**音乐吧online网站概要设计**

**V1.0**

**三峡大学 数字媒体技术专业**

评审日期： 2019 年 5 月 10 日

**目录**

[1. 导言 1](#_Toc24785387)

[1.1 编写目的 1](#_Toc24785388)

[1.2 范围 1](#_Toc24785389)

[1.3 引用标准 1](#_Toc24785390)

[1.4 参考资料 1](#_Toc24785391)

[1.5 版本更新信息 2](#_Toc24785392)

[2. 项目设计原则简介 2](#_Toc24785393)

[3. 结构体系设计 3](#_Toc24785394)

[3.1 表现层 5](#_Toc24785395)

[3.2 控制层 7](#_Toc24785396)

[3.3 业务逻辑层 7](#_Toc24785397)

[3.4 数据持久层 8](#_Toc24785398)

[3.5 域模型层 9](#_Toc24785399)

[3.6 接口设计 9](#_Toc24785400)

[3.6.1 内部接口 9](#_Toc24785401)

[4. 功能模块设计 10](#_Toc24785402)

[4.1 功能模块设计总述 10](#_Toc24785403)

[4.2 前台基本业务模块设计 12](#_Toc24785404)

[4.2.1 模块CM1：登录系统 12](#_Toc24785405)

[4.2.2 模块CM2：注册模块 12](#_Toc24785406)

[4.2.3 模块CM3：发表动态 13](#_Toc24785407)

[4.2.4 模块CM4：评论系统 14](#_Toc24785408)

[4.2.5 模块CM5：用户搜索 15](#_Toc24785409)

[4.2.6 模块CM6：私信管理系统 16](#_Toc24785410)

[4.2.7 模块CM7：信息修改 17](#_Toc24785411)

[4.2.8 模块CM8：会员管理 18](#_Toc24785412)

[4.3 后台管理模块设计 19](#_Toc24785413)

[4.3.1 模块AM1：用户管理 19](#_Toc24785414)

[4.3.2 模块AM2：歌曲管理 20](#_Toc24785415)

[5. 数据库设计 21](#_Toc24785416)

[5.1 数据库种类特点 21](#_Toc24785417)

[5.2 数据库逻辑结构 22](#_Toc24785418)

[5.3 物理结构设计 23](#_Toc24785419)

[5.3.1 表1：Users表 24](#_Toc24785420)

[5.3.2 表2：ForumPassage表 25](#_Toc24785421)

[5.3.3 表3：Hobby表 25](#_Toc24785422)

[5.3.4 表4：Comment表 25](#_Toc24785423)

[5.3.5 表5：MusicBox表 26](#_Toc24785424)

[5.3.6 表6：Member表 26](#_Toc24785425)

[5.3.7 表7：Music表 26](#_Toc24785426)

[5.3.8 表8：PlayMusic表 27](#_Toc24785427)

[5.3.9 表9：SearchHistory表 27](#_Toc24785428)

[5.3.10 表10：Users\_has\_PlayMusic表 27](#_Toc24785429)

[5.3.11 表11：Message表 28](#_Toc24785430)

[5.3.12 表12：Manager表 28](#_Toc24785431)

[6. 界面设计 28](#_Toc24785432)

[6.1 首页设计 28](#_Toc24785433)

[6.2 登录/注册界面 29](#_Toc24785434)

[6.3 歌曲播放界面 30](#_Toc24785435)

[6.4 论坛界面 31](#_Toc24785436)

[6.5 私信界面 32](#_Toc24785437)

[6.6 歌曲上传 33](#_Toc24785438)

[6.7 个人主页 34](#_Toc24785439)

[6.8 会员界面设计 35](#_Toc24785440)

[6.9 搜索界面 36](#_Toc24785441)

[6.10 音乐播放 37](#_Toc24785442)

# 导言

## 编写目的

该文档根据音乐吧Oline网站的功能和性能，阐述了音乐吧Oline网站的概要设计，包括框架设计，功能模块设计，数据库设计，界面设计等部分。

本文档的预期读者包括：

* 设计开发人员
* 项目管理人员
* 测试人员
* 用户

## 范围

该文档的目的是解决整个项目系统怎么“怎么做的问题”。在这里，主要是根据用户提出的需求进行全面设计。

## 引用标准

[1]《SPM 课程网站概要设计》文档 北京邮电大学

## 参考资料

[1] 软件工程：实践者的研究方法（原书第八版·本科教学版）/（美）罗杰 S.普莱斯曼（Roger S.Pressman），（美）布鲁斯 R.马克西姆（Bruce R.Maxim）著；郑仁杰等译.—北京：机械工业出版社；

[2] UML2面向对象的分析与设计/谭火彬著.—北京：清华大学出版社；

[3] 人月神话（40周年中文纪念表）/(美)布鲁克斯（Brooks,F.P.）著；UML China翻译组，汪颖译.—北京：清华大学出版社。

## 版本更新信息

本文档更新记录如表1-1 所示。

表1-1 版本更新信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 001 | 2019.5.10 | 0.0 | 全部 | 初始发布版本 |
| 002 | 2019.6.1 | 1.0 | 全部 | 修改、增加 |

# 项目设计原则简介

音乐吧Oline应用程序是根据热爱听音乐，热爱创作音乐的用户的需求进行设计的音乐分享交流平台。该平台具有功能的多样性，不局限于听歌，歌曲下载，还融合了线上交友功能，可以互相关注，进行歌曲评论。也给独立创作的音乐人提供一个上传作品的平台，建立爱听音乐的人与爱创作音乐的人之间的联系。为网民提供一个轻松快捷的线上音乐分享交流平台。

在整个系统设计的过程中遵循以下设计原则：

1.实用性：实用性是系统的主要设计原则，系统设计必须最大可能地满足用户的需求，做到操作方便、界面友好、可即时更新，能适应不同层次用户的需求。

2.先进性：信息技术发展迅速，系统设计尽可能采用先进的技术标准和技术方法。

3.以用户为中心的处理：个性化服务充分体现了这一点，根据用户当前偏向爱好，配置页面功能布局及展现内容，贴合用户操作。

4.使用便捷。系统要有设计良好的人机交互界面，即使系统的操作界面简单易用，又能具有较强的适用性，满足不同计算机使用水平的用户使用。

5.灵活和易维护：采用开放的体系架构，基于开放源代码的技术框架和数据库系统，使用高效率的开源和免费开发工具，具备完整的文档说明。在维护方面，主要考虑两个层面，一是对于开发人员来讲，系统编码容易调整，可适应需求的变化和调整；二是对于系统管理维护人员来说，能够对系统进行便捷的维护和管理。

6.安全可靠：选择安全可靠的软硬件运行平台，并在系统设计和实现的时候关注系统的安全控制和执行效率，提供相应的安全防护功能，保证系统具有较高的安全性和可靠性。安全性方面，要考虑系统的安全、数据管理的安全、网络安全。保证用户权限、数据安全和系统的稳定性。

7.单一职责原则：我们系统在面向对象设计部分采取单一职责原则，其核心思想为：一个类，最好只做一件事，只有一个引起它的变化。单一职责原则可以看做是低耦合、高内聚在面向对象原则上的引申，将职责定义为引起变化的原因，以提高内聚性来减少引起变化的原因。从而最终提高我们系统的可修改性和可维护性。

# 结构体系设计

音乐吧online系统本着软件开发的设计原则，采用浏览器/服务器（B/S）的体系结构。为了满足网站后台的增删改查和平台用户创建信息与信息收发的基本操作，在软件架构上，采用五层体系结构：控制层，表现层，业务逻辑层，数据持久层，域模型层；在设计实现上，我们采用MVC的设计模式：Model 模型层、View 视图层、Controller 控制层；在体系架构上，Rose在线情缘系统选择用SSM架构。

SM（Spring+SpringMVC+MyBatis）框架集由Spring、MyBatis两个开源框架整合而成（SpringMVC是Spring中的部分内容），常作为数据源较简单的web项目的框架。其中spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架，即不再需要程序员去显式地“new”一个对象，而是让Spring框架帮你来完成这一切。SpringMVC分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。MyBatis是一个支持普通SQL查询，存储过程和高级映射的优秀持久层框架，Mybatis通过配置文件关联到各实体类的Mapper文件，Mapper文件中配置了每个类对数据库所需进行的sql语句映射。在每次与数据库交互时，通过sqlSessionFactory拿到一个sqlSession，再执行sql命令。

SSM框架关系如图3-1所示：

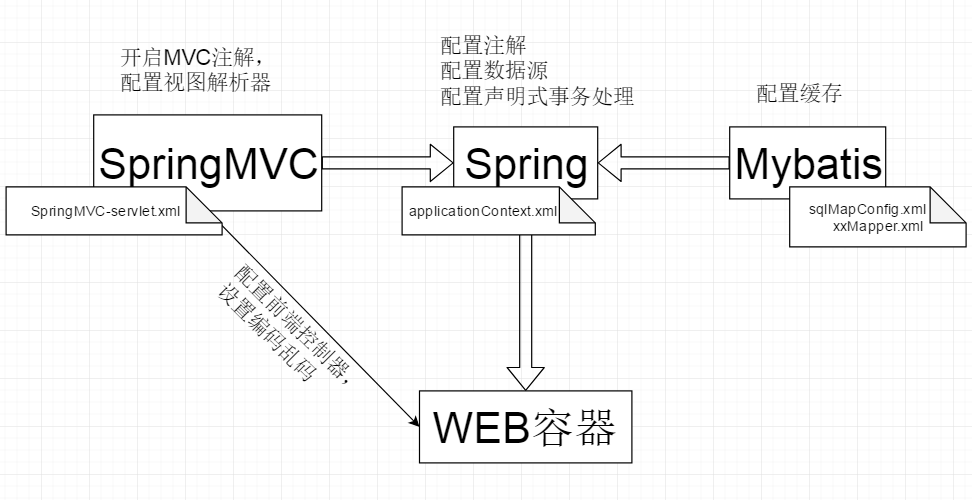


图3-1 SSM框架关系示意图

SSM架构运行流程如图3-2所示：

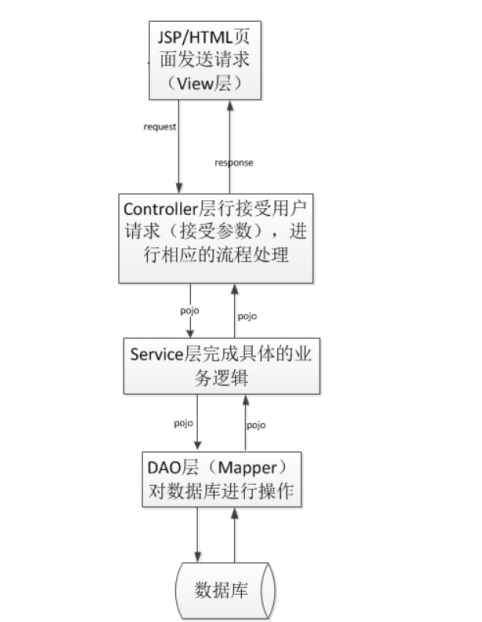


图3-2 SSM架构运行流程

如果要实现一个功能，具体流程如下：

1.创建数据库DataBase，根据需求分析中所需要的各种实体，创建对应的实体表，并给出表中的属性和约束条件。

2.写实体类entity，根据需求中需要的实体类，也即我们所创建的数据表，定义类的属性，并给出每个属性的set和get方法，主要是参照数据库中表的字段来设置，当然，也可以利用之后所学的逆向工程直接进行entity层代码生成。

3.写Mapper.xml（Mybatis），这里定义所需要实现的功能，对应要对数据库进行的那些操作，主要实现各种增删改查的工作等, 比如 insert、selectAll、selectByKey、delete、update等。

4.写Mapper.java，将Mapper.xml中的操作按照id映射成Java函数。

5.写Service.java（ServiceImpl.java），为控制层提供服务，接受控制层的参数，完成相应的功能，并返回给控制层,注意这里用到注解@Service以及@Autowired进行service层定义以及DAO层类注入。

6.写Controller.java，连接页面请求和服务层，获取页面请求的参数，通过自动装配，映射不同的URL到相应的处理函数，并获取参数，对参数进行处理，之后传给服务层，注意这里用到@Controller以及@Autowired注解。

7.写JSP页面调用，请求哪些参数，需要获取什么数据。

整体来看，在SSM架构上，结合资源调用情况，形成如下五层结构：控制层&&表现层，业务逻辑层，数据持久层，域模型层。分层结构图如图3-3所示。

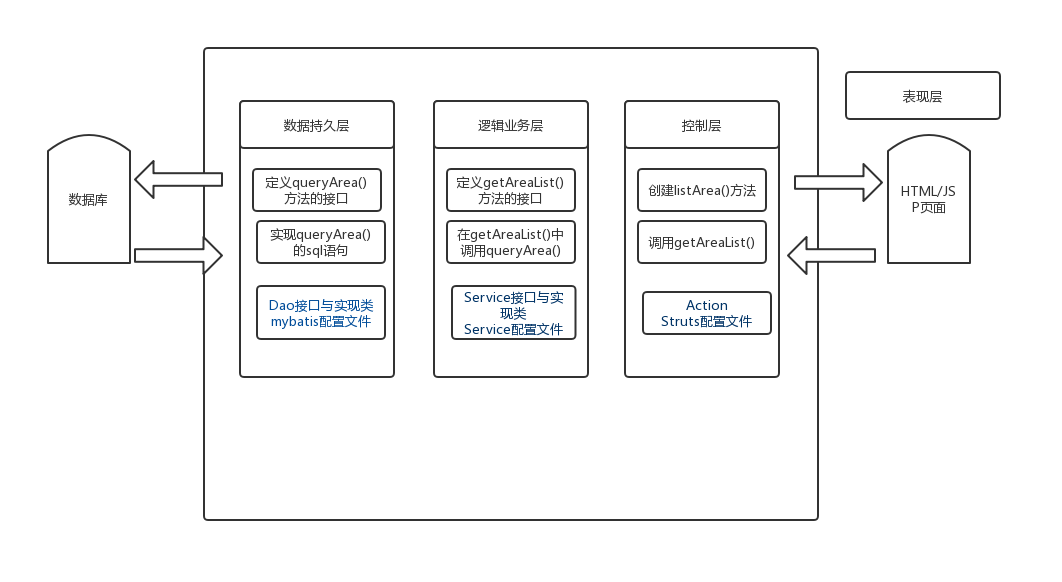


图3-3 系统分层结构示意图

## 表现层

结合用户身份判定，用于展示系统的业务信息以及接受用户输入信息。将来自用户的信息发送到对应的业务层进行处理，并接受后台处理的结果， 并结合用户身份将其返回到前端进行展示，实现系统与用户的动态交互。

表现层功能包括：

* 根据用户身份信息进行识别登录，显示不同界面。
* 根据用户身份信息进行识别实用系统，进行不同的数据和逻辑交互，满足用户需求 。
* 采用 Struts 标签库以及JSTL 标签库开发 JSP 页面。
* 通过页面控件，完成统一风格的页面部署，形成较好的页面风格。
* 采用多种展现形式，html 页面，excel 等，图片，PDF，视频播放插件等等。
* 对前端提交数据进行合规性校验，如登入时账号密码长度，类型，格式等等。
* 将页面请求发送到验证层，并接受验证层返回，将结果在前端展示。
* 提供过滤器，进行请求预处理。
* 提供文件上传功能

表现层主要由下面组件构成：

1.分发器与拦截器组件：统一处理分发来自页面的请求，通常将所有的请求分发情况都配置到 Struts 的配置文件中。分发器与拦截器包结构见表 3-1。

2.网页视图组件：根据业务逻辑层的处理数据进行页面的展示。网页视图部分样例表间表 3-2。

表 3-1 视图层包结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| Com.buptsse.spm.interceptor | MathodCacheInterceptor | 用于设置方法缓冲的拦截器 |
| Com.buptsse.spm.filter | CharacterEncodingFilter | 用于设置页面字符编码的过滤器 |
| PageEhCacheFilter | 用于设置页面缓冲的过滤器 |

表 3-2 网页视图部分样例表

|  |  |  |
| --- | --- | --- |
| 文件名称 | JSP | 说明 |
| 后台登录 | loginin.jsp | 显示后台登录界面 |
| 后台管理 | index.jsp | 显示后台管理界面 |
| 用户登录 | login.jsp | 显示用户登录界面 |
| 个人中心 | personnal.jsp | 显示个人中心界面 |
| 用户注册 | register.jsp | 显示用户注册界面 |
| 音乐搜索 | search.jsp | 显示音乐搜索界面 |
| 私信信息 | information.jsp | 显示用户会员界面 |

## 控制层

根据不同的请求触发点，接收从表现层传输的用户请求信息，并将封装好的实体对象发送到对应的业务处理单元，同时接收业务逻辑层处理结果， 指定相应的表现层 jsp 页面展现需求数据，实现页面跳转与信息现实。

控制层功能包括：

* 接收并处理从表现层传入的各种输入以及输出各种异常提示信息或处理结果信息。
* 对于输入的数据进行数据校验，过滤非法数据。
* 向业务控制层发送处理请求。
* 接收业务逻辑层请求处理结果。
* 返回状态符，实现页面跳转。

控制包部分结构见表3-3。

表 3-3 控制包部分结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| com.music.controler | MusicManagerController | 实现音乐管理页面跳转 |
| AdministratorLoginController | 实现管理员登录页面跳转 |
| ArticleManagerController | 实现文章管理页面跳转 |
| CommentsManagerController | 实现评论管理页面跳转 |
| MemberManagerController | 实现成员管理页面跳转 |

## 业务逻辑层

业务逻辑层为系统的核心层，提供了大量业务服务组件，负责处理控制层发送过来的业务数据，系统总绝大部分业务处理都在该层实现。该层实现了各种逻辑判断，即业务逻辑的封装，实现各种需求功能，并将处理后的数据传输给控制层，再由控制层进行简单处理。如果需要进行数据库操作，则调用数据持久层进行数据库相关操作。

业务逻辑层功能包含：

* 实现各种业务处理逻辑或处理算法，比如用户筛选，用户查询等等。
* 向控制层返回处理数据信息。
* 向持久层发送数据操作的请求，进行对数据信息的增删改查操作。
* 作为控制层的服务层，提供接口供周边系统调用。

业务逻辑层主要包含下面的组件：

业务逻辑处理：负责处理各类业务逻辑的 service 组成，有关页面的跳转可以从 struts 的配置文件中看到，由于系统业务逻辑层的实现类较多， 此处列举一些核心的实现类，如表 3-4 所示。

表 3-4 业务逻辑层包部分结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| com.music.service | InformationManagerService | 私信管理模块Service接口 |
| AdministratorLoginService | 管理员登录Service接口 |
| MusicManagerService | 音乐管理模块Service接口 |
| ForumManagerService | 论坛管理模块Service接口 |
| MemberManagerService | 会员管理模块Service接口 |

## 数据持久层

在 音乐吧online系统中，数据持久层负责数据的持久操作，例如和数据库交互与数据库进行连接交互。该层抽象和封装了所有对持久化存储介质的访问。数据持久层可以访问任何的数据，除了数据库，还包括缓存存数据等。在该项目框架的搭建中，是通过整合Mybatis来访问数据库。

数据持久层包含功能：

* 对数据库中的用户信息，音乐信息，会员信息等数据进行增、删、改、查操作。
* 实现事务处理，保证数据读写正常。
* 实现意外错误操作的数据恢复。

数据持久层包结构如表 3-5 所示。

表 3-5 数据持久层部分结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 文件名 | 说明 |
| com.music.online | MusicMapper | 音乐模块接口 |
| AddressMapper | 地址模块接口 |
| ManagerMapper | 管理模块接口 |
| InformationMapper | 私信模块接口 |
| FindMapper | 搜索模块接口 |
| Usrs\_has\_ActivityMapper | 过渡表接口 |
| UsrsMapper | 用户模块接口 |

## 域模型层

利用Mybatis在域模型基础之上来实现对数据的修改。

域模型层包含功能：为数据库中的表与数据实现一种数据载体，是数据库中表中数据到 Java 对象的一种映射。

域模型层结构见表 3-6。

表 3-6 域模型层部分结构表

|  |  |  |
| --- | --- | --- |
| 包名 | 类名 | 说明 |
| com.music.bean | AccountAndPassword | 管理员账户和密码的存储 |
| MusicMessage | 音乐信息表数据的存储 |
| Manager | 管理员信息的存储 |
| Member | 会员表信息的存储 |
| Msg | 把后台的数据封装返回到前端 |
| FindMusic | 搜索音乐表信息的存储 |
| Usrs | 用户表信息的存储 |
| Usrs\_has\_Forum | 过渡表信息的存储 |

## 接口设计

### 内部接口

Spring 整合 Mybatis配置：

在 MyBatis 中，SqlSessionFactory 的实例需要使用 SqlSessionFactoryBuilder 创建；而在集成环境中，则可以使用 MyBatis-Spring 整合包中的 SqlSessionFactoryBean 来代替。

SqlSessionFactoryBean 封装了使用 SqlSessionFactoryBuilder 创建 SqlSessionFactory 的过程，我们可以在 Spring 中以配置文件的形式，通过配置 SqlSessionFactoryBean 获得 SqlSessionFactory 实例。

# 功能模块设计

## 功能模块设计总述

本系统为用户交互式管理平台，主要包括得模块为两部分：前台基本业务模块和后台管理模块，如图4-1所示。其中，图4-2是前台基本业务模块图，图4-3是后台管理模块图。

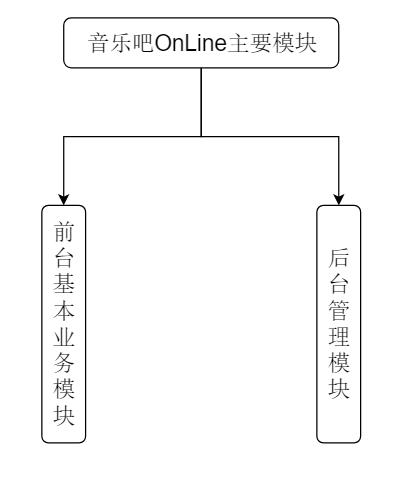


图4-1 Rose在线情缘主要模块图

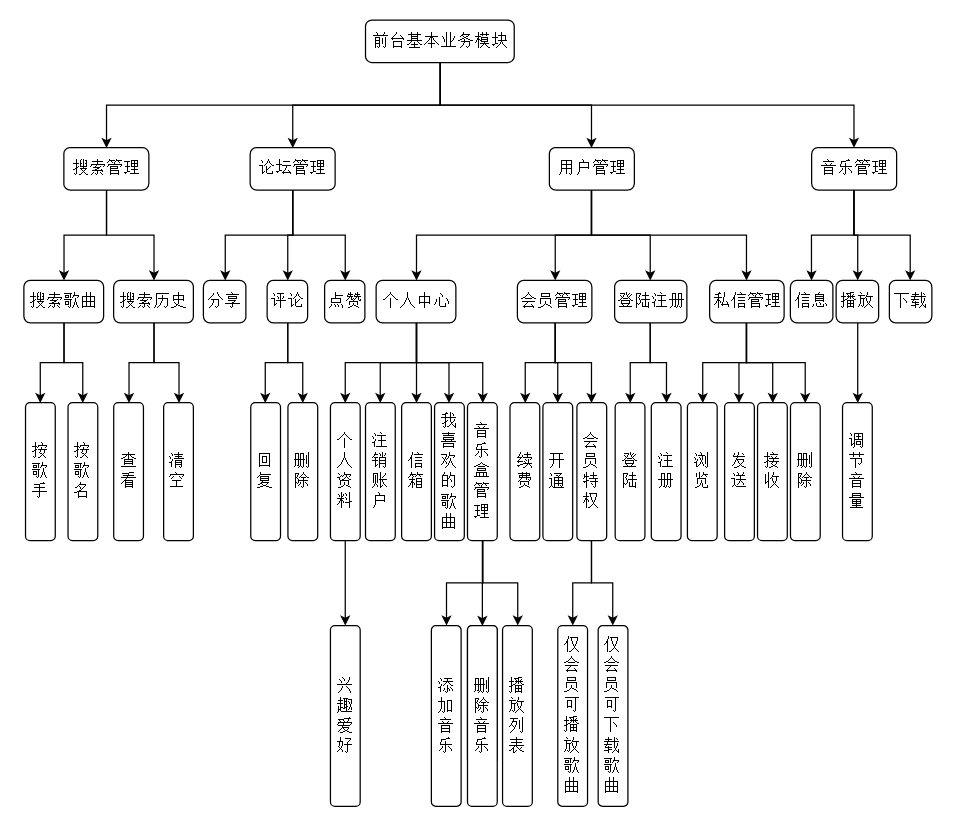


图4-2 Rose在线情缘前台基本业务模块图

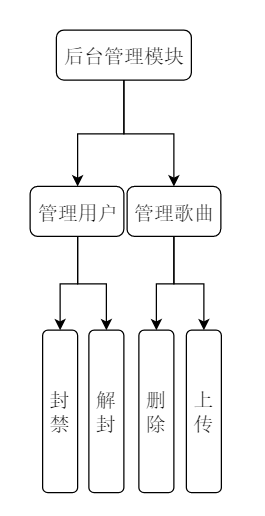


图4-3 Rose在线情缘后台管理模块图

## 前台基本业务模块设计

### 模块CM1：登录系统

**编号：**CM1；

**模块名称：**登录系统；

**功能简介：**本模块为系统登录模块，即用户登录系统的入口。在此模块中， 用户输入自己的注册邮箱和密码，系统在后台数据库进行查询操作后，返回布尔值，表示该输入是否正确，输入正确则进入系统，错误则对用户进行相应提示；

**输入：**注册邮箱，密码；

**输出：**用户是否登录成功；

**操作流程：**如图4-4所示。

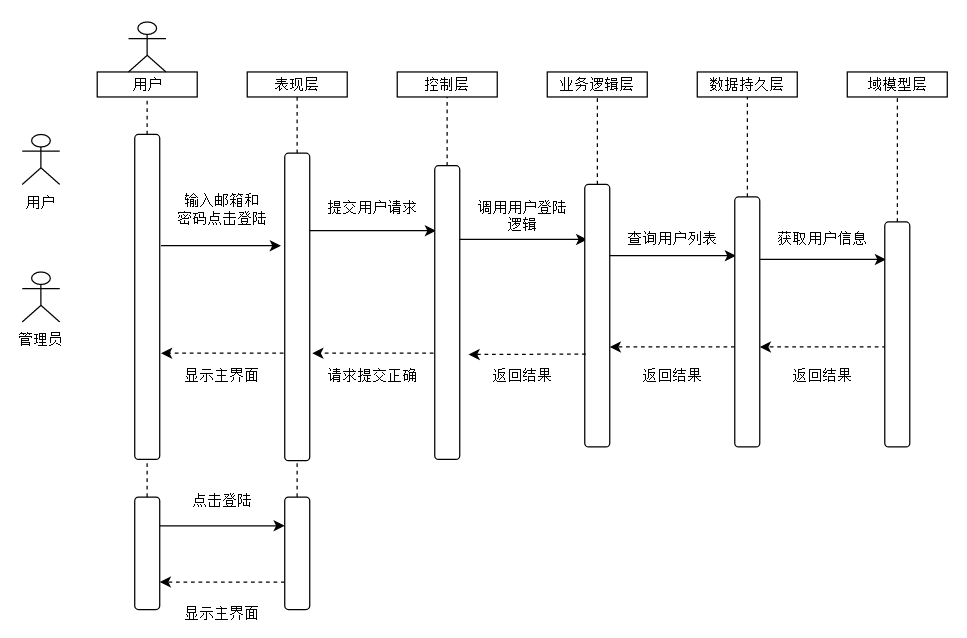


图4-4 登录系统操作流程

### 模块CM2：注册模块

**编号：**CM2；

**模块名称：**注册系统；

**功能简介：**本模块为系统注册模块。用户首次进入系统时。可通过本模块进行注册。在此模块中，系统显示注册界面，用户输入相关必要的身份信息，单击确定。若注册成功，系统将以注册邮箱作为账号，用户设定密码作为密码存入后台数据库。注册完成后，用户可使用注册成功的账号和密码登录系统；

**输入：**注册邮箱，密码，验证码等相关信息；

**输出：**用户是否注册成功；

**备注：**对于未登录系统的游客用户，系统将自动限制一部分功能的显示；

**操作流程：**如图4-5所示。

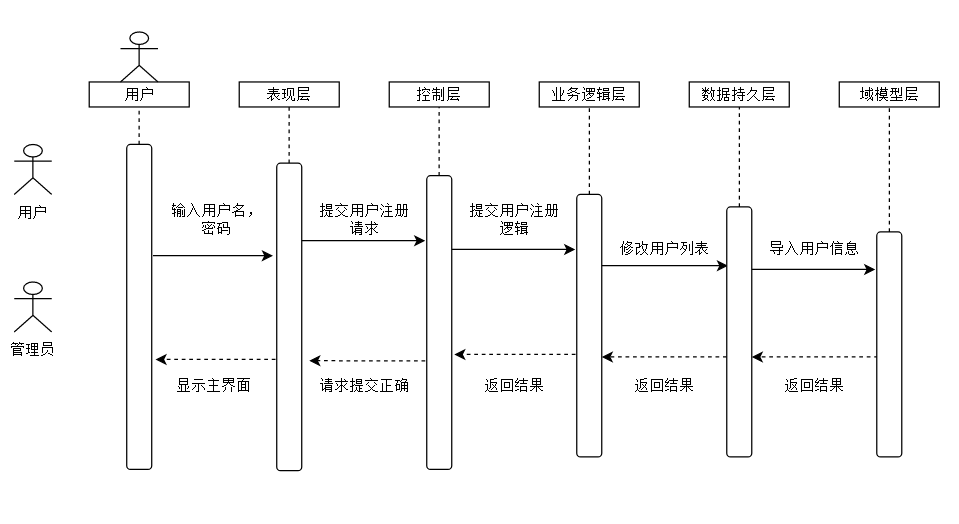


图4-5 注册系统操作流程

### 模块CM3：发表动态

**编号：**CM3；

**模块名称：**发表动态；

**功能简介：**用户点击网页的论坛板块，网页按时间排序呈现出用户（包括自己）最近发布的动态，每个动态可进行大致浏览和点击阅读；用户点击发表动态按钮，跳转发表动态页面，用户上传写好的动态内容，完毕后点击发布动态，系统更新动态列表，更新成功后动态成功显示在论坛页面供其他用户阅读；用户点击我的动态按钮，系统刷新，网页按时间排序呈现出用户自己最近发布的动态；

**输入：**文本信息；

**输出：**页面刷新；

**操作流程：**如图4-6所示。

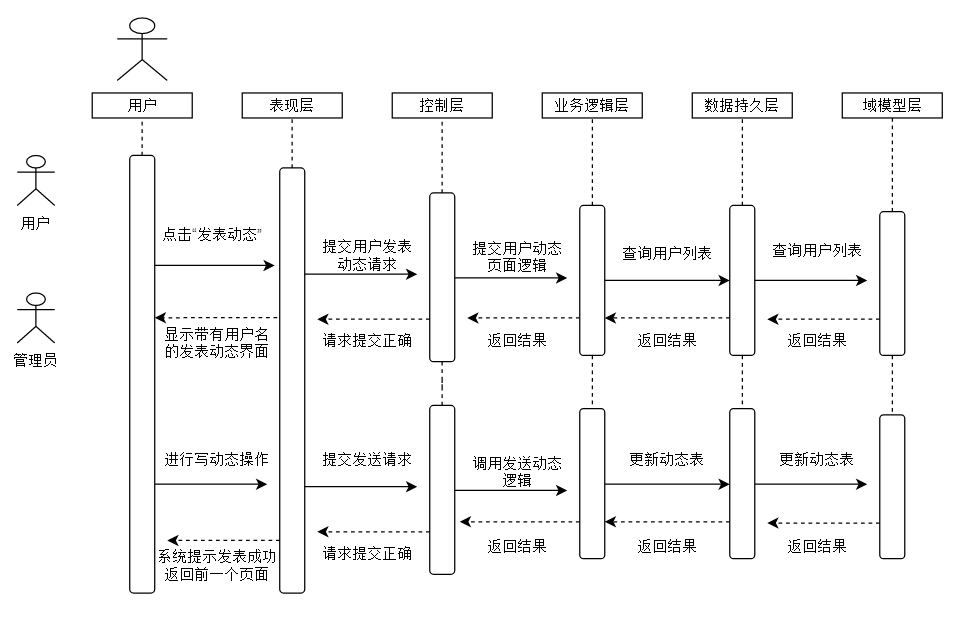


图4-6 论坛系统操作流程

### 模块CM4：评论系统

**编号：**CM4；

**模块名称：**评论系统；

**功能简介：**用户进入论坛界面，可以看到其他用户发布的动态以及动态下的评论，用户可以对其他用户发布的动态进行评论，也可以对评论进行回复，系统更新评论列表，更新成功后评论显示在论坛页面可供其他用户查看；

**输入：**文本信息；

**输出：**页面刷新；

**备注：**自己发表的评论可以进行修改和删除操作，但是不能操作他人评论

**操作流程：**如图4-7所示。

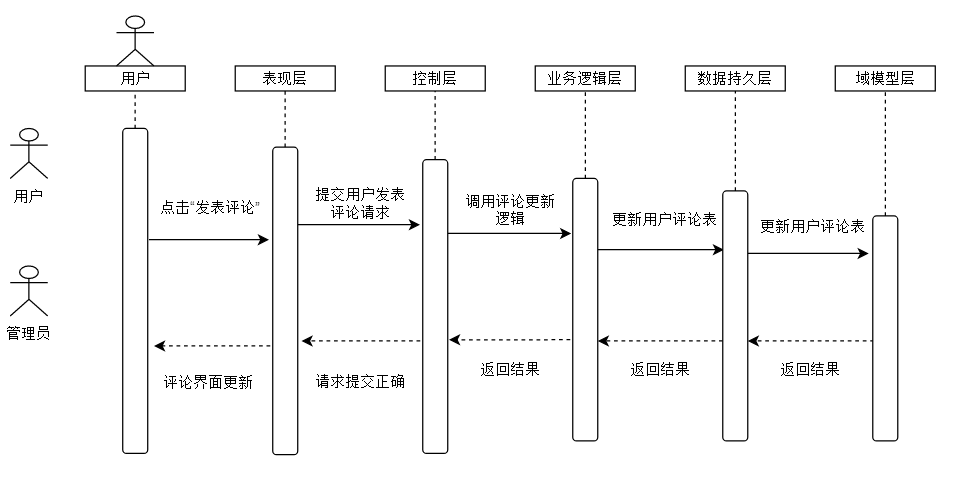


图4-7 论坛系统操作流程

### 模块CM5：用户搜索

**编号：**CM5；

**模块名称：**用户搜索；

**功能简介：**用户进入网站搜索模块，通过输入歌曲名或歌手的关键字可进行精确搜索，输入筛选条件也可进行条件搜索。点击立即搜索按钮系统将访问数据库中的歌曲表和歌手表，筛选返回符合条件的信息，响应成功后刷新界面并显示满足条件的信息；

**输入：**文本信息、鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图4-8所示。

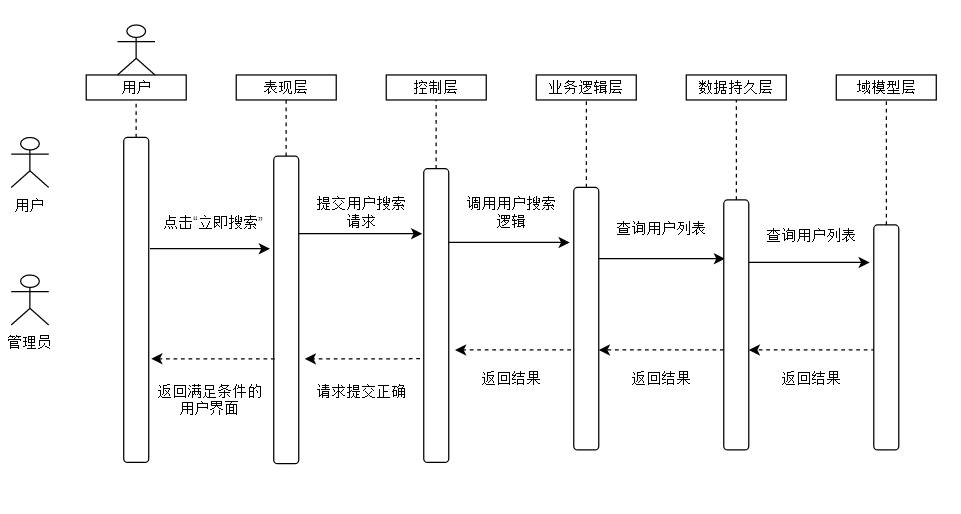


图4-8 用户搜索操作流程

### 模块CM6：私信管理

**编号：**CM6；

**模块名称：**私信管理系统；

**功能简介：**用户进入播放界面点击用户或歌手的头像，进入该用户的主页，点击发私信按钮在弹出来的对话框中输入想要发送的信息，点击发送按钮，系统将给收件人发送消息，提醒其有新的私信；

**输入：**文本信息；

**输出：**页面刷新；

**操作流程：**如图4-9所示。

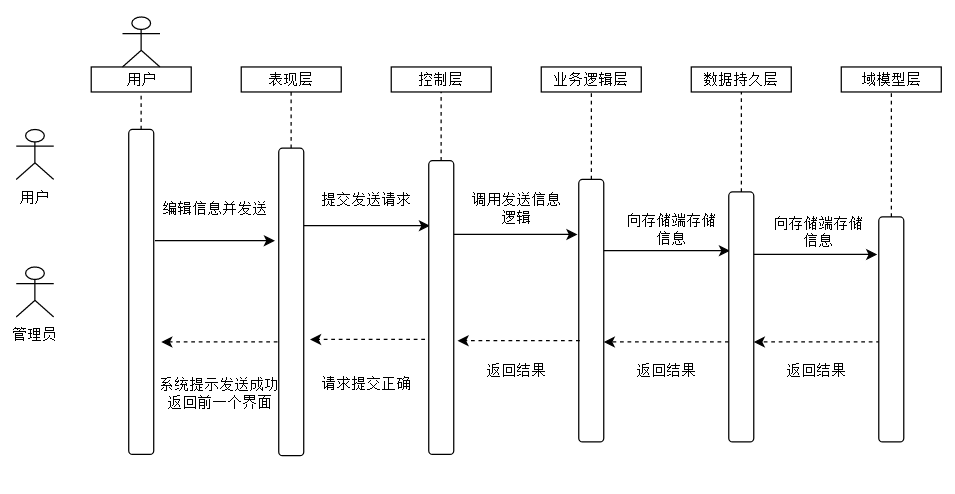


图4-9 私信管理系统操作流程

### 模块CM7：信息修改

**编号：**CM7；

**模块名称：**信息修改；

**功能简介：**用户点击网页的个人中心模块，点击个人资料选项，这里呈现的是用户的个人详细信息，鼠标点击信息选项卡可进行信息修改，修改完毕点击保存，个人信息将会存入用户信息表，更新的用户信息替换掉原来信息保存在个人中心的个人资料里；

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图4-10所示。

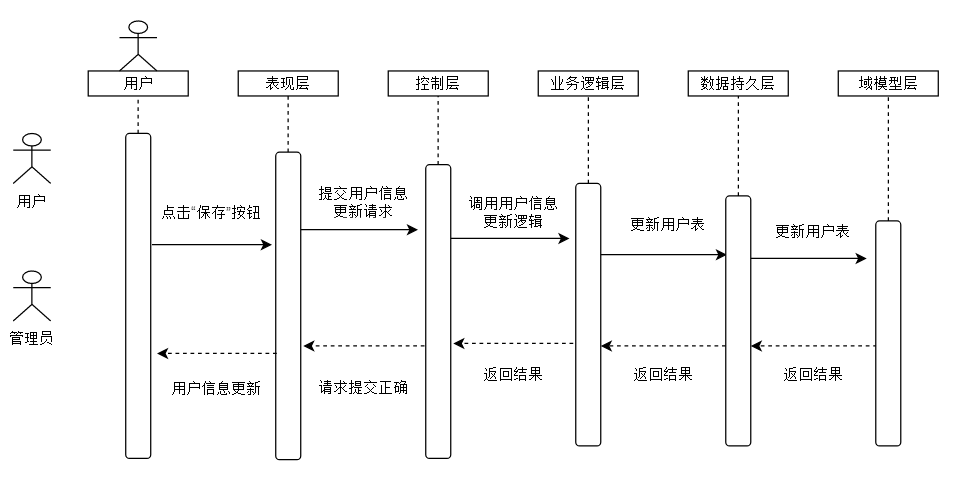


图4-10 信息修改操作流程

### 模块CM8：会员管理

**编号：**CM8；

**模块名称：**会员管理；

**功能简介：**用户进入个人中心界面，点击会员按钮进入会员界面，用户可以在该界面开通/续费会员，若开通/续费成功过，系统更新会员表，页面刷新并提示用户开通/续费成功的信息；

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图4-11所示。

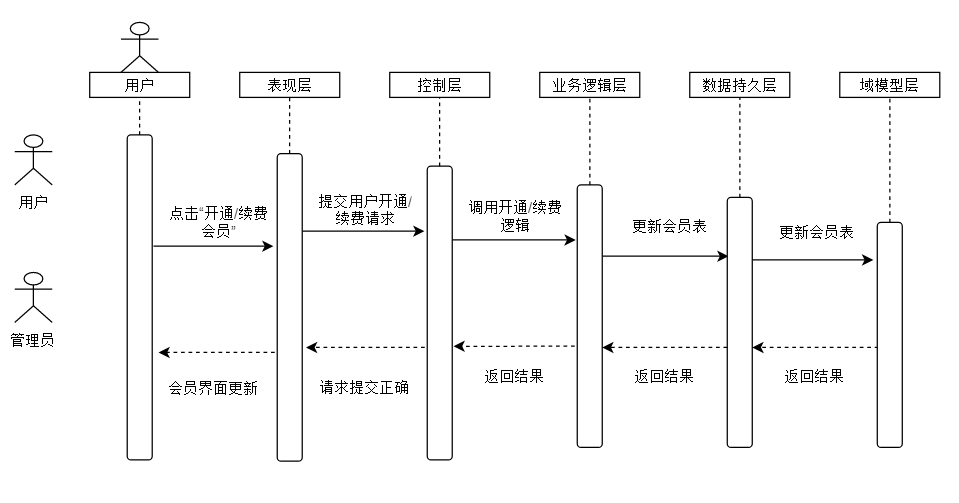


图4-11 信息修改操作流程

## 后台管理模块设计

### 模块AM1：用户管理

**编号：**AM1；

**模块名称：**用户管理；

**功能简介：**管理员进入“用户管理”模块，选择查看系统中已经存在的用户信息。系统显示全部用户信息，包括用户名，用户邮箱，用户状态， 并提供封禁，解禁，查询操作。管理员选择相应的用户进行封禁，解禁，并点击系统提示的确认操作。系统更新用户列表；

管理员输入完整用户邮箱名，选择查询操作。系统显示该用户，并提供封禁，解禁操作；

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图4-12所示。

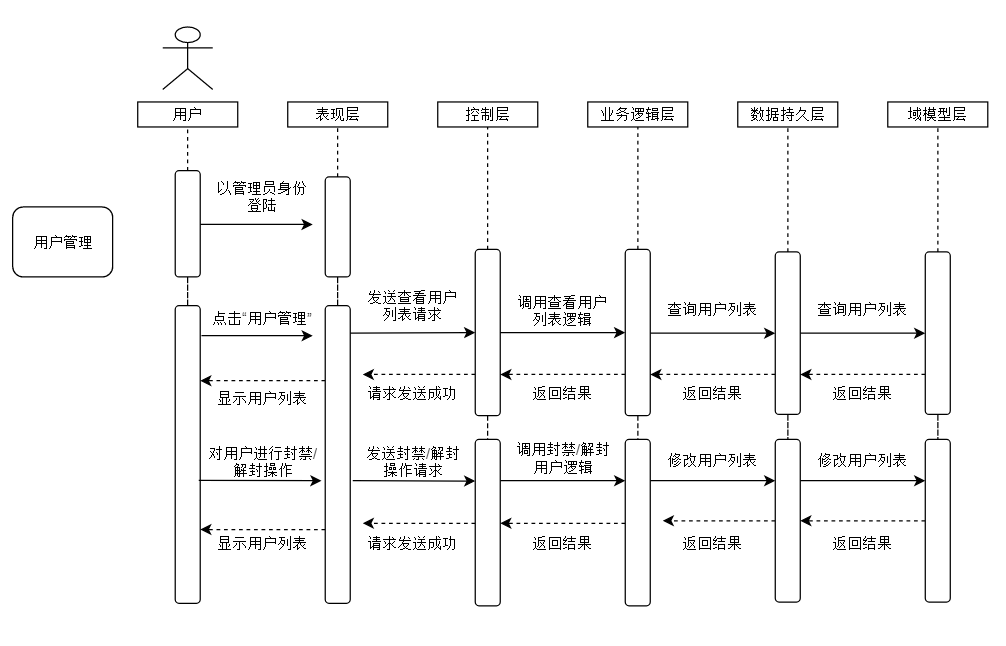


图4-12 用户管理操作流程

### 模块AM2：歌曲管理

**编号：**AM2；

**模块名称：**歌曲管理；

**功能简介：**管理员进入“歌曲管理”模块，系统显示当前系统中存在的“歌曲信息”列表，包括歌曲名称，音乐人，并提供上传和删除操作。管理员选择上传歌曲，并确认操作。系统更新歌曲信息列表。管理员选择删除歌曲，并确认操作。系统更新歌曲信息列表；

**输入：**鼠标点击事件；

**输出：**页面刷新；

**操作流程：**如图4-13所示。

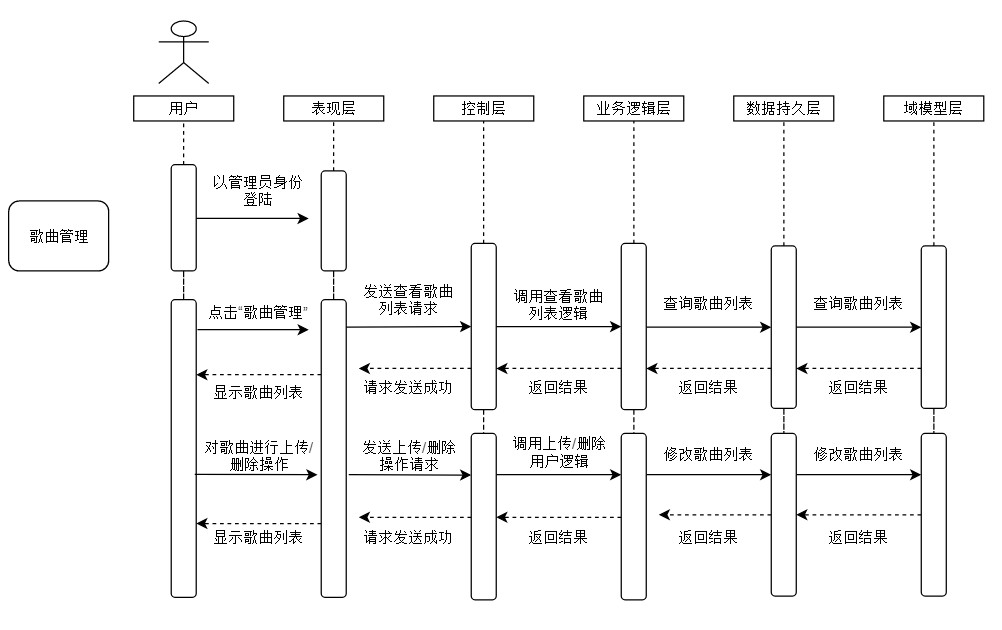


图4-13 歌曲管理操作流程

# 数据库设计

## 数据库种类特点

本项目采用了MySQL 5.7.23的数据库。MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL具有以下几个特性：

（1）MySQL为多种编程语言提供了API。这些编程语言包括C、C++、Python、Java、Perl、PHP、Eiffel、Ruby,.NET和Tcl等。

（2）优化的 SQL查询算法，有效地提高查询速度。

（3）提供多语言支持，常见的编码如中文的GB2312、BIG5，日文的 Shift\_JIS

等都可以用作数据表名和数据列名。

（4）提供 TCP/IP、ODBC和JDBC等多种数据库连接途径。

（5）提供用于管理、检查、优化数据库操作的管理工具。

（6）支持大型的数据库。可以处理拥有上千万条记录的大型数据库。

（7）MySQL使用标准的SQL数据语言形式。

## 数据库逻辑结构

经过充分的调研和分析，我们将“音乐吧online平台”的数据中设计了11个实体，其中包括Users表，Forum Passage表，Hobby表，Comment表，Music Box表，Member表，Music表，PlayMusic表，SearchHistory表，Message表，Manager表Users\_has\_PlayMusic。针对每个实体，又有其对应的属性。他们之间的关系主要包括以下几项：

* 用户表，搜索历史表，论坛文章表，兴趣爱好表独立存在，用户可以有多个搜索历史，用户可以在论坛发表多篇动态，可以有多个爱好，因此用户与爱好，论坛文章，搜索历史是一对多的关系。用户发表的动态可以有多个评论，所以文章和评论是一对多的关系。音乐盒中可以有多个音乐播放列表，所以音乐盒和播放列表是一对多的关系
* 一首音乐可以被放入多个音乐播放列表，一个音乐播放列表可以放入多首音乐，所以音乐与音乐播放列表是多对多的关系。
* 用户只有一个音乐盒，而每个音乐盒对应的用户只有一个，所以用户和音乐盒是一对一的关系。
* 用户发表的一篇文章可以有多个评论，所以文章和评论是一对多的关系。
* 管理员表是独立存在的，它负责管理员登录后台对网站进行管理。

综上分析，音乐吧online平台前端的实体关系图（概念数据模型）如下图5-1所示。

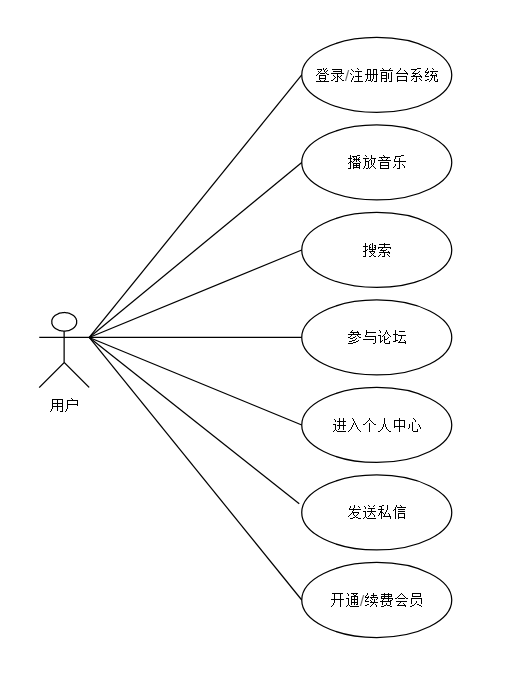


图5-1 前端实体关系图

## 物理结构设计

根据上面的实体关系，设计数据库表以及根据数据模型图5-2可知，系统一共应有12张表。下面是关于表的详细说明。

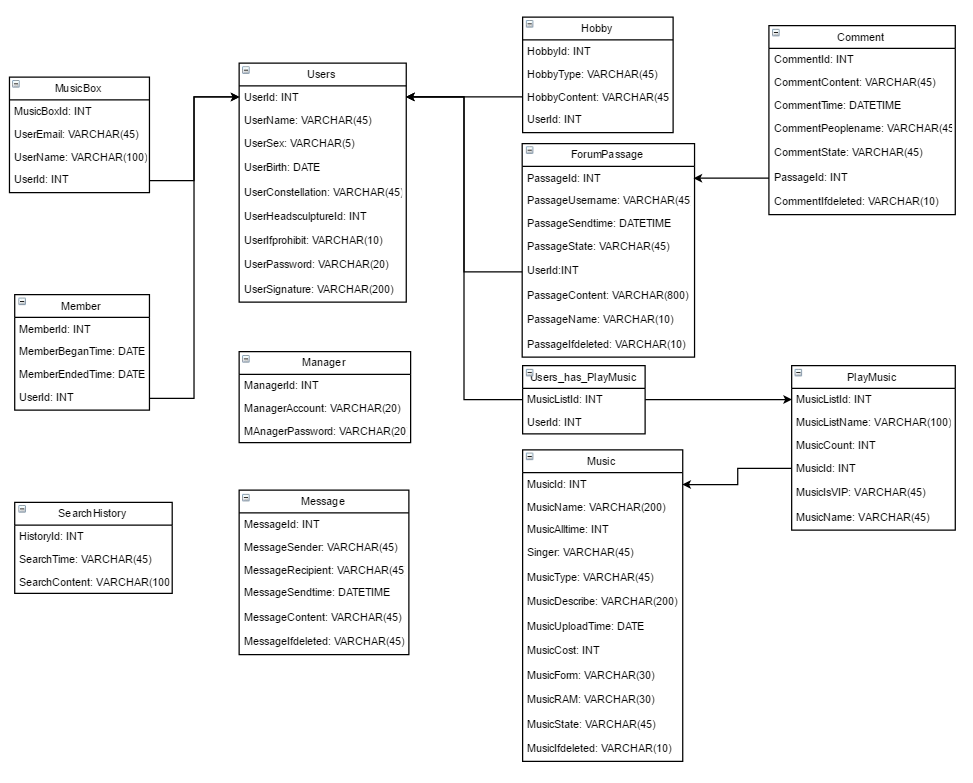


图5-2 数据库表的关系图

### 表1：Users表

用户表标识用户信息，其表结构如下：

表 5-1 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Users表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | UserId | Int(11) | N | 主键 |
| 用户名 | UserName | Varchar(45) | N |  |
| 用户性别 | UserSex | Varchar(5) | N |  |
| 用户出生日期 | UserBirth | Date | Y |  |
| 用户星座 | UserConstellation | Varchar(45) | Y |  |
| 用户头像标号 | UserHeadsculptureId | Int(11) | Y |  |
| 用户是否被封禁 | UserIfprohibit | Varchar(10) | Y |  |
| 用户密码 | UserPassword | Varchar(20) | N |  |
| 用户个人说明 | UserSignature | Varchar(200) | Y |  |

### 表2：ForumPassage表

论坛文章表标识的是用户发表音乐相关评论动态的信息，其表结构如下：

表 5-2 论坛文章表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ForumPassage表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | PassageId | Int(11) | N | 主键 |
| 动态作者 | PassageUsrname | Varchar(45) | N |  |
| 动态发表时间 | PassageSendtime | Datetime | N |  |
| 动态状态 | PassageState | Varchar(45) | Y |  |
| 作者邮箱 | UserEmail | Varchar(45) | N | 外键 |
| 动态内容 | PassageContent | Varchar(800) | N |  |
| 动态标题 | PassageName | Varchar(10) | N |  |
| 动态是否被删除 | PassageIfdeleted | Varchar(10) | Y |  |

### 表3：Hobby表

兴趣爱好表标识的是用户个人的兴趣爱好，其表结构如下：

表 5-3 兴趣爱好表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hobby表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | HobbyId | Int(11) | N | 主键 |
| 爱好类型 | HobbyType | Varchar(45) | N |  |
| 爱好内容 | HobbyContent | Varchar(45) | N |  |
| 用户邮箱 | UserEmail | Varchar(45) | N | 外键 |

### 表4：Comment表

评论表标识的是用户对音乐文章动态进行评论的信息，其表结构如下：

表 5-4 评论表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comment表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | CommentId | Int(11) | N | 主键 |
| 评论内容 | CommentContent | Varchar(45) | N |  |
| 评论时间 | CommentTime | Datetime | N |  |
| 评论用户名字 | CommentPeoplename | Varchar(45) | Y |  |
| 评论状态 | CommentState | Varchar(45) | N |  |
| 被评论动态标号 | PassageId | Int(11) | N | 外键 |
| 评论是否被删除 | CommentIfdeleted | Varchar(10) | N |  |

### 表5：MusicBox表

音乐盒表标识的是用户对应的音乐盒的信息，其表结构如下：

表 5-5 音乐盒表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MusicBox表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | MusicBoxId | Int(11) | N | 主键 |
| 用户邮箱 | UserEmail | Varchar(45) | Y |  |
| 用户名字 | UserName | Varchar(100) | Y |  |
| 用户标号 | UserId | Datetime | N | 外键 |

### 表6：Member表

会员表标识的是用户是否为会员、会员相关的信息，其表结构如下：

表 5-6 会员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Member表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | MemberId | Int(11) | N | 主键 |
| 成为会员时间 | MemberBeganTime | date | N |  |
| 会员到期时间 | MemberEndedTime | date | N |  |
| 用户Id | UserId | Int(11) | N | 外键 |

### 表7：Music表

音乐表标识的是网站中所有歌曲的信息，其表结构如下：

表 5-7 歌曲表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Music表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | MusicId | Int(11) | N | 主键 |
| 音乐名称 | MusicName | Varchar(200) | N |  |
| 音乐时长 | MusicAlltime | Int(11) | N |  |
| 演唱歌手 | Singer | Varchar(45) | N |  |
| 音乐类型 | MusicType | Varchar(45) | N |  |
| 音乐描述 | MusicDescribe | Varchar(200) | Y |  |
| 音乐上传时间 | MusicUploadTime | Date | N |  |
| 音乐是否为会员歌曲 | MusicCost | Int(11) | N |  |
| 音乐格式 | MusicForm | Varchar(30) | N |  |
| 音乐大小 | MusicRAM | Varchar(30) | N |  |
| 音乐状态 | MusicState | Varchar(45) | N |  |
| 音乐是否被删除 | MusicIfdeleted | Varchar(10) | N |  |

### 表8：PlayMusic表

音乐播放列表标识的是歌单的名字，歌单中歌曲的名字、数量等，其表结构如下：

表 5-8 音乐播放表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PlayMusic表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | MusicListId | Int(11) | N | 主键 |
| 列表名称 | MusicListName | Varchar(100) | N |  |
| 歌曲数量 | MusicCount | Int(11) | N |  |
| 歌曲ID | MusicId | Varchar(100) | N | 外键 |
| 歌曲是否为会员歌曲 | MusicIsVIP | Varchar(45) | N |  |
| 歌曲名称 | MusicName | Varchar(45) | N |  |

### 表9：SearchHistory表

搜索历史表标识的是用户在搜索界面搜索歌曲的信息，其表结构如下：

表 5-9 搜索历史表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SearchHistory表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | HistoryId | Int(11) | N | 主键 |
| 搜索时间 | SearchTime | Varchar(45) | N |  |
| 搜索内容 | SearchContent | Varchar(100) | N |  |

### 表10：Users\_has\_PlayMusic表

过渡表标识的是音乐播放列表和用户多对多之间的过渡信息，其表结构如下：

表 5-10 过渡表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usrs\_has\_PlayMusic表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 列表标号 | MusicListId | Int(11) | N | 联合主键 |
| 用户标号 | UsrId | Int(11) | N | 联合主键 |

### 表11：Message表

信息表标识的是所有网站内私信的信息，其表结构如下：

表 5-11 私信表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Message表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | MessageId | Int(11) | N | 主键 |
| 信息发送人 | MessageSender | Varchar(45) | N | 外键 |
| 信息收件人 | MessageRecipient | Varchar(45) | N | 外键 |
| 信息发送时间 | MessageSendtime | Datetime | N |  |
| 信息内容 | MessageContent | Varchar(45) | N |  |
| 信息是否被删除 | MessageIfdeleted | Varchar(45) | Y |  |

### 表12：Manager表

管理员表标识的是后台管理员的登录信息，其表结构如下：

表 5-12 管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Manager表 | | | | |
| **字段名** | **字段代码** | **字段类型** | **可否为空** | **备注** |
| 自增主键 | ManagerId | Int(11) | N | 主键 |
| 管理员账号 | ManagerAccount | Varchar(20) | Y |  |
| 管理员密码 | ManagerPassword | Varchar(20) | Y |  |

# 界面设计

## 首页设计

平台首页采用图6-1中的格式设计。网页顶部为“音乐吧Oline”的图标。图标右方为搜索栏，登录/注册，私信。

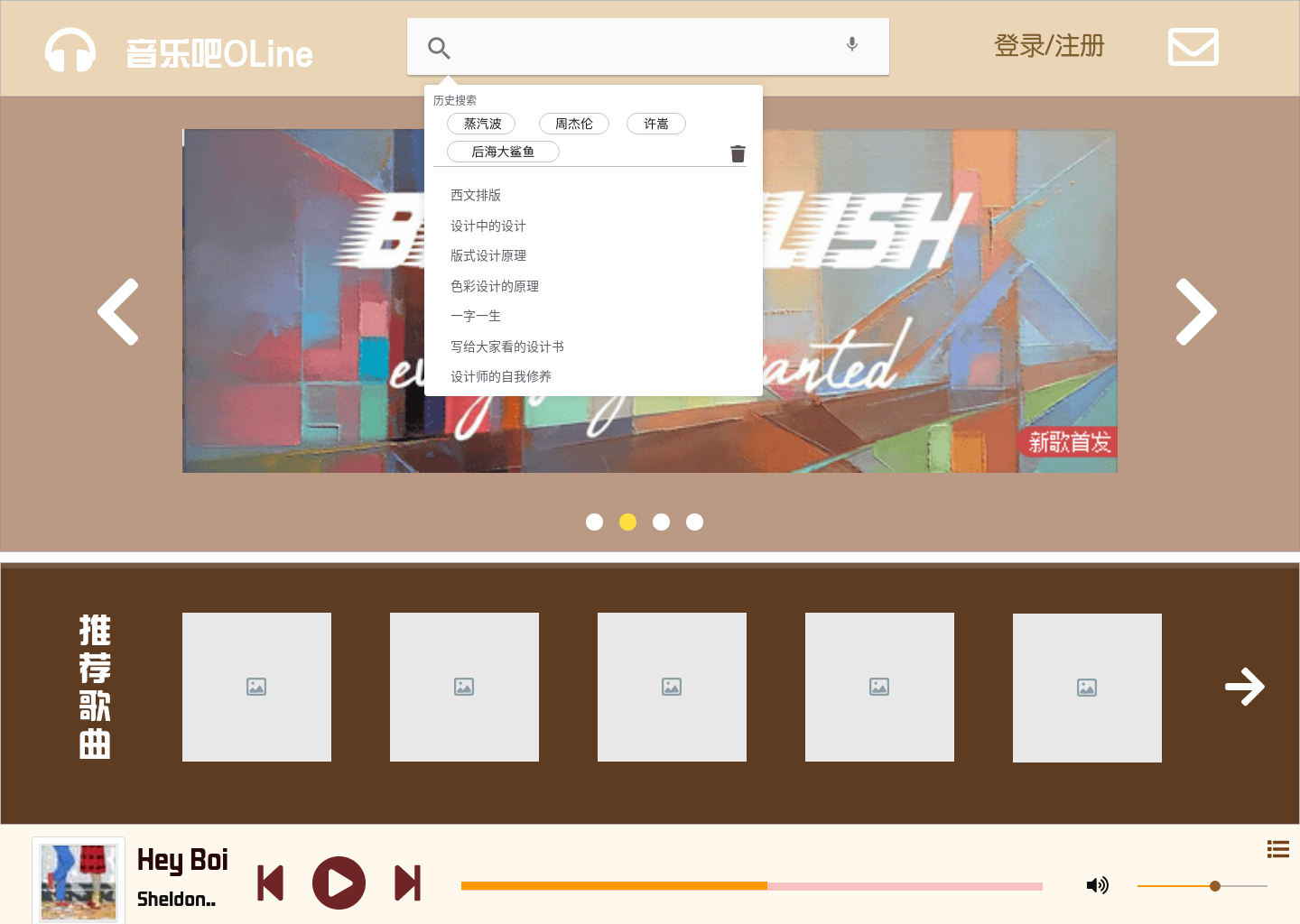
搜索栏会弹出下拉列表，包含搜索历史及相关热点。

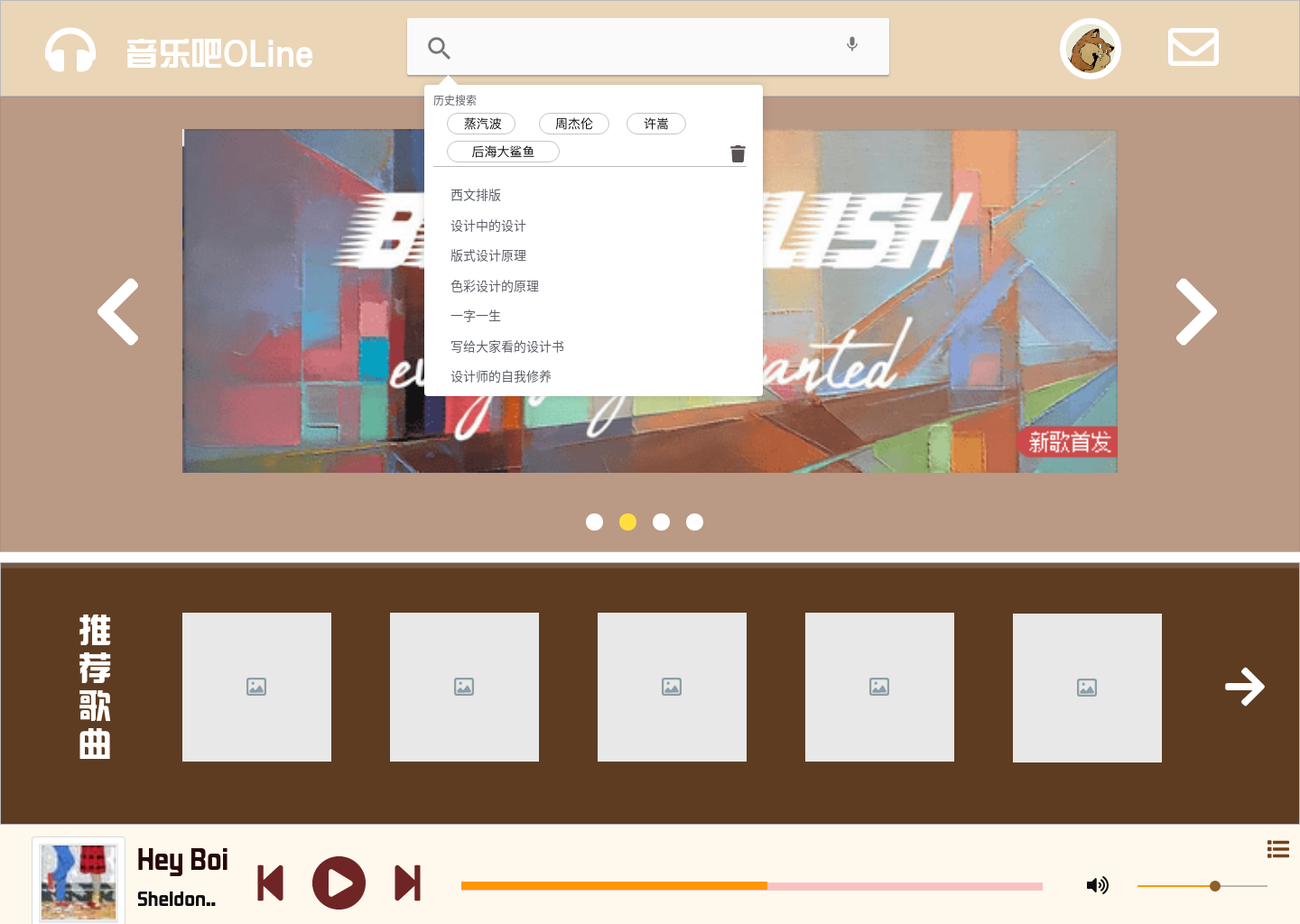
如果登录则显示用户头像，如图6-2.

点击私信图标跳转到私信界面，点击登录/注册跳转到登录/注册界面。

首页主要包括推荐歌曲及新歌速递。

下方为歌曲播放条，点击歌曲封面跳转到歌曲播放界面，点击右上角列表图标跳转到歌曲列表界面。

图6-1 首页设计图

6-2

## 登录/注册界面

注册界面要求填写用户名，绑定邮箱，手机号，密码信息。



图6-3

## 歌曲播放界面

播放列表主要为展示当前播放的歌曲列表，表示播放为哪一首歌曲，左侧有推荐歌曲图标，左上角文字点击可跳转至首页。

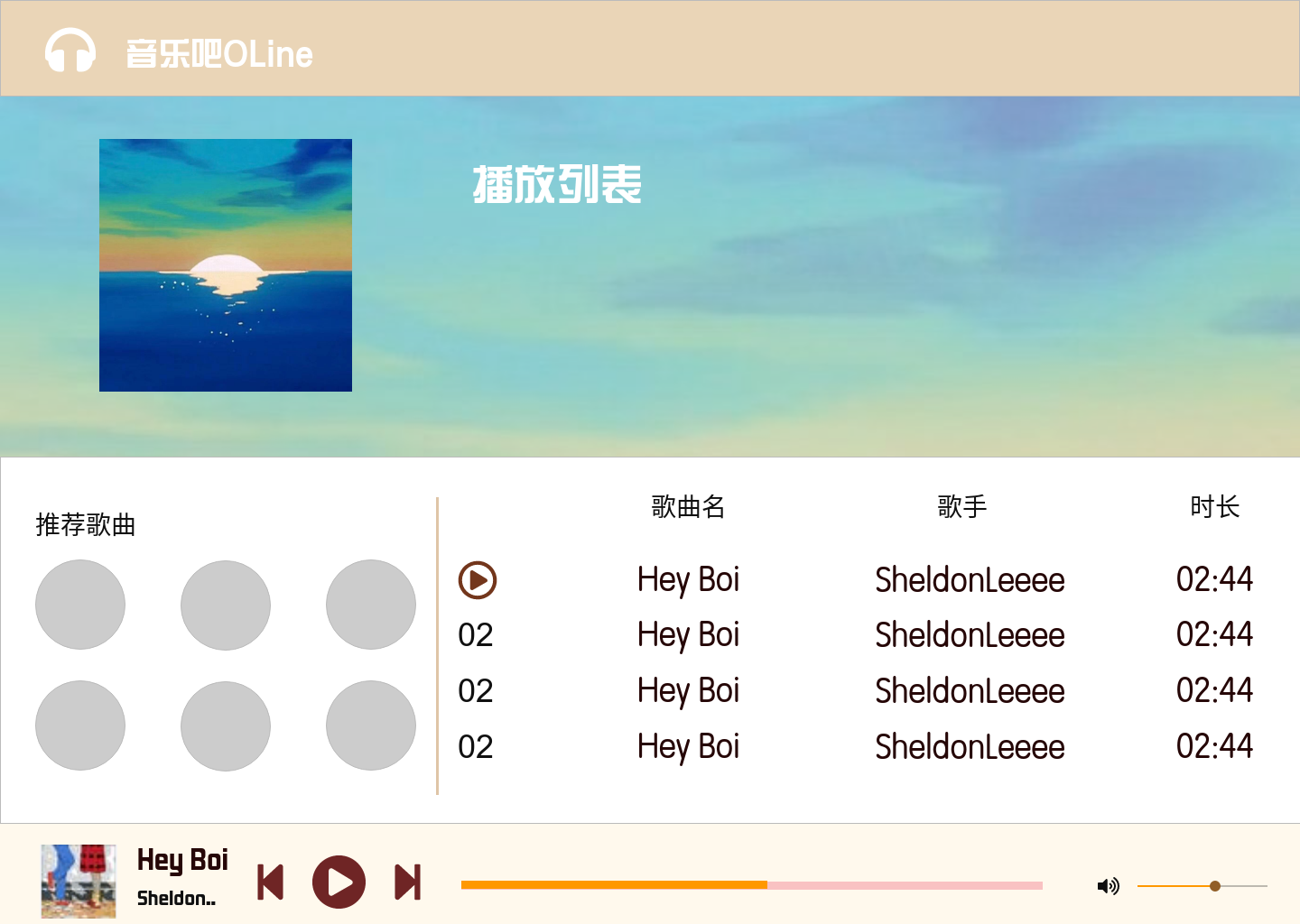


图6-4 Rose在线情缘个人中心页面设计图

## 论坛界面

论坛界面可以展示用户转发，分享，自行编辑的动态。可对动态进行点赞，评价，评论。左上角文字点击可跳转至首页，右上角用户头像可跳转至个人中心。

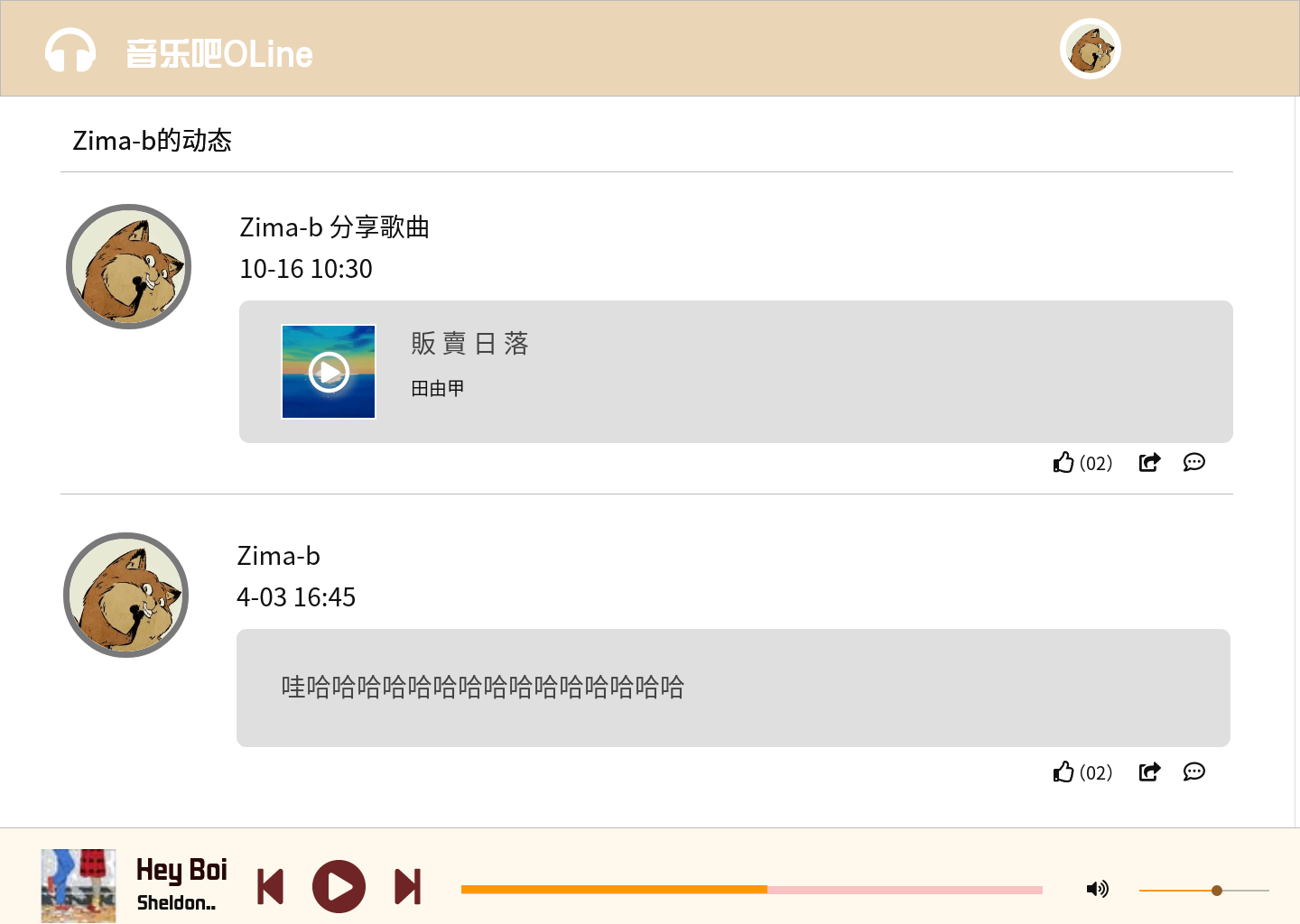
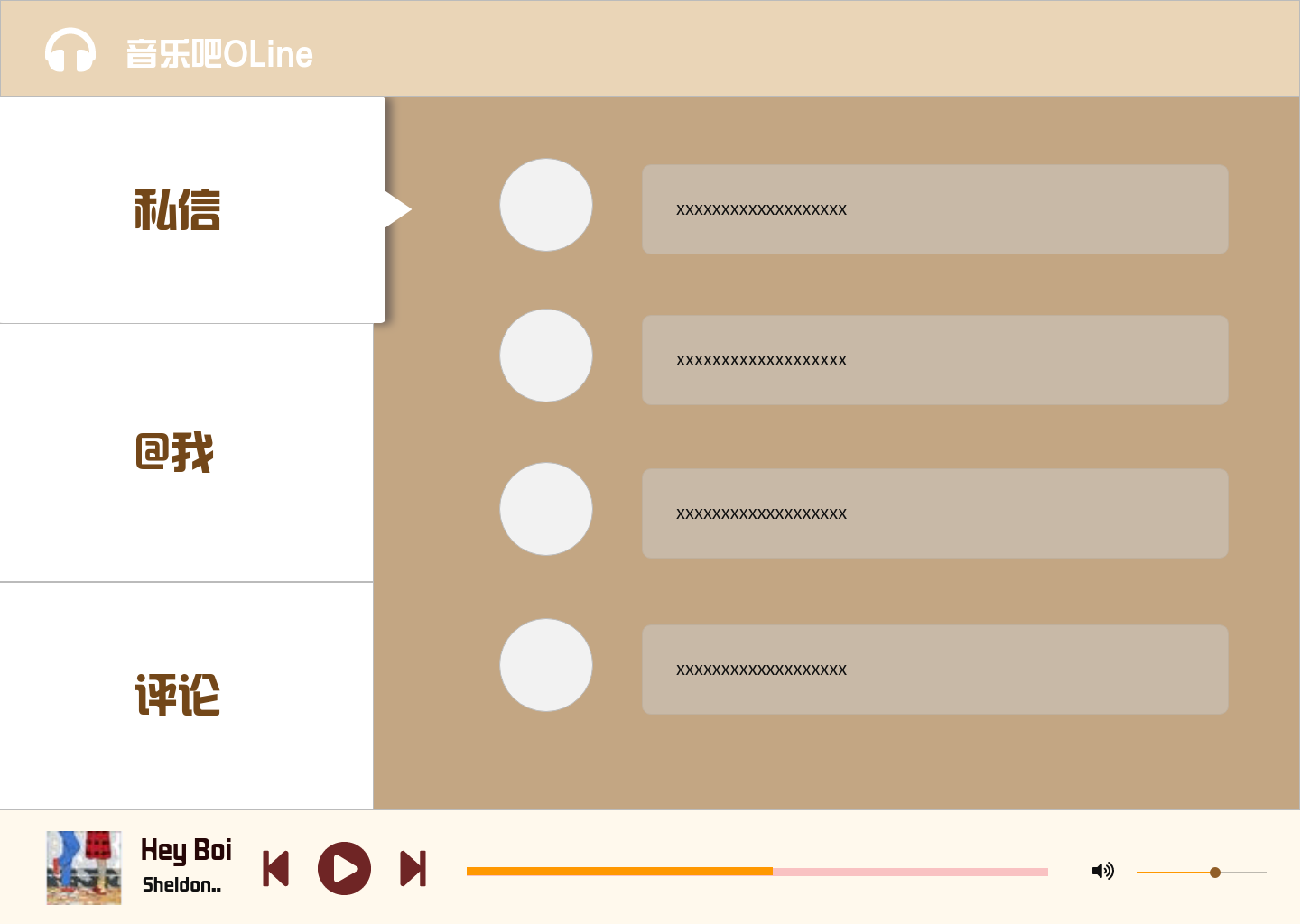


图6-14 Rose在线情缘活动页面设计图

## 私信界面

分为私信，评论，@我三个部分。



## 歌曲上传



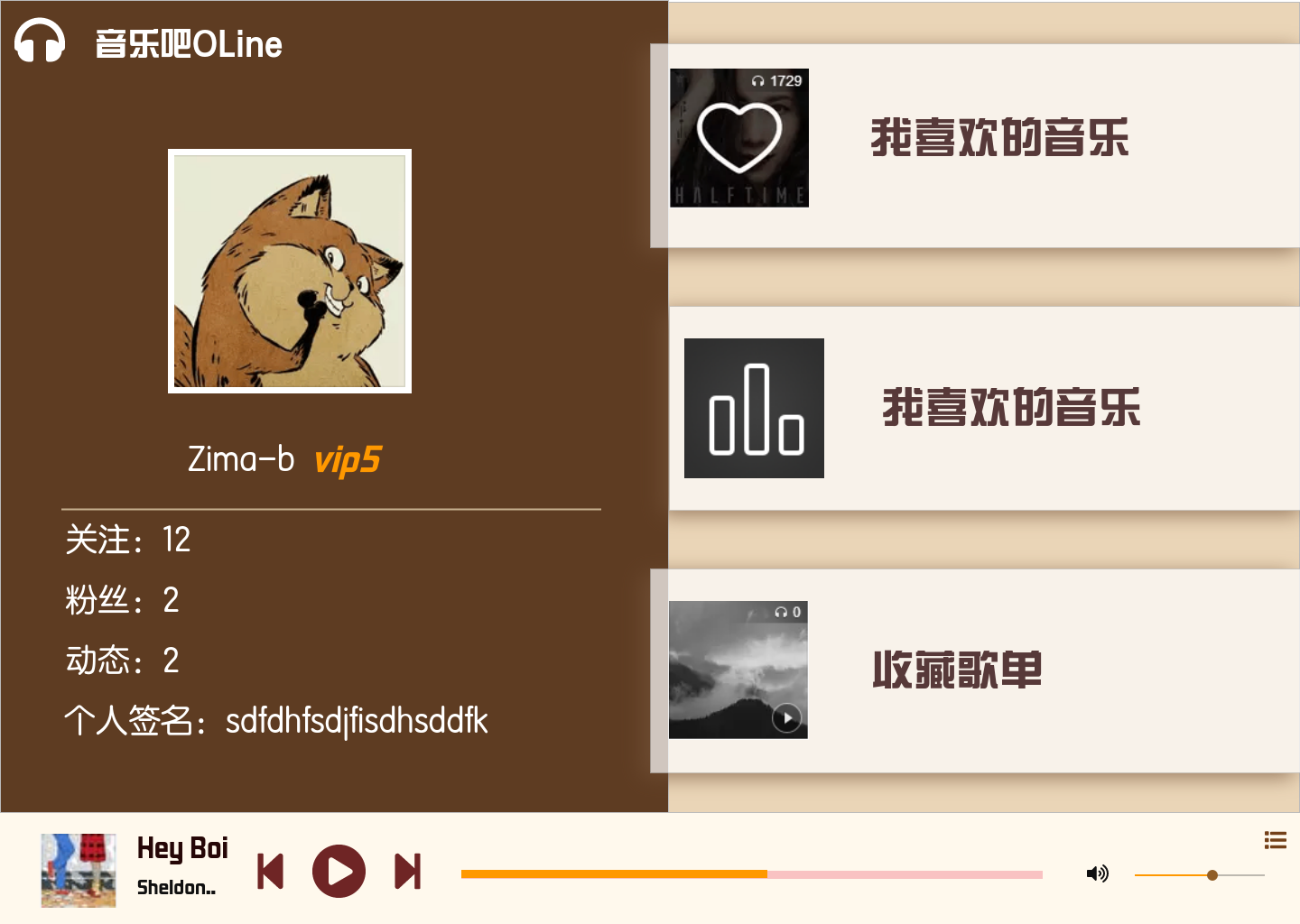
图6-21 Rose在线情缘用户搜索页面设计图

对于所有用户提供精确搜索服务，用户可以根据自身理想要求在线搜索满足条件的用户，用户可以输入昵称进行精确搜索，也可以输入用户的基本信息如性别，年龄区间，工作地区，身高区间，学历，收入区间，职业和星座进行大致搜索。点击立即搜索按钮后，相同昵称的用户将会以头像照片的形式显示在下方区域内，同时一个区域最多显示8名用户，最初随机推荐8名用户进行显示。点击用户头像便可出现给我写信按钮，方便给被搜索的用户留言或交流，搜索服务为用户提供更精确的用户查找，提更多和心仪伴侣沟通交流的机会。

管理员登录对管理员提供对登录网站的服务，登录界面如图6-22所示。

图6-22 后台管理员登录页面设计图

## 个人主页



后台主界面显示后台基本信息，具体设计如图6-23所示。

图6-23 后台主页面设计图

## 会员界面设计



后台会员管理界面显示所有会员基本信息，为管理员提供对用户的封禁、解封操作。管理员通过上方搜索栏输入用户完整邮箱精确搜索用户。具体设计如图6-24所示。

图6-24 后台会员管理页面设计图

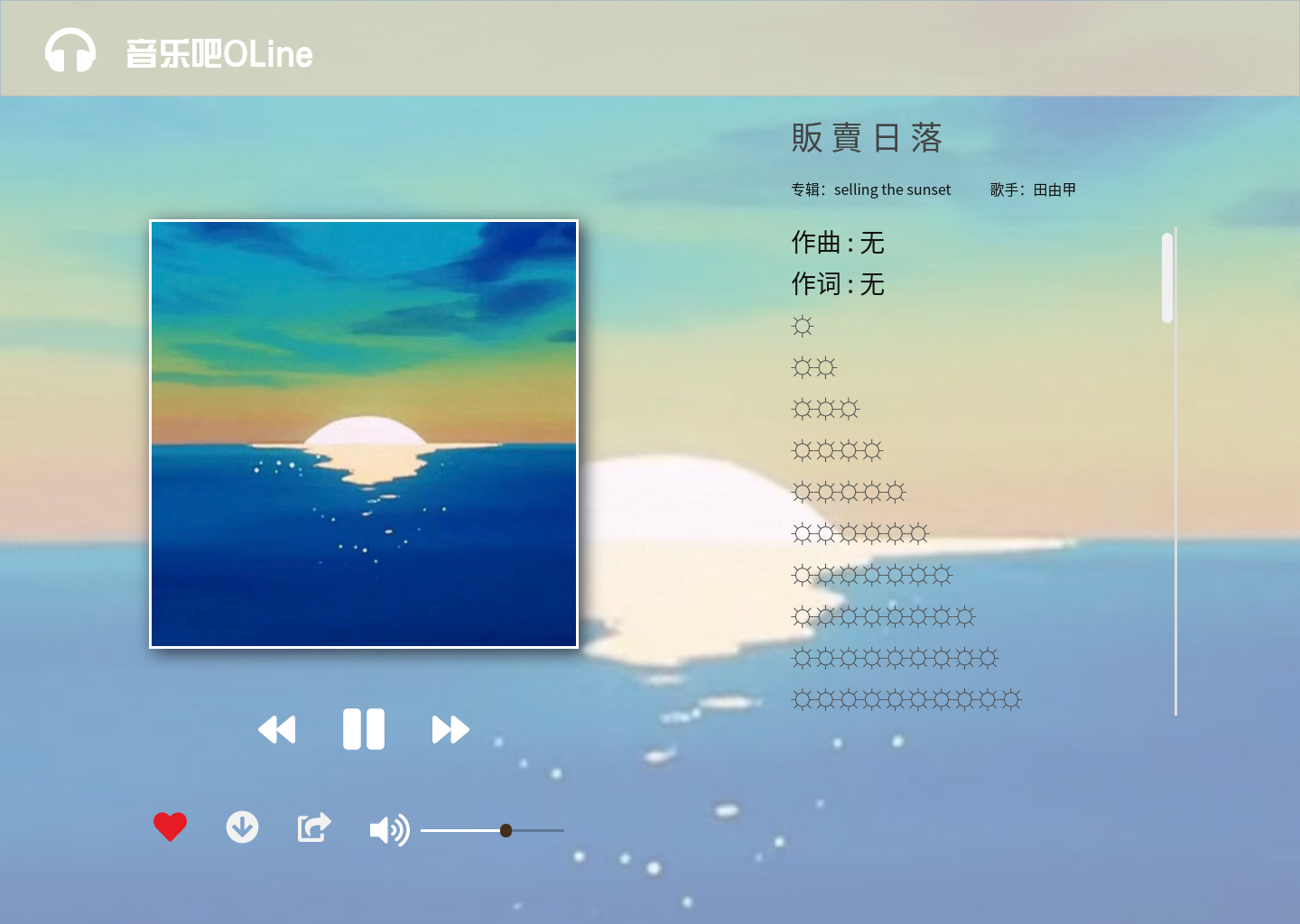
## 搜索界面



后台活动管理界面显示所有活动的基本信息，为管理员提供对活动的删除、恢复操作。管理员通过上方搜索栏输入活动的完整id精确搜索活动。具体设计如图6-25所示。

图6-25 后台活动管理页面设计图

## 音乐播放



### 

后台文章管理界面显示所有文章的基本信息，为管理员提供对文章的删除、恢复操作。管理员通过上方搜索栏输入文章的完整id精确搜索文章。具体设计如图6-26所示。

图6-26 后台文章管理页面设计图

后台评论管理界面显示所有评论的基本信息，为管理员提供对评论的删除、恢复操作。管理员通过上方搜索栏输入评论的完整id精确搜索评论。具体设计如图6-27所示。

图6-27 后台评论管理页面设计图