



教师：王晓东

评分：

# Java 应用与开发课程

## 爱音乐项目设计说明书

姓名	刘阳	任浩辰
学号	18020031052	18180021016
专业	计算机科学与技术	教育技术学
日期	2019-01-03	

中国海洋大学  
信息科学与工程学院计算机系

# 《爱音乐》

---

## 目录

目录.....	I
1. 系统概述.....	- 1 -
2. 总体设计.....	- 4 -
3. 模块设计.....	- 5 -
3.1. 模块命名规则.....	- 5 -
3.2. 模块汇总 .....	- 5 -
3.3. 子系统 A 的模块设计 .....	- 8 -
子系统 B 的模块设计.....	错误!未定义书签。
4. 数据库设计 .....	- 17 -
4.1. 数据库环境说明.....	- 42 -
4.2. 逻辑设计 .....	- 43 -
4.3. 物理设计 .....	- 45 -
5. 用户界面设计 .....	- 50 -
5.1. 界面的关系图和工作流程图.....	- 50 -
6. 附录：用户手册 .....	- 51 -

## 1. 系统概述

本系统是运用 Java Web 开发技术 ( Servlet、Session 等 ) 的在线音乐播放器 , 包含以下功能 :

### 1. 用户

- 用户的注册、登录和登出

### 2. 歌单、歌曲

- 创建、删除歌单
- 上传歌曲
- 上传歌单封面
- 浏览所有歌单和歌曲
- 已登录的用户可以对歌曲进行 Like 标记 , 或取消已 Like 标记的歌曲
- 按 Like 标记播放歌曲
- 显示歌曲的播放热度
- 对歌曲进行播放次数的排名 , 可可视化显示 Top20

### 3. 播放歌曲

- 在线播放音乐

### 4. 评论

- 对歌单进行评论
- 查看歌单评论

## 2. 需求分析（需求规格说明）

### 1 ) 注册

创建用户，要求用户名、和密码，并验证用户名和密码的合法性，不能重复以及不能为空值，符合规范。

### 2 ) 登录

使用已注册的用户名和密码进行登录，并使用 Cookie 和 Session 保持登录状态，用于要求用户登录的相关操作。

### 3 ) 登出

点击登出，即退出登录，不能进行要求用户登录的相关操作。

### 4 ) 浏览歌单、歌曲

页面显示歌单，并且内含所属的歌曲。

### 5 ) 播放歌曲

对于在页面上显示的歌曲，可点击进行播放。

### 6 ) 播放器

在页面底显示音乐播放器，对音乐的播放进行控制。

### 7 ) 标记 Like

在已登录的状态可以对歌曲进行 Like 标记 或者对已有 Like 标记的进行取消 Like 标记。

### 8 ) 我喜欢的歌曲

对于用户已标记为 Like 的歌曲，会特殊显示，并可以按照 Like 的歌曲进行浏览和播放。

### 9 ) 上传歌单、歌曲和歌单封面

## 《爱音乐》

---

已登录的用户可以上传自己创建的歌单，要求输入歌单名字、上传歌单内含的多首歌曲以及歌单的封面。

10 ) 排行榜

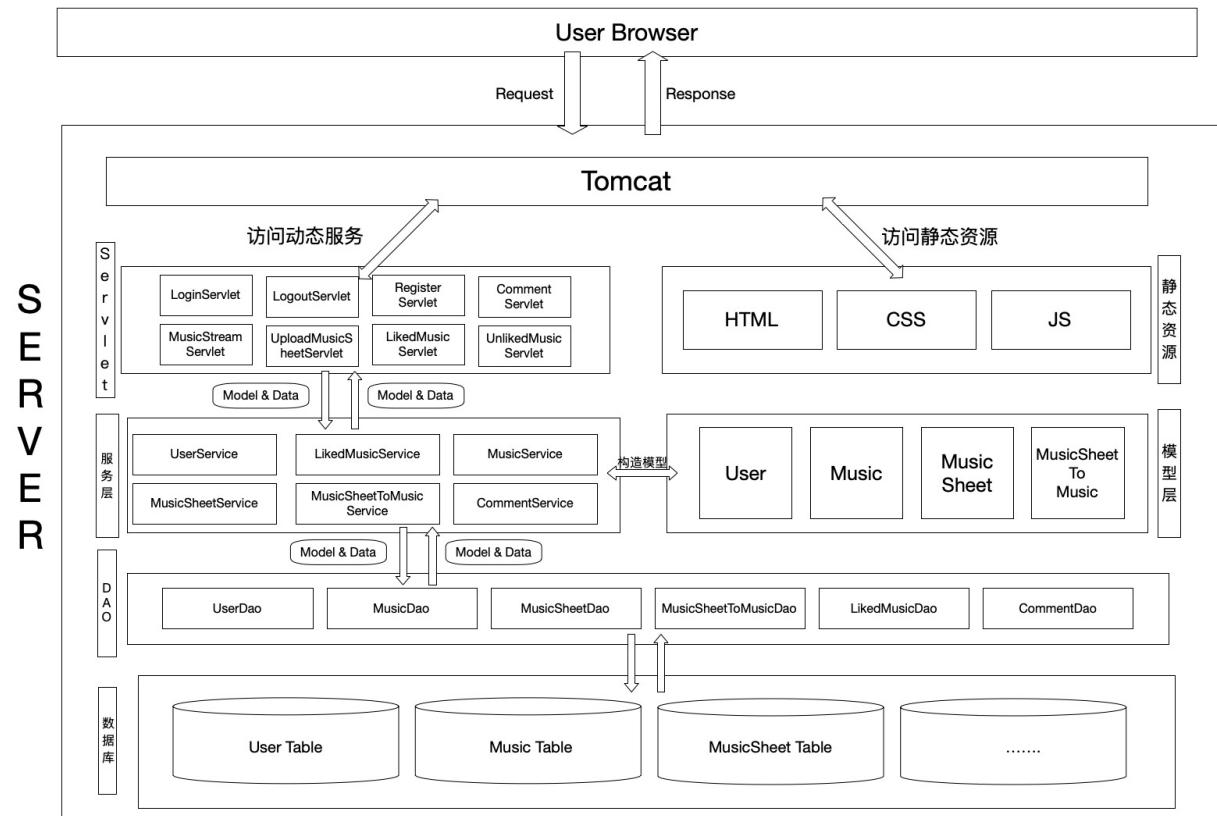
按照播放量对歌曲进行排行，可视化显示 Top20。

11 ) 评论

已登录的用户可以对歌单进行评论，所有的访客都可以查看特定歌单的相应评论。

### 3. 总体架构设计

本项目主要分为 Servlet、Service、Model 以及 Dao 后端架构，前端 HTML、CSS、JS 相关的页面显示美化，配合 MySQL 数据库对相关信息进行存储，以流的方式在前端进行歌曲的播放。



# 《爱音乐》

---

## 4. 模块设计

### 4.1. 模块命名规则

### 4.2. 模块汇总

子系统 dao	
模块名称	功能简述
CommentDao	评论数据库处理接口
LikedMusicDao	处理标记 LikedMusic 数据内容接口
MusicDao	处理 Music 数据内容接口
MusicSheetDao	处理 MusicSheet 数据内容接口
MusicSheetToMusicDao	处理 Music 和 MusicSheet 关系内容接口
UserDao	处理 User 数据内容接口
子系统 dao-impl	
模块名称	功能简述
CommentDaoImpl	数据库处理 CommentDao 接口的实现 , 处理评论的增加与查看
LikedMusicDaoImpl	数据库处理 LikedMusicDao 接口的实现 , 处理标记与去 Like 标记
MusicDaoImpl	数据库处理 MusicDao 接口的实现 , 处理 Music 模型的增删查改
MusicSheetDaoImpl	数据库处理 MusicSheetDao 接口的实现 , 处理 MusicSheet 模型的增删查改
MusicSheetToMusicDaoImpl	数据库处理 MusicSheetToMusicDao 接口的

## 《爱音乐》

	实现 ,处理 MusicSheet 和 Music 的对应关系
UserDaoImpl	数据库处理 UserDao 接口的实现 , 处理 User 模型的增加与查看
子系统 model	
模块名称	功能简述
Music	歌曲模型 , 存储数据对象类
MusicSheet	歌单模型 , 存储数据对象类
MusicSheetToMusic	歌单与歌曲对应 , 存储数据类
User	用户模型 , 存储数据对象类
子系统 service	
模块名称	功能简述
CommentService	评论处理服务
LikedMusicService	歌曲标记 Like 服务
MusicService	歌曲服务
MusicSheetService	歌单服务
MusicSheetToMusicService	歌单歌曲对应服务
UserService	用户服务
子系统 servlet	
模块名称	功能简述
CommentServlet	url /comment POST 发表评论接口 url /comment GET 获取当前歌单的评论接

## 《爱音乐》

	口
LikedMusicServlet	url /liked_music GET 过去当前用于所有喜爱歌曲接口
LikeMusicServlet	url /like POST 标记选中歌曲为喜爱接口
LoginServlet	url /login POST 登录接口
LogoutServlet	url /logout GET POST 登出接口
MusicDownloadServlet	url /downloadMusic POST 下载歌曲接口
MusicSheetAndFileUploadServlet	url /upload POST 上传歌单文件接口
MusicSheetPictureDownloadServlet	url /downloadPicture POST 下载歌单图片接口
MusicSheetQueryServlet	url /queryMusicSheets GET 获取 Top20 接口
MusicUploadServlet	url /sheetUpload POST 上传歌单信息接口
MusicStreamServlet	url /music POST 返回歌曲流接口
RegisterServlet	url /register POST 注册接口
UnlikeServlet	url /unlike POST 取消标记为喜爱的歌曲
子系统 util	
模块名称	功能简述
DatabaseUtil	数据库工具类
JsonReader	Json 解析工具类
MD5Util	MD5 工具类

## 4.3. 子系统 A 的模块设计

### 4.3.1. 模块 dao-CommentDao

模块名称	CommentDao
功能描述	评论数据库处理接口
接口与属性	<code>void insert(User user, MusicSheet musicSheet, String content);</code> 插入评论 <code>HashMap&lt;Integer, String&gt; getComment(MusicSheet musicSheet)</code> 获取特定歌单用户 id 和评论的对应
数据结构与算法	仅为接口
补充说明	仅为接口

### 4.3.2. 模块 dao- LikedMusicDao

模块名称	LikedMusicDao
功能描述	处理标记 LikedMusic 数据内容接口
接口与属性	<code>void insert(User user, Music music);</code> 插入 LikeMusic <code>void delete(User user, Music music);</code> 删除 LikeMusic <code>List&lt;Music&gt; findLikedMusic(User user);</code>

## 《爱音乐》

	获取 User 的 LikeMusic  boolean isLiked(User user, Music music);  判断 User 是否 LikeMusic
数据结构与算法	仅为接口
补充说明	仅为接口

### 4.3.3. 模块 dao-MusicDao

模块名称	MusicDao
功能描述	处理 Music 数据内容接口
接口与属性	public int insert(Music mu) throws SQLException;  插入 Music  public void update(Music mu) throws SQLException;  更新 Music  public void delete(int id) throws SQLException;  删除 Music  public void deleteByMd5value(String md5value) throws SQLException;  通过 Md5 删除 Music  public Music findById(int id) throws SQLException;  通过 id 查找 Music

	<pre>public List&lt;Music&gt; findAll() throws SQLException;</pre> <p>获取所有的 Music</p> <pre>public Music findByMd5value(String md5value) throws SQLException;</pre> <p>通过 md5 查找 Music</p>
数据结构 与算法	仅为接口
补充说明	仅为接口

#### 4.3.4. 模块 dao- MusicSheetDao

模块名称	MusicSheetDao
功能描述	处理 MusicSheet 数据内容接口
接口 与属性	<pre>public int insert(MusicSheet ms) throws SQLException;</pre> <p>插入 MusicSheet</p> <pre>public void update(MusicSheet ms) throws SQLException;</pre> <p>更新 MusicSheet</p> <pre>public void delete(int id) throws SQLException;</pre> <p>删除 MusicSheet</p> <pre>public MusicSheet findById(int id) throws SQLException;</pre> <p>通过 id 查找 MusicSheet</p> <pre>public MusicSheet findByUuid(String uuid) throws SQLException;</pre>

	<pre>SQLException;  通过 uuid 查找 MusicSheet  public List&lt;MusicSheet&gt; findAll() throws SQLException;  获取所有的 MusicSheet  void deleteByUuid(String uuid) throws SQLException;  通过 uuid 删除 MusicSheet  List&lt;MusicSheet&gt; findLatest(int num) throws SQLException;  获取 TopMusicSheet</pre>
数据结构	仅为接口
与算法	
补充说明	仅为接口

#### 4.3.5. 模块 dao- MusicSheetToMusicDao

模块名称	MusicSheetToMusicDao
功能描述	处理 Music 和 MusicSheet 关系内容接口
接口 与属性	<pre>public int insert(MusicSheetToMusic mstm) throws SQLException;  插入 MusicSheet 和 Music 对应关系  public void delete(int id) throws SQLException;  删除 MusicSheet 和 Music 对应关系  public List&lt;Integer&gt; findByMusicSheetId(int id) throws SQLException;</pre>

	<pre>SQLException;</pre> <p>通过 id 查找 MusicSheet 和 Music 对应关系</p>
数据结构 与算法	仅为接口
补充说明	仅为接口

#### 4.3.6. 模块 dao- UserDao

模块名称	UserDao
功能描述	处理 User 数据内容接口
接口 与属性	<pre>int insert(User user);</pre> <p>插入 User</p> <pre>User findById(int id);</pre> <p>通过 id 查找 User</p> <pre>User findByName(String name);</pre> <p>通过 name 查找 User</p> <pre>User findOne(String name, String passMd5Value);</pre> <p>通过 name 和 passMd5Value 查找 User</p> <pre>List&lt;User&gt; findAll();</pre> <p>获取所有的 User</p>
数据结构 与算法	仅为接口

补充说明	仅为接口
------	------

## 4.3.7. 模块 dao-impl- CommentDaoImpl

模块名称	CommentDaoImpl
功能描述	数据库处理 CommentDao 接口的实现，处理评论的增加与查看
接口 与属性	<code>void insert(User user, MusicSheet musicSheet, String content);</code>  插入评论  <code>HashMap&lt;Integer, String&gt; getComment(MusicSheet musicSheet)</code>  获取特定歌单用户 id 和评论的对应  <code>public static void makePreparations();</code>  准备阶段数据库创建 comment 表
数据结构 与算法	
补充说明	

## 4.3.8. 模块 dao-impl- LikedMusicDaoImpl

模块名称	LikedMusicDaoImpl
功能描述	数据库处理 LikedMusicDao 接口的实现，处理标记与去 Like 标记
接口 与属性	<code>void insert(User user, Music music);</code>  插入 LikeMusic

## 《爱音乐》

	<pre>void delete(User user, Music music);  删除 LikeMusic  List&lt;Music&gt; findLikedMusic(User user);  获取 User 的 LikeMusic  boolean isLiked(User user, Music music);  判断 User 是否 LikeMusic  public static void makePreparations();  准备阶段数据库创建 like_music 表</pre>
数据结构 与算法	
补充说明	

### 4.3.9. 模块 dao-impl- MusicDaoImpl

模块名称	MusicDaoImpl
功能描述	数据库处理 MusicDao 接口的实现，处理 Music 模型的增删查改
接口 与属性	<pre>public int insert(Music mu) throws SQLException;  插入 Music  public void update(Music mu) throws SQLException;  更新 Music  public void delete(int id) throws SQLException;  删除 Music</pre>

## 《爱音乐》

	<pre>public void deleteByMd5value(String md5value) throws SQLException;</pre> <p>通过 Md5 删除 Music</p> <pre>public Music findById(int id) throws SQLException;</pre> <p>通过 id 查找 Music</p> <pre>public List&lt;Music&gt; findAll() throws SQLException;</pre> <p>获取所有的 Music</p> <pre>public Music findByMd5value(String md5value) throws SQLException;</pre> <p>通过 md5 查找 Music</p> <pre>public static void makePreparations();</pre> <p>准备阶段数据库创建 music 表</p>
数据结构 与算法	
补充说明	

### 4.3.10. 模块 dao-impl- MusicSheetDaoImpl

模块名称	MusicSheetDaoImpl
功能描述	数据库处理 MusicSheetDao 接口的实现，处理 MusicSheet 模型的增删查改
接口	<pre>public int insert(MusicSheet ms) throws SQLException;</pre>

与属性	<p>插入 MusicSheet</p> <pre>public void update(MusicSheet ms) throws SQLException;</pre> <p>更新 MusicSheet</p> <pre>public void delete(int id) throws SQLException;</pre> <p>删除 MusicSheet</p> <pre>public MusicSheet findById(int id) throws SQLException;</pre> <p>通过 id 查找 MusicSheet</p> <pre>public MusicSheet findByUuid(String uuid) throws SQLException;</pre> <p>通过 uuid 查找 MusicSheet</p> <pre>public List&lt;MusicSheet&gt; findAll() throws SQLException;</pre> <p>获取所有的 MusicSheet</p> <pre>void deleteByUuid(String uuid) throws SQLException;</pre> <p>通过 uuid 删除 MusicSheet</p> <pre>List&lt;MusicSheet&gt; findLatest(int num) throws SQLException;</pre> <p>获取 TopMusicSheet</p> <pre>public static void makePreparations();</pre> <p>准备阶段数据库创建 musicsheet 表</p>
数据结构 与算法	
补充说明	

#### 4.3.11. 模块 dao-impl- MusicSheetToMusicDaoImpl

模块名称	MusicSheetToMusicDaoImpl
功能描述	数据库处理 MusicSheetToMusicDao 接口的实现 , 处理 MusicSheet 和 Music 的对应关系
接口与属性	<pre>public int insert(MusicSheetToMusic mstm) throws SQLException;           插入 MusicSheet 和 Music 对应关系            public void delete(int id) throws SQLException;           删除 MusicSheet 和 Music 对应关系            public List&lt;Integer&gt; findByMusicSheetId(int id) throws SQLException;           通过 id 查找 MusicSheet 和 Music 对应关系            public static void makePreparations();           准备阶段数据库创建 musicsheet_music 表</pre>
数据结构与算法	
补充说明	

#### 4.3.12. 模块 dao-impl- UserDaoImpl

模块名称	UserDaoImpl
功能描述	数据库处理 UserDao 接口的实现 , 处理 User 模型的增加与查看

## 《爱音乐》

接口 与属性	<pre>int insert(User user);</pre> <p>插入 User</p> <pre>User findById(int id);</pre> <p>通过 id 查找 User</p> <pre>User findByName(String name);</pre> <p>通过 name 查找 User</p> <pre>User findOne(String name, String passMd5Value);</pre> <p>通过 name 和 passMd5Value 查找 User</p> <pre>List&lt;User&gt; findAll();</pre> <p>获取所有的 User</p> <pre>public static void makePreparations();</pre> <p>准备阶段数据库创建 user 表</p>
数据结构 与算法	
补充说明	

### 4.3.13. 模块 model- Music

模块名称	Music
功能描述	歌曲模型，存储数据对象类
接口	<pre>private int id;</pre>
与属性	Music 的 id

```
private String name;  
Music 的名字  
  
private String singer;  
Music 的歌手  
  
private String md5value;  
Music 的 MD5 值  
  
public Music() {}  
Music 的无参构造函数  
  
public Music(String md5value, String name)  
通过 MD5 和名字构造 Music  
  
public Music(String md5value, String name, String singer)  
通过 MD5、名字和歌手构造 Music  
  
public int getId()  
获取 Music 的 id  
  
public void setId(int id)  
设置 Music 的 id  
  
public String getName()  
获取 Music 的名字  
  
public void setName(String name)  
设置 Music 的名字  
  
public String getSinger()  
获取 Music 的歌手
```

	<pre>public void setSinger(String singer)</pre> <p>设置 Music 的歌手</p> <pre>public String getMd5value()</pre> <p>获取 Music 的 MD5 值</p> <pre>public void setMd5value(String md5value)</pre> <p>设置 Music 的 MD5 值</p>
数据结构 与算法	
补充说明	

#### 4.3.14. 模块 model- MusicSheet

模块名称	MusicSheet
功能描述	歌曲模型，存储数据对象类
接口 与属性	<pre>private int id;</pre> <p>MusicSheet 的 id</p> <pre>private String uuid;</pre> <p>MusicSheet 的 uuid</p> <pre>private String name;</pre> <p>MusicSheet 的名字</p> <pre>private String creatorId;</pre> <p>MusicSheet 的创建者 id</p>

```
private String creator;  
MusicSheet 的创建者  
  
private String dateCreated;  
MusicSheet 的创建时间  
  
private String picture;  
MusicSheet 的歌单封面  
  
private Map<String, String> musicItems;  
MusicSheet 对应的歌曲 MD5 值  
  
public MusicSheet()  
MusicSheet 的无参构造函数  
  
public MusicSheet(String name)  
通过 MusicSheet 的名字构造 MusicSheet  
  
public void setUuid(String uuid)  
设置 MusicSheet 的 uuid  
  
public String getUuid()  
获取 MusicSheet 的 uuid  
  
public String getCreatorId()  
获取 MusicSheet 的创建者 id  
  
public void setCreatorId(String creatorId)  
设置 MusicSheet 创建者 id  
  
public void setDateCreated(String dateCreated)  
设置 MusicSheet 的创建时间
```

	<pre>public String getDateCreated()</pre> <p>获取 MusicSheet 的创建时间</p> <pre>public String getCreator()</pre> <p>获取 MusicSheet 的创建者</p> <pre>public void setCreator(String creator)</pre> <p>设置 MusicSheet 的创建者</p> <pre>public String getName()</pre> <p>获取 MusicSheet 的名字</p> <pre>public void setName(String name)</pre> <p>设置 MusicSheet 的名字</p> <pre>public String getPicture()</pre> <p>获取 MusicSheet 的图片</p> <pre>public void setPicture(String picture)</pre> <p>设置 MusicSheet 的图片</p> <pre>public Map&lt;String, String&gt; getMusicItems()</pre> <p>获取 MusicSheet 的 Music 的对应关系</p> <pre>public void setMusicItems(Map&lt;String, String&gt; musicItems)</pre> <p>设置 MusicSheet 的 Music 的对应关系</p> <pre>public int getId()</pre> <p>获取 MusicSheet 的 id</p> <pre>public void setId(int id)</pre> <p>设置 MusicSheet 的 id</p>
--	---

数据结构 与算法	
补充说明	

## 4.3.15. 模块 model- MusicSheetToMusic

模块名称	MusicSheetToMusic
功能描述	歌单与歌曲对应，存储数据类
接口 与属性	<pre>private int id; MusicSheetToMusic 的自增 id  private int musicSheetId; MusicSheet 的 id  private int musicId; Music 的 id  public MusicSheetToMusic() MusicSheetToMusic 的无参构造函数  public MusicSheetToMusic(int musicSheetId, int musicId) 通过 Music 的 id 和 Musicsheet 的 id 构造 MusicSheetToMusic  public int getMusicSheetId() 获取 MusicSheetToMusic 歌单的 id  public void setMusicSheetId(int musicSheetId) 设置 MusicSheetToMusic 歌单的 id  public int getMusicId()</pre>

## 《爱音乐》

	<p>获取 MusicSheetToMusic 歌曲 id</p> <pre>public void setMusicId(int musicId)</pre> <p>设置 MusicSheetToMusic 歌曲 id</p> <pre>public int getId()</pre> <p>获取 MusicSheetToMusic 的 id</p> <pre>public void setId(int id)</pre> <p>设置 MusicSheetToMusic 的 id</p>
数据结构 与算法	
补充说明	

### 4.3.16. 模块 model- User

模块名称	User
功能描述	用户模型，存储数据对象类
接口 与属性	<pre>private int id;</pre> <p>User 的 id</p> <pre>private String name;</pre> <p>User 的用户名</p> <pre>private String passMd5value;</pre> <p>User 的密码 MD5 值</p> <pre>public User()</pre>

	<p>User 的无参构造函数</p> <pre>public User(String name, String passMd5value)</pre> <p>通过 User 的用户名和密码 MD5 值构造 User</p> <pre>public User(int id, String name)</pre> <p>通过 User 的 id 和用户名构造 User</p> <pre>public User(int id, String name, String passMd5value)</pre> <p>通过 User 的 id 用户名密码的 MD5 值构造 User</p> <pre>public int getId()</pre> <p>获取 User 的 id</p> <pre>public void setId(int id)</pre> <p>设置 User 的 id</p> <pre>public String getName()</pre> <p>获取 User 的用户名</p> <pre>public void setName(String name)</pre> <p>设置 User 的用户名</p> <pre>public String getPassMd5value()</pre> <p>获取 User 的密码 MD5 值</p> <pre>public void setPassMd5value(String passMd5value)</pre> <p>设置 User 的密码 MD5 值</p>
数据结构 与算法	
补充说明	

## 4.3.17. 模块 service-CommentService

模块名称	CommentService
功能描述	评论处理服务
接口与属性	<p>CommentDaoImpl commentDao</p> <p>CommentDao 处理数据库数据对象</p> <pre>public void comment(User user, MusicSheet musicSheet, String content)</pre> <p>添加一条评论</p> <pre>public HashMap&lt;Integer, String&gt; getComments(MusicSheet musicSheet)</pre> <p>获取指定 MusicSheet 的用户对应的评论内容</p>
数据结构与算法	
补充说明	

## 4.3.18. 模块 service- LikedMusicService

模块名称	LikedMusicService
功能描述	歌曲标记 Like 服务
接口与属性	<p>LikedMusicDaoImpl likedMusicDao</p> <p>LikedMusicDao 处理数据库数据对象</p>

## 《爱音乐》

	<pre>public LikedMusicService()</pre> <p>服务的构造函数</p> <pre>public void like(User user, Music music)</pre> <p>标记一首歌曲为 Like</p> <pre>public void unlike(User user, Music music)</pre> <p>去除一首歌曲的 Like</p> <pre>public List&lt;Music&gt; getUserLikedMusic(User user)</pre> <p>获取指定 User 的 Like 的所有歌曲</p>
数据结构 与算法	
补充说明	

### 4.3.19. 模块 service- MusicService

模块名称	MusicService
功能描述	歌曲服务
接口 与属性	<p>MusicDao musicDao</p> <p>musicDao 处理数据库数据对象</p> <pre>public MusicService()</pre> <p>Music 服务构造函数</p> <pre>public int create(Music mu) throws SQLException</pre> <p>添加一首歌曲</p>

## 《爱音乐》

	<pre>public void getAll() throws SQLException  获取所有的歌曲  public String getFilePathByMd5value(String md5value) throws SQLException  通过歌曲的 MD5 值获取歌曲文件的路径  public Music getMusicByMd5Value(String md5value) throws SQLException  通过歌曲的 MD5 值获取 Music</pre>
数据结构 与算法	
补充说明	

### 4.3.20. 模块 service- MusicSheetService

模块名称	MusicSheetService
功能描述	歌单服务
接口 与属性	<pre>MusicSheetDao musicSheetDao MusicSheetDao 处理数据库数据对象  MusicDao musicDao  MusicDao 处理数据库数据对象  MusicSheetToMusicDao mstmDao  MusicSheetToMusicDao 处理数据库数据对象</pre>

## 《爱音乐》

	<pre>public MusicSheetService()  歌单服务的构造函数      public int create(MusicSheet ms) throws SQLException  创建一个歌单      public void createOrUpdate(MusicSheet ms) throws SQLException  更新一个歌单      public List&lt;MusicSheet&gt; getAll() throws SQLException  获取所有的歌单      public List&lt;MusicSheet&gt; getLatest(int num) throws SQLException  获取最新的歌单      public MusicSheet getMusicByUuid(String uuid)  通过歌单的 uuid 获取歌单      public String getMusicSheetPictureUrl(String uuid) throws SQLException  获取歌单封面的 url      public void delete(String uuid) throws SQLException  删除歌单</pre>
数据结构 与算法	
补充说明	

## 4.3.21. 模块 service- MusicSheetToMusicService

模块名称	MusicSheetToMusicService
功能描述	歌单歌曲对应服务
接口 与属性	<p>MusicSheetToMusicDao mstmDao</p> <p>MusicSheetToMusicDao 处理数据库数据对象</p> <p>public MusicSheetToMusicService()</p> <p>MusicSheetToMusic 服务构造函数</p> <p>public int create(MusicSheetToMusic mstm) throws SQLException</p> <p>创建歌单和歌曲的对应</p>
数据结构 与算法	
补充说明	

## 4.3.22. 模块 service- UserService

模块名称	UserService
功能描述	用户服务
接口 与属性	<p>UserDaoImpl userDao</p> <p>UserDao 处理数据库数据对象</p> <p>public int create(User user)</p>

	<p>创建一个用户</p> <pre>public List&lt;User&gt; getAll()</pre> <p>获取所有的用户</p> <pre>public User getOne(int id)</pre> <p>通过 id 获取用户</p> <pre>public User getOne(String name)</pre> <p>通过名字获取用户</p> <pre>public User getOne(String name, String passMd5Value)</pre> <p>通过名字和密码 MD5 值获取用户</p>
数据结构 与算法	
补充说明	

#### 4.3.23. 模块 servlet- CommentServlet

模块名称	CommentServlet
功能描述	<p>url /comment POST 发表评论接口</p> <p>url /comment GET 获取当前歌单的评论接口</p>
接口 与属性	<p>protected void doPost(HttpServletRequest request, HttpServletResponse response)</p> <p>发表评论</p> <p>protected void doGet(HttpServletRequest request,</p>

## 《爱音乐》

	HttpServletResponse response) 获取当前歌单的评论
数据结构 与算法	
补充说明	

### 4.3.24. 模块 servlet- LikedMusicServlet

模块名称	LikedMusicServlet
功能描述	url /liked_music GET 获取当前用于所有喜爱歌曲接口
接口 与属性	protected void doPost(HttpServletRequest request, HttpServletResponse response)  获取当前用于所有喜爱歌曲  protected void doGet(HttpServletRequest request, HttpServletResponse response)  获取当前用于所有喜爱歌曲
数据结构 与算法	
补充说明	

### 4.3.25. 模块 servlet- LikeMusicServlet

模块名称	LikeMusicServlet
------	------------------

## 《爱音乐》

功能描述	url /like POST 标记选中歌曲为喜爱接口
接口 与属性	protected void doPost(HttpServletRequest request, HttpServletResponse response)  标记选中歌曲为喜爱  protected void doGet(HttpServletRequest request, HttpServletResponse response)  标记选中歌曲为喜爱
数据结构 与算法	
补充说明	

### 4.3.26. 模块 servlet- LoginServlet

模块名称	LoginServlet
功能描述	url /login POST 登录接口
接口 与属性	protected void doPost(HttpServletRequest request, HttpServletResponse response)  登录  protected void doGet(HttpServletRequest request, HttpServletResponse response)  非法 400
数据结构	

与算法	
补充说明	

#### 4.3.27. 模块 servlet- LogoutServlet

模块名称	LogoutServlet
功能描述	url /logout GET POST 登出接口
接口 与属性	<pre>protected void doPost(HttpServletRequest request, HttpServletResponse response)     登出  protected void doGet(HttpServletRequest request, HttpServletResponse response)     登出</pre>
数据结构	
与算法	
补充说明	

#### 4.3.28. 模块 servlet- MusicDownloadServlet

模块名称	MusicDownloadServlet
功能描述	url /downloadMusic POST 下载歌曲接口
接口 与属性	<pre>protected void doPost(HttpServletRequest request, HttpServletResponse response)</pre>

	下载歌曲 <pre>protected void doGet(HttpServletRequest request, HttpServletResponse response)</pre> 下载歌曲
数据结构 与算法	
补充说明	

#### 4.3.29. 模块 servlet- MusicSheetAndFileUploadServlet

模块名称	MusicSheetAndFileUploadServlet
功能描述	url /upload POST 上传歌单文件接口
接口 与属性	<pre>protected void doPost(HttpServletRequest request, HttpServletResponse response)</pre> 上传歌单文件 <pre>protected void doGet(HttpServletRequest request, HttpServletResponse response)</pre> 上传歌单文件
数据结构 与算法	
补充说明	

### 4.3.30. 模块 servlet- MusicSheetPictureDownloadServlet

模块名称	MusicSheetPictureDownloadServlet
功能描述	url /downloadPicture POST 下载歌单图片接口
接口 与属性	<pre>protected void doPost(HttpServletRequest request, HttpServletResponse response) 下载歌单图片</pre> <pre>protected void doGet(HttpServletRequest request, HttpServletResponse response) 下载歌单图片</pre>
数据结构 与算法	
补充说明	

### 4.3.31. 模块 servlet- MusicSheetQueryServlet

模块名称	MusicSheetQueryServlet
功能描述	url /queryMusicSheets GET 获取 Top20 接口
接口 与属性	<pre>protected void doPost(HttpServletRequest request, HttpServletResponse response) 下载歌单图片</pre> <pre>protected void doGet(HttpServletRequest request, HttpServletResponse response)</pre>

## 《爱音乐》

---

	下载歌单图片
数据结构 与算法	
补充说明	

### 4.3.32. 模块 servlet- MusicUploadServlet

模块名称	MusicUploadServlet
功能描述	url /sheetUpload POST 上传歌单信息接口
接口 与属性	protected void doPost(HttpServletRequest request, HttpServletResponse response)  上传歌单信息  protected void doGet(HttpServletRequest request, HttpServletResponse response)  上传歌单信息
数据结构 与算法	
补充说明	

### 4.3.33. 模块 servlet- MusicStreamServlet

模块名称	MusicStreamServlet
功能描述	url /music POST 返回歌曲流接口

接口 与属性	<pre>protected void doPost(HttpServletRequest request, HttpServletResponse response)  返回歌曲流  protected void doGet(HttpServletRequest request, HttpServletResponse response)  返回歌曲流</pre>
数据结构 与算法	
补充说明	

#### 4.3.34. 模块 servlet- RegisterServlet

模块名称	RegisterServlet
功能描述	url /register POST 注册接口
接口 与属性	<pre>protected void doPost(HttpServletRequest request, HttpServletResponse response)  注册  protected void doGet(HttpServletRequest request, HttpServletResponse response)  非法 400</pre>
数据结构 与算法	

补充说明	
------	--

### 4.3.35. 模块 servlet- UnlikeServlet

模块名称	UnlikeServlet
功能描述	url /unlike POST 取消标记为喜爱的歌曲
接口 与属性	protected void doPost(HttpServletRequest request, HttpServletResponse response)  取消标记为喜爱的歌曲  protected void doGet(HttpServletRequest request, HttpServletResponse response)  取消标记为喜爱的歌曲
数据结构 与算法	
补充说明	

### 4.3.36. 模块 util-DatabaseUtil

模块名称	DatabaseUtil
功能描述	数据库工具类
接口 与属性	public static String URL;  数据库 URL  public static String USERNAME;

	<p>数据库的用户名</p> <pre>public static String PASSWORD;</pre> <p>数据库的密码</p> <pre>public static String DRIVER;</pre> <p>数据库的驱动</p> <pre>private static ResourceBundle resourceBundle</pre> <p>数据库配置文件</p> <pre>public static Connection getConnection()</pre> <p>获取数据库的联机</p> <pre>public static void close(ResultSet resultSet, Statement statement, Connection connection)</pre> <p>关闭数据库的连接</p>
数据结构 与算法	
补充说明	

### 4.3.37. 模块 util-JsonReader

模块名称	JsonReader
功能描述	Json 解析工具类
接口 与属性	public static JSONObject receivePost(HttpServletRequest request)

	解析 Json 为 JSONObject
数据结构 与算法	
补充说明	

#### 4.3.38. 模块 util-MD5Util

模块名称	MD5Util
功能描述	MD5 工具类
接口 与属性	<pre>private static final String[] hexDigits</pre> <p>所有的 MD5 字符选择</p> <pre>public static String MD5Encode(String origin, String charset)</pre> <p>编码为 MD5</p> <pre>public static String byteArrayToHexString(byte[] b)</pre> <p>byte 数组转为字符串</p> <pre>public static String MD5Encode(File file)</pre> <p>文件编码为 MD5</p> <pre>public static String byteToHexString(byte b)</pre> <p>byte 转为字符串</p>
数据结构 与算法	
补充说明	

## 5. 数据库设计

### 5.1. 数据库环境说明

数据库 : MySQL

连接数据库 jar 包 : mysql-connector-java-8.0.18.jar

设计工具 : OmniGraffle

数据库工具 : DataGrip、 Navicat

编程工具 : IntelliJ IDEA

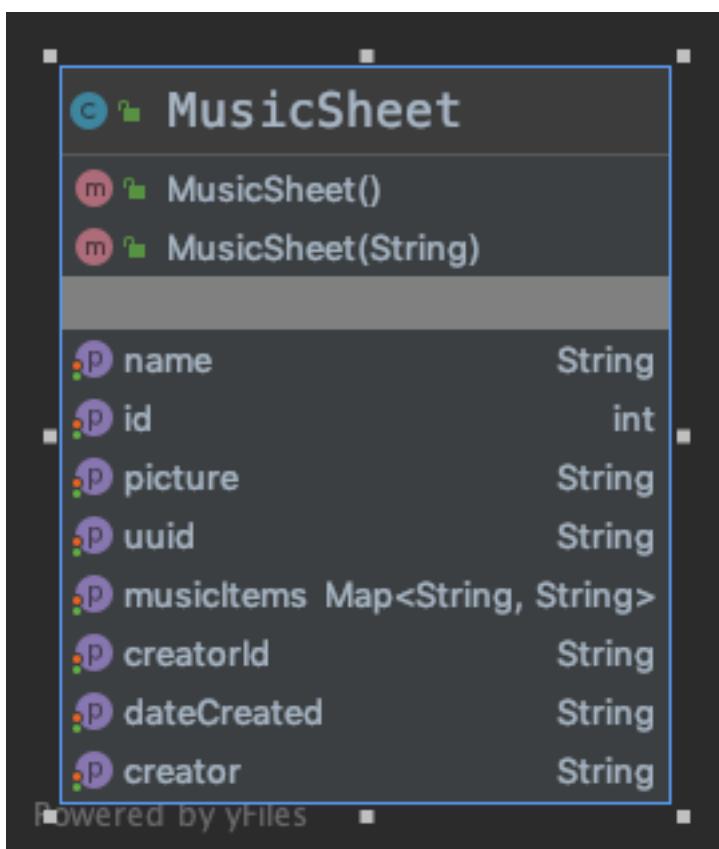
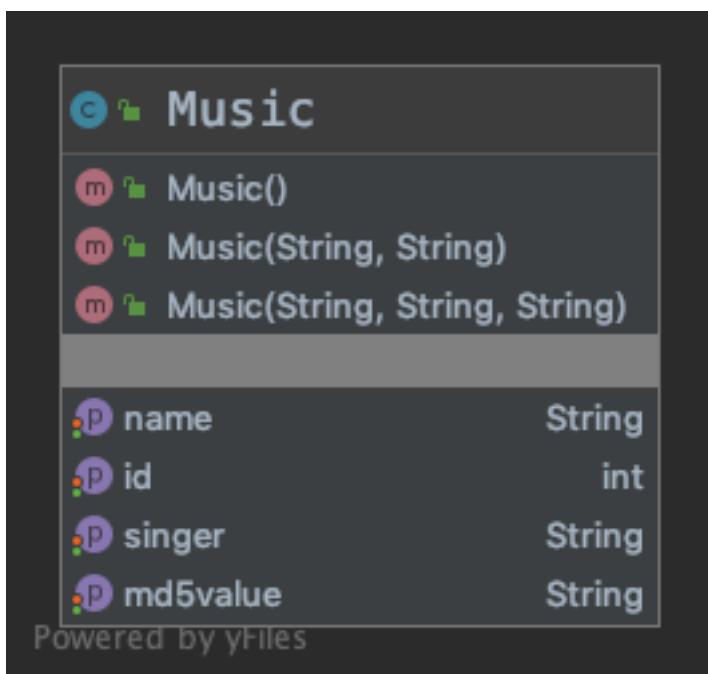
测试连接工具 : IntelliJ IDEA 内置 DataBase 工具 , 配合 mysql-connector-java.jar

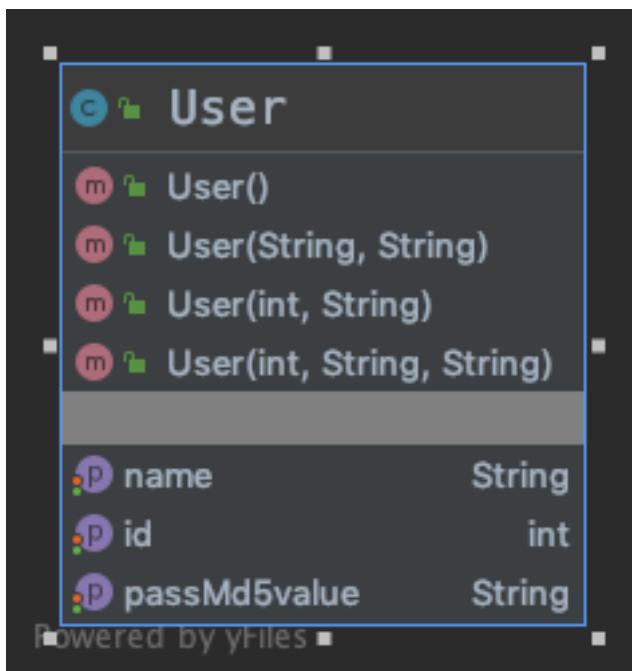
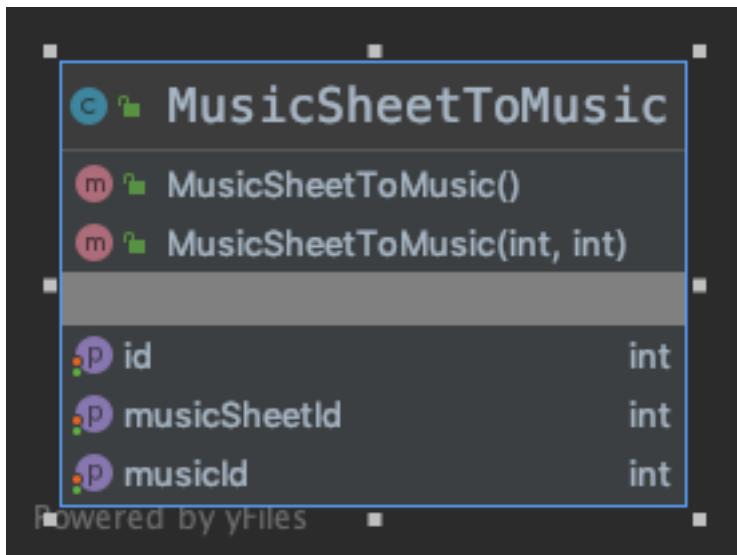
本项目数据库名 : musicserver

本项目数据库所含表名 : music、 musicsheet、 musicsheet\_music、 liked\_music、 user、 comment

本项目数据库编码格式 : UTF-8

## 5.2. 逻辑设计





数据库 music 表与 Music 类属性对应关系

`Id` → `Music.id`

`Name` → `Music.name`

`Md5value` → `Music.md5value`

`Singer` → `Music.singer`

数据库 musicsheet 表与 MusicSheet 类属性对应关系

Id → MusicSheet.id

Uuid → MusicSheet.uuid

Name → MusicSheet.nname

CreatorId → MusicSheet.creatorId

Creator → MusicSheet.creator

Picture → MusicSheet.picture

数据库 musicsheet\_music 表与 MusicSheetToMusic 类属性对应关系

Id → MusicSheetToMusic.id

musicsheetId → MusicSheetToMusic.musicsheetId

MusicId → MusicSheetToMusic.musicId

数据库 user 表与 User 类属性对应关系

Id → User.id

Name → User.name

Passmd5value → User.passMd5Value

## 5.3. 物理设计

### 5.3.1. 表汇总

表名	功能说明
music	歌曲信息表
musicsheet	歌单信息表

## 《爱音乐》

musicsheet_music	歌单歌曲对应关系表		
liked_music	标记 Like 表		
user	用户表		
comment	评论表		

### 5.3.2. 表 music

表名	music		
列名	数据类型 ( 精度范围 )	空/非空	约束条件
id	UNSIGNED INT	非空	主键
md5value	CHAR(255)	非空	Music.md5value
name	CHAR(255)	空	Music.name
singer	CHAR(255)	空	Music.singer
like	INT	非空	default 0
补充说明	<ul style="list-style-type: none"><li>● Id 歌曲自增主键</li><li>● Md5value 歌曲 MD5 值</li><li>● Name 歌曲名</li><li>● Singer 歌曲歌手</li><li>● Like 播放次数</li></ul>		

### 5.3.3. 表 musicsheet

表名	musicsheet		
列名	数据类型 ( 精度范围 )	空/非空	约束条件

## 《爱音乐》

id	UNSIGNED INT	非空	主键
uuid	CHAR(255)	非空	MusicSheet.uuid
name	CHAR(255)	空	MusicSheet.name
creatorId	CHAR(255)	空	MusicSheet.creatorId
creator	CHAR(255)	空	MusicSheet.creator
dateCreated	CHAR(255)	空	MusicSheet.dateCreated
picture	CHAR(255)	空	MusicSheet.picture
补充说明	<ul style="list-style-type: none"><li>● Id 歌单自增主键</li><li>● Uuid 歌单的 MD5 值</li><li>● Name 歌单名</li><li>● creatorId 歌单的创建者 id</li><li>● creator 歌单的创建者</li><li>● dateCreated 歌单的创建时间</li><li>● picture 歌单封面</li></ul>		

### 5.3.4. 表 musicsheet\_music

表名	musicsheet_music		
列名	数据类型 ( 精度范围 )	空/非空	约束条件
id	UNSIGNED INT	非空	主键
musicsheetId	UNSIGNED INT	非空	MusicSheet.id
musicId	UNSIGNED INT	非空	Music.id

# 《爱音乐》

补充说明	<ul style="list-style-type: none"><li>● Id 歌单歌曲对应自增 ID</li><li>● musicsheetId 歌单 id</li><li>● musicId 歌曲 id</li></ul>
------	---

## 5.3.5. 表 like\_music

表名	like_music		
列名	数据类型 ( 精度范围 )	空/非空	约束条件
user	UNSIGNED INT	非空	User.id
music	UNSIGNED INT	非空	Music.id
补充说明	<ul style="list-style-type: none"><li>● User 用户 id</li><li>● Music 歌曲 id</li></ul>		

## 5.3.6. 表 user

表名	user		
列名	数据类型 ( 精度范围 )	空/非空	约束条件
id	UNSIGNED INT	非空	主键
name	CHAR(255)	空	User.username
passmd5value	CHAR(255)	空	User.passmd5value
补充说明	<ul style="list-style-type: none"><li>● Id 自增主键</li><li>● Name 用户名</li><li>● Passmd5value 密码的 MD5 值</li></ul>		

## 5.3.7. 表 comment

表名	comment
----	---------

## 《爱音乐》

---

列名	数据类型 ( 精度范围 )	空/非空	约束条件
user	UNSIGNED INT	非空	User.id
music_sheet	UNSIGNED INT	非空	MusicSheet.id
content	CHAR(255)	空	Default ""
补充说明	<ul style="list-style-type: none"><li>● User 用户 id</li><li>● Music_sheet 歌单 id</li><li>● Content 评论内容</li></ul>		

## 6. 用户界面设计

### 6.1. 界面的关系图和工作流程图

登录界面如下



## 《爱音乐》

点击 我喜欢 查看当前用户喜欢的歌曲

The screenshot shows a user profile for '一只小码农' (A Little Programmer) with a blue cat icon. Below it, a message says 'Hi,来看看哪些歌是你的最爱'. A table lists three songs:

Name	MD5	Operation
飘向远方.mp3	7993fdff33153cf32e42e2f441502a0a	PLAY ❤️
哈哈哈.m4a	cc9d71fe0820fe8fc2b4dece9cd01d8d	PLAY ❤️
阴天快乐.m4a	a1aa02bb57f1bdf722e06c29a5be07d5	PLAY ❤️

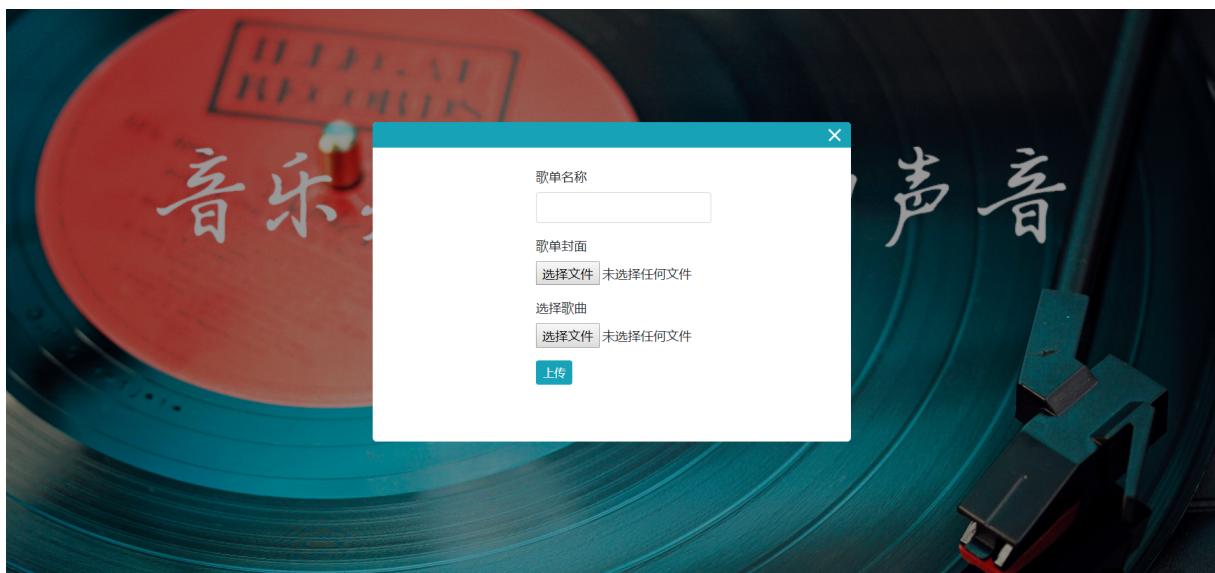
At the bottom, there is a playback bar showing 0:00 / 3:58.

点击 创建歌单 可以上传歌曲

The screenshot shows a red vinyl record with the text 'RECORDS' and '音乐是思维着的声音' (Music is a thinking sound). Below the record, a button says '点击上传' (Click to upload).

点击 点击上传 按钮后

## 《爱音乐》



即可上传自己的歌单

点击播放排名可以查看播放量高的几首歌曲

The screenshot shows a music application interface with a teal header bar containing a logo, the text '爱音乐', and navigation links: '歌单', '我喜欢', '创建歌单', and '播放排名'. Below the header is a list of five songs with their file extensions and playback progress bars:

歌曲名	文件类型	播放进度
飘向远方.mp3	.mp3	100%
哈哈哈.m4a	.m4a	~80%
阴天快乐.m4a	.m4a	~70%
Hello.mp3	.mp3	~60%
Monica.mp3	.mp3	~50%

## 附录：用户手册

登录/注册 操作如下：

**用户登录**

用户名

密码

**登录** **没有账户** **取消**

输入用户名和密码，点击登录按钮，通过验证后进入网站主界面；点击没有帐户则跳转到注册界面

填写用户名，输入两次密码即可完成注册

**用户注册**

用户名

密码

确认密码

**注册** **取消**

主界面（歌单界面）：

# 《爱音乐》

The screenshot shows a music application interface. At the top, there is a navigation bar with a logo and links for '歌单' (Playlists), '我喜欢' (I Like), '创建歌单' (Create Playlist), and '播放排名' (Playlist Ranking). Below the navigation bar, a song player is displayed for 'Soundwave'. The song player includes a play button, a progress bar showing '0:00 / 0:00', and a volume control icon. Below the player, a section titled 'Cherry的几首歌' (Cherry's songs) shows a list of three songs: '飘向远方.mp3', '哈哈哈.m4a', and '阴天快乐.m4a'. Each song entry includes its name, MD5 hash, and operation buttons for 'PLAY' and a heart icon. The bottom of the screen features a standard music player control bar with icons for play/pause, previous track, next track, and volume.

点击桃心按钮，可以变换按钮样式，代表用户喜欢/不喜欢这首歌曲

This screenshot shows a user's playlist titled 'Cherry的几首歌'. It includes creator information ('Creator: Cherrison' and 'Date created: 2019-12-22 15:58:54'). Below this, a list of three songs is shown: '飘向远方.mp3', '哈哈哈.m4a', and '阴天快乐.m4a'. Each song has a 'Name', 'MD5', and an 'Operation' column containing a 'PLAY' button and a heart icon. The heart icon is currently filled, indicating the song is liked. The bottom of the screen has a control bar with a play/pause button, a progress bar showing '0:00 / 3:58', and other standard media controls.

This screenshot shows another view of a user's playlist titled 'Cherry的几首歌'. It includes creator information ('Creator: Cherrison' and 'Date created: 2019-12-22 15:58:54'). Below this, a list of three songs is shown: '飘向远方.mp3', '哈哈哈.m4a', and '阴天快乐.m4a'. Each song has a 'Name', 'MD5', and an 'Operation' column containing a 'PLAY' button and a heart icon. The heart icon is currently unfilled, indicating the song is not liked. The bottom of the screen has a control bar with a play/pause button, a progress bar showing '0:00 / 3:58', and other standard media controls.

点击 PLAY 按钮播放音乐，可以通过底部栏中的音乐播放控制器进行控制

This screenshot shows a song player for a song that has just started playing, as indicated by the progress bar at '0:00 / 3:58'. The player includes a play button, a progress bar, and a volume control icon. The bottom of the screen features a standard music player control bar with icons for play/pause, previous track, next track, and volume.

点击栏目中的 我喜欢，查看用户喜欢的歌曲

## 《爱音乐》

The screenshot shows the 'Love Music' application interface. At the top, there is a navigation bar with the logo 'J 爱音乐' and links for '歌单', '我喜欢', '创建歌单', and '播放排名'. Below the navigation bar, a user profile section displays a circular icon with a blue cat-like character, the text '用户: 一只小码农', and a message 'Hi, 来看看哪些歌是你的最爱'. Underneath this, a table lists three songs in the library:

Name	MD5	Operation
飘向远方.mp3	7993fdff33153cf32e42e2f441502a0a	<button>PLAY</button>
哈哈哈.m4a	cc9d71fe0820fe8fc2b4dece9cd01d8d	<button>PLAY</button>
阴天快乐.m4a	a1aa02bb57f1bdf722e06c29a5be07d5	<button>PLAY</button>

At the bottom of the screen, there is a playback control bar with a play button, the text '0:00 / 3:58', and other standard media controls.

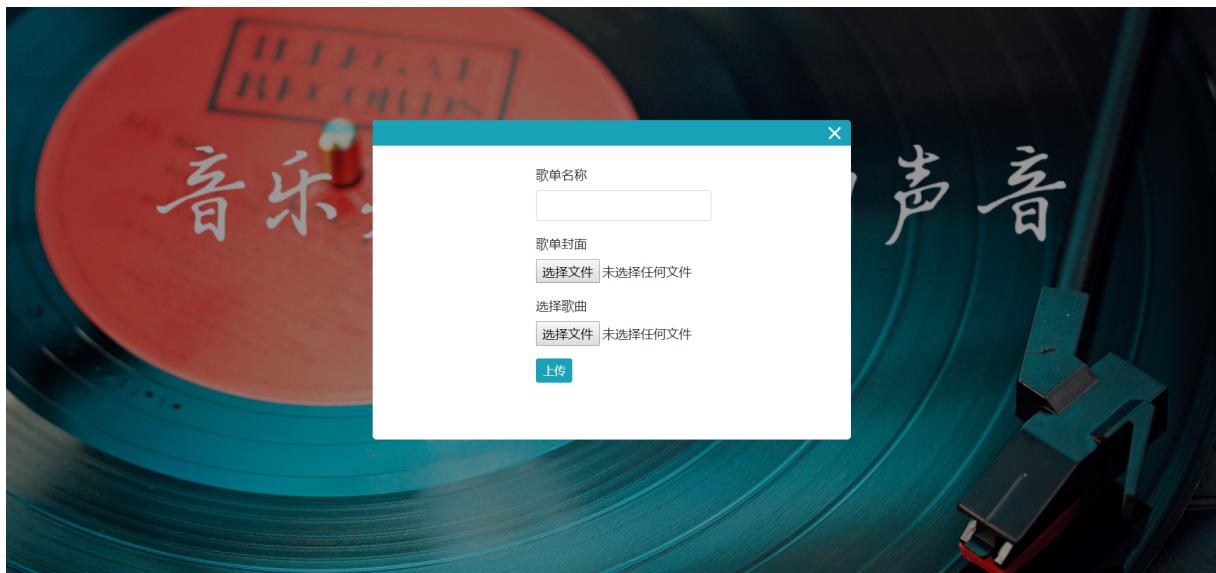
点击栏目中的 创建歌单，进入创建歌单界面

The screenshot shows the 'Create Playlist' interface. At the top, there is a navigation bar with the logo 'J 爱音乐' and links for '歌单', '我喜欢', '创建歌单', and '播放排名'. The main area features a large image of a vinyl record with a red label that has the Chinese text '音乐是思维着的声音'. Below the image, there is a button labeled '点击上传' (Click to Upload). The overall theme is musical and retro.

点击 点击上传按钮

可以填写自定义歌单名，在本地文件中选择歌单封面图片，以及在本地音乐中选择要上传的音乐

## 《爱音乐》



点击栏目中的 播放排名，查看播放量 top5 的歌曲

歌曲	播放进度
飘向远方.mp3	100%
哈哈哈.m4a	100%
阴天快乐.m4a	100%
Hello.mp3	100%
Monica.mp3	100%