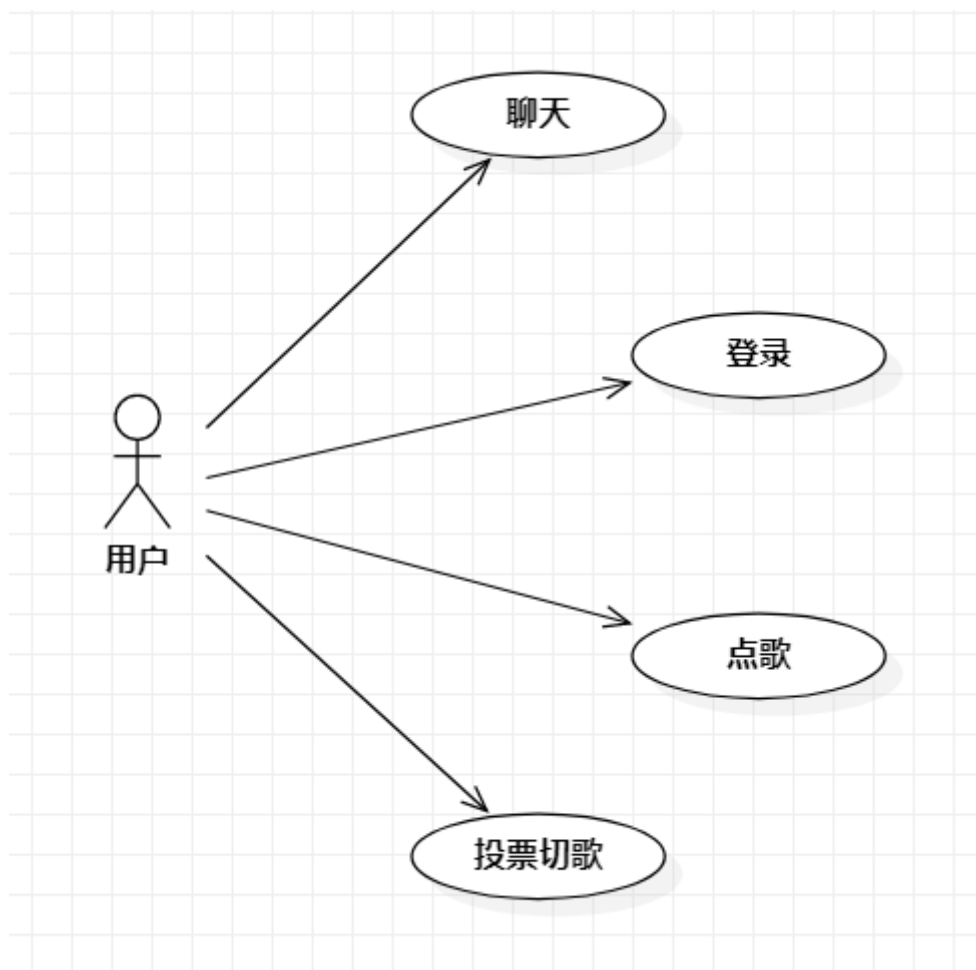
## 2.3 设计类图



## 2. 4 系统功能结构

### 2. 4. 1 用户用例图及用例描述

- 用例图



- 用例描述

- 聊天

A	B
用例名称	聊天
用例编号	001
参与者	用户，聊天系统
简要说明	用户在公共聊天室当中自由聊天
前置条件	已注册用户加入公共聊天室
基本事件流	1. 用户输入聊天内容 2. 内容显示在公共聊天室
其他事件流	无
异常事件流	输入内容不能为空
后置条件	发送成功

- 登录

1	用例名称	登录
2	用例编号	006
3	参与者	用户，数据库系统
4	简要说明	用户登录系统
5	前置条件	用户已经注册成功
6	基本事件流	用户输入信息，后台数据库系统校验用户输入的信息
7	其他事件流	无
8	异常事件流	无
9	后置条件	用户登录成功

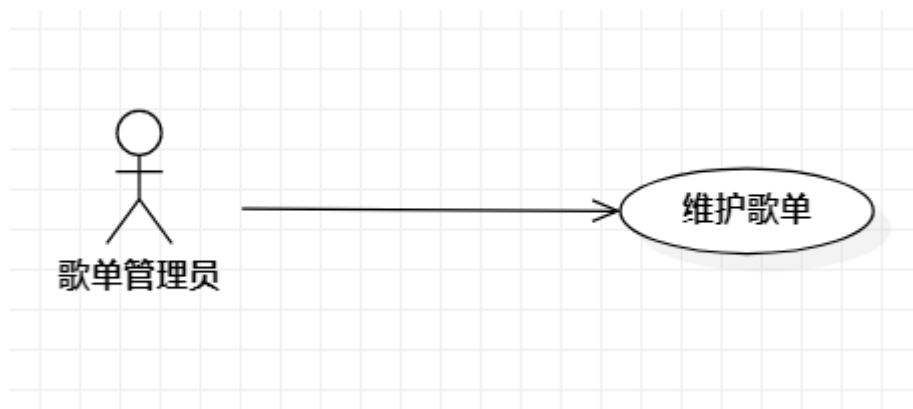
○ 投票切歌

	A	B	
1	用例名称	投票切歌	
2	用例编号	003	
3	参与者	用户，播放系统	
4	简要说明	用户发起切歌投票	
5	前置条件	已注册用户加入公共聊天室	
6	基本事件流	1. 用户发起切歌投票。 2. 每个用户参与投票后结束。 3. 播放系统根据投票数据决定是否切歌（平票不切歌）。	
7	其他事件流	无	
8	异常事件流	无	
9	后置条件	切歌成功或失败	
10			
11			

2. 4. 2 管理员用例图及用例描述

2. 4. 2. 1 歌单管理员

- 用例图

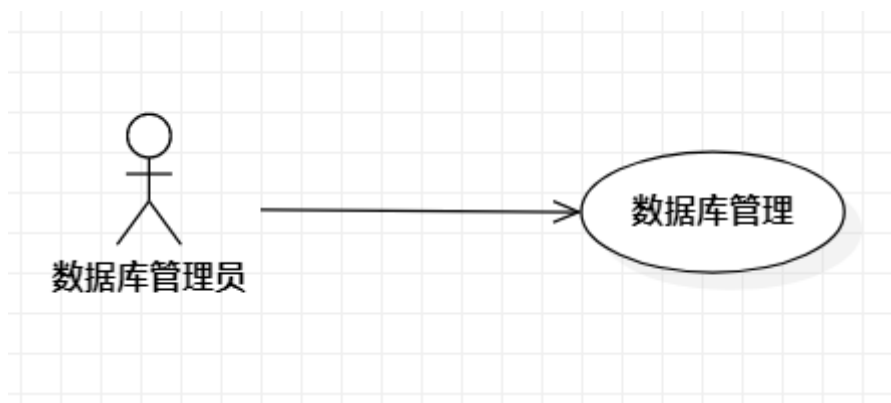


• 用例描述

	A	B
1	用例名称	维护歌单
2	用例编号	005
3	参与者	歌单管理员
4	简要说明	管理员管理歌单的质量并防止有人恶意点歌
5	前置条件	无
6	基本事件流	1. 管理员更改增删播放系统的用户点歌歌单。 2. 对于恶意点歌的歌曲进行删除。
7	其他事件流	无
8	异常事件流	无
9	后置条件	无
10		

## 2. 4. 2. 2 数据库管理员

• 用例图



• 用例描述

用例名称	数据库管理	
用例编号	005	
参与者	数据库管理员	
简要说明	数据库管理员管理用户的信息，对其进行增删改查，保存用户信息	
前置条件	游客输入注册信息	
基本事件流	对数据进行增删改查，保存数据	
其他事件流	无	
异常事件流	无	
后置条件	信息更改成功	

## 三、程序模块设计

### 3.1 用户接口模块

其中包括注册、登录、头像、搜索歌曲、点歌、投票切歌、聊天室讨论、我的收藏、查看歌单模块。

### 3.2 管理员接口模块

其中包括歌单管理员、数据库管理员两个子模块。

#### 3.2.1 歌单管理员

歌单管理员拥有用户的所有操作，还包括维护在线歌单以及维护黑名单模块；

#### 3.2.2 数据库管理员

数据库管理员拥有数据库信息管理以及服务器数据管理模块。

### 3.3 数据库设计

见《数据库设计说明书》。

### 3.4 安全性设计

#### 3.4.1 用户行为

- 尽可能实现用户的权限最小化
  - 应用用户的权限最小化，控制应用用户对文件、数据的访问，记录并统计登录历史；对重要信息资源设置敏感标记并控制对设置敏感标记资源的操作。
- 对前端输入信息进行验证
  - 将输入验证策略作为应用程序设计的核心要素。假定所有的输入都是恶意的，不要依赖于客户端的验证，虽然使用客户端验证可以减少客户端和服务器之间的信息传递次数。
  - 要做到限制、拒绝或者净化输入，输入验证的首选方法是从开始就限制允许输入的内容。按照已知的有效类型、模式和范围验证数据要比通过查找已知有害字符的数据验证方法容易。

设计应用程序时，应了解应用程序需要输入什么内容。与潜在的恶意输入相比，有效数据的范围通常是更为有限的集合。为了使防御更为彻底，可能还需要拒绝已知的有害输入，达到净化输入的效果。

- 对密码加密
  - 应用系统应对系统的使用用户密码进行加密（可以是软加密），包括密码的产生、密码录入、密码修改、密码的传输、密码的保存。软加密时应确保软加密算法具有足够的强度，并且确保密钥存储安全，对密钥的访问应严格控制。同时，还应采取必要的措施，确保软加密算法的安全。
- 提供应用系统用户的身份识别功能
  - 身份识别是信息安全服务的基础，基本原则是要做到用户区分的唯一性，认证是基于身份识别的，身份识别最常见的形式就是用户 ID，与密码组合标识一个用户身份。
- 界面的设计
  - 用户明确自己的权限，明确自己能够进行的操作，不让用户误以为拥有实际上没有的权限
- 限制第三方对加密文件的操作权限
  - 比如只允许浏览，禁止复制、剪切、截屏、修改等操作，设置好查看次数和截止打开日期等

### 3.4.2 设计行为

- 降低软件和接口的复杂性
  - 接口和软件的复杂容易导致程序出现不可预料的错误，简化接口能有效避免这种错误出现的机率
    - 单元的代码长度控制在 66 行到 132 行
    - 模块有唯一的入口和出口
    - 模块循环有正常退出条件
    - 清晰定义模块的所有输入输出并进行范围检测
    - 设计文档明确标识出所有安全性关键的设计要素
- 提高软件的健壮性
  - 接受错误输入，输出错误提示信息，软件能判断出操作员的输入操作正确性，在遇到不正确（或不合理）输入和操作时，软件拒绝该操作的执行软件提醒操作员注意错误的输入或操作软件指出错误的类型和纠正措施
  - 对输入参数进行合法性检查，对非法参数进行处理，返回错误代码
- 算法与数据管理
  - 对于规定时间内完成规定时间的模块应使用规定时间内得出结果的算法
  - 用统一的符号来表示参数、常量和标志，以便在不改变源程序逻辑的情况下，对它们进行修改
  - 文件必须唯一且用于单一目的，文件在使用前必须成功地打开，在使用结束必须成功地关闭，文件的属性应与对它的使用相一致
  - 对关键下标，使用前进行范围检查
  - 慎用易错架构：浮点数、指针、递归、中断、继承、别名、无界数组、动态内存分配，全局变量，公共数据和公共变量
- 风险隔离
  - 划分模块，防止组件之间特殊的相互作用和交叉耦合干扰，减少软件验证过程的工作量，最小化安全相关组件的规模
  - 运用信息隐蔽技术，使信息仅对有权和需要访问它的程序开放。
    - 信息隐蔽可以避错的三个理由
      - 降低了信息意外讹误的概率
      - 可以帮助在程序中建立防火墙，降低信息问题影响的范围
      - 由于信息被局部化，程序员更少地产生错误，验证人员更容易找到缺陷
    - 常见需要隔离的信息



- 安全关键的数据
  - 容易被改动的区域
  - 复杂的数据
  - 复杂的逻辑
  - 在编程语言层次上的操作
- 异常处理
  - 内部异常处理
    - 在运行阶段对于预期范围内的异常，异常处理措施应保证系统处于安全状态，并持续运行
    - 在运行阶段，对于超出预期的异常，异常处理措施最低限度应使系统转入安全状态
    - 在异常发生之后，采取措施，保证数据的完整性
    - 在异常发生之后，采取措施，保证敏感和关键数据不被泄露
  - 外部异常处理
    - 周期性检测外部输入 / 输出设备的状态，并在发生失效时转回到某个安全状态
    - 对于非法的外部中断，软件应能自动切换到安全状态