

测试报告

测试报告

1 引言

1.1 目标

1.2 背景

1.3 范围

测试报告的适用范围

1.4 引用文件

2 测试计划

2.1 目的

测试活动范围

测试方法

资源

软件项

要测试的特征

测试任务及负责人

风险

2.2 详细说明

2.2.1 测试计划标识符

2.2.2 引言

软件项概述

软件特征

2.2.3 测试项

1. 音乐资源管理

2. 主页建设和信息管理

3. 歌曲播放

4. 审核与管理

影响范围

2.2.4 要测试的特征

1. 用户注册与登录

2. 音乐资源管理

4. 个人主页和信息管理

5. 社交互动

6. 歌曲播放

7. 审核与管理

8. 安全性与隐私保护

2.2.5 不要测试的特征

1. 第三方插件及工具的功能

2. 支付和交易功能

3. 非常规使用场景

4. 多语言支持

2.3 测试方法

1. 单元测试

2. 集成测试

3. 系统测试

4. 验收测试

测试充分性和结束准则

需求跟踪技术

2.4 测试通过准则

1. 功能测试通过准则
2. 用户体验测试通过准则
3. 性能测试通过准则
4. 集成测试通过准则
5. 系统测试通过准则
- 补充结束准则

2.5测试任务

1. 需求分析与测试计划制定
2. 测试用例设计
3. 测试环境搭建
4. 单元测试编写与执行
5. 集成测试编写与执行
6. 系统测试编写与执行
7. 用户体验测试
8. 回归测试
9. 测试报告编写

2.6环境要求

- 必要的特性
- 希望的特性

3 测试设计说明

3.1 目的

1. 详细规定测试方法
2. 标识要测试的特征

3.2 详细说明

3.2.1 测试设计说明标识符

3.2.2要测试的特征

1. 用户注册与登录
2. 音乐资源管理
3. 个人主页和信息管理
4. 歌曲播放
5. 审核与管理
6. 安全性
7. 性能和稳定性
8. 兼容性

3.2.3方法细化

1. 具体测试技术
2. 测试结果分析方法
3. 选择测试用例的依据
4. 测试用例的共性属性

3.2.4特征通过准则

1. 用户注册与登录
2. 音乐资源管理
3. 个人主页和信息管理
4. 歌曲播放
5. 审核与管理
6. 安全性
7. 性能和稳定性
8. 兼容性

4 测试用例说明

4.1 目的

1. 功能性测试用例
2. 性能测试用例

- 3. 兼容性测试用例
- 4. 安全性测试用例
- 5. 用户体验测试用例
- 6. 回归测试用例
- 4.2 详细说明
 - 4.2.1 测试用例说明标识符
 - 4.2.2 输入说明
 - 1. 用户注册与登录
 - 2. 音乐资源管理
 - 3. 个人主页和信息管理
 - 4. 歌曲播放
 - 5. 审核与管理
 - 6. 安全性
 - 7. 性能和稳定性
 - 8. 兼容性数据库、文件和系统消息
 - 4.2.3 输出说明
 - 1. 用户注册与登录
 - 2. 音乐资源管理
 - 3. 个人主页和信息管理
 - 4. 歌曲播放
 - 5. 审核与管理
 - 6. 安全性
 - 7. 性能和稳定性
 - 8. 兼容性
- 5 测试规程说明
 - 5.1 目的
 - 5.2 详细说明
 - 5.2.1 测试规程说明标识符
 - 5.2.2 特殊要求
 - 5.2.3 规程步骤
- 6 测试日志
 - 6.1 后端测试
 - 6.2 前端测试
 - 主体功能测试
 - 用户登录注册测试
 - 管理员测试
 - 消息测试
 - 注销模块
 - 用户信息修改模块
 - 用户关注和取消关注模块
 - 播放歌曲模块
 - 上传歌曲模块
 - 修改歌曲模块
 - 删除歌曲模块
 - 创建歌单模块
 - 查看歌单模块
 - 修改歌单模块
 - 删除歌单模块
 - 添加歌曲至歌单模块
 - 收藏歌单模块
 - 取消收藏歌单模块

添加喜欢的歌曲模块
取消喜欢的歌曲模块
关注模块
更新信息模块
上传信息模块
听歌周报和创作周报模块
评论消息模块
投诉歌曲或歌单模块
申诉模块
查看和处理投诉信息模块
查看和处理申诉信息模块
添加评论模块
查看评论模块
删除评论模块

7 测试总结报告

- 7.1 测试结果
- 7.2 性能测试
- 7.3 安全性测试
- 7.4 兼容性测试
- 7.5 测试评价

1 引言

1.1 目标

1.2 背景

为了缓解学生在学习生活中的压力，并促进各类原创音乐的分享与交流，"iMusic"在线音乐分享平台应运而生。该平台旨在为用户提供一个开放且互动的音乐社区，用户不仅可以上传和管理自己的音乐作品，还可以浏览和欣赏其他用户的音乐作品。平台还特别注重用户的隐私保护和操作便利性，力求为用户提供一个安全、便捷和丰富的使用体验。

为了确保"iMusic"平台能够顺利上线，并且在使用过程中能够提供稳定、高效且安全的服务，对其进行全面且细致的测试是必不可少的。测试工作的主要目的包括：

- 验证功能实现：**确保平台的各项功能，如音乐资源管理、主页建设与信息管理、歌曲播放、审核与管理等均能按照需求文档中描述的功能要求正常运行。
- 用户体验保障：**通过测试用户界面的交互性和易用性，确保用户在使用过程中能获得流畅且满意的体验。
- 隐私及安全性：**验证平台在处理用户数据和隐私保护方面的措施，确保用户信息的安全。
- 稳定性和性能：**通过负载测试和性能测试，确保平台在高并发情况下仍能稳定运行，并且响应迅速。
- 发现并修复缺陷：**在测试过程中发现系统中的缺陷和漏洞，并及时修复，以提高平台的整体质量和可靠性。

通过以上测试，"iMusic"平台将能够以最佳状态服务于用户，为广大音乐爱好者提供一个优质的音乐分享与交流平台。

1.3范围

测试报告的适用范围

本测试报告适用于“iMusic”在线音乐分享平台的所有模块及其相关功能。本报告涵盖了以下主要测试内容：

1. 功能测试：

- 音乐资源管理模块：包括歌曲上传、管理、展示、推荐、检索、分类、歌单管理、歌词文件定制与导出等功能的测试。
- 主页建设和信息管理模块：包括个人主页编辑、个人信息管理、收藏夹管理、歌单分享、消息提醒、关注用户、批量导入收藏等功能的测试。
- 歌曲播放模块：包括歌曲信息查看、歌曲播放、播放控制、评论编辑、历史播放记录查看、听歌周报查看等功能的测试。
- 审核与管理模块：包括歌曲和歌单的投诉、审查、申诉、删除、取消分享、消息提醒等功能的测试。

2. 用户体验测试：

- 用户界面交互性测试：确保界面设计符合用户操作习惯，交互流程流畅且合理。
- 操作便捷性测试：确保平台各功能的操作简单易懂，用户能够轻松上手。

3. 兼容性测试：

- 浏览器兼容性测试：确保平台在不同浏览器（如Chrome、Firefox、Safari、Edge等）上的兼容性和表现一致性。
- 设备兼容性测试：确保平台在不同设备（如PC、手机、平板等）上的兼容性和表现一致性。

本测试报告为“iMusic”平台的全面测试提供了详细的记录和分析，适用于项目团队的开发、测试、运维人员，以及其他相关利益方。

1.4 引用文件

2 测试计划

2.1目的

测试活动范围

本次测试活动的范围涵盖“iMusic”在线音乐分享平台的所有功能模块，包括但不限于音乐资源管理、主页建设和信息管理、歌曲播放、审核与管理等。具体测试内容包括功能测试、用户体验测试、安全性测试、性能和稳定性测试、兼容性测试等多个方面。

测试方法

- 功能测试**：采用黑盒、白盒测试方法，主要测试平台各功能模块的实现情况，确保其符合需求文档的描述。
- 用户体验测试**：通过实际用户操作和使用情况进行测试，评估用户界面的友好性和操作的便捷性。
- 安全性测试**：采用渗透测试和漏洞扫描等方法，确保用户数据的隐私和安全。
- 性能和稳定性测试**：通过负载测试工具模拟高并发场景，评估平台在不同负载下的性能表现。

5. **兼容性测试**：在多种浏览器和设备上进行测试，确保平台的兼容性和一致性。

资源

- **测试人员**：软工开发小组的所有人员。
- **测试环境**：配置多个测试环境，包括不同版本的浏览器、操作系统和设备，以覆盖各种可能的使用场景。
- **测试工具**：主要使用django的单元测试、集成测试，JMeter（性能测试）、Burp Suite（安全性测试）、Selenium（功能测试）。

软件项

被测试的软件项为“iMusic”在线音乐分享平台，包括：

- 用户端Web应用程序
- 管理端Web应用程序
- 后台服务与数据库

要测试的特征

- 音乐资源管理功能
- 用户个人主页及信息管理功能
- 歌曲播放功能
- 投诉与审核管理功能
- 用户隐私设置
- 需求文档中描述的其他功能
- 平台性能与稳定性
- 多浏览器和多设备兼容性

测试任务及负责人

- **需求分析和测试计划制定**：尹耀
- **功能测试**
 - **前端**：牛梓涵
 - **后端**：卢思文
- **用户体验测试**：薛惠天，张旖霜
- **兼容性测试**：尹耀
- **测试报告编写**：卢思文，牛梓涵

风险

1. **时间风险**：若测试过程中发现大量缺陷，可能影响项目进度。
2. **资源风险**：测试人员或环境不足可能导致测试覆盖不全。
3. **技术风险**：某些功能模块可能因技术实现复杂性导致测试困难。

4. **需求变更风险**：需求变更可能导致测试计划调整，增加工作量。

2.2详细说明

2.2.1 测试计划标识符

为了确保测试计划在项目管理过程中的唯一性和可追溯性，我们为“iMusic”在线音乐分享平台的测试计划分配了以下唯一标识符：

IMUSIC-TEST-PLAN-2024-V1.0

- **IMUSIC**: 表示该测试计划所属的项目名称为“iMusic”在线音乐分享平台。
- **TEST-PLAN**: 表示该文档类型为测试计划。
- **2024**: 表示测试计划制定的年份。
- **V1.0**: 表示测试计划的版本号，当前为第一个正式版本。

该标识符将用于所有与该测试计划相关的文档、沟通和追踪记录中，以确保所有相关人员能够准确地引用和定位本测试计划。

2.2.2引言

本测试报告旨在概述和评估“iMusic”在线音乐分享平台的测试过程和结果。该平台是一个为学生和音乐爱好者提供的在线音乐分享和交流社区，旨在缓解学习压力并促进原创音乐的传播和交流。以下是对要测试的软件项和其主要特征的概述：

软件项概述

iMusic 在线音乐分享平台包括以下主要组成部分：

- **用户端Web应用程序**：面向普通用户，提供音乐上传、管理、播放、分享和社交互动等功能。
- **管理端Web应用程序**：面向平台管理员，提供歌曲和歌单的审核、投诉处理和用户管理等功能。
- **后台服务与数据库**：支持平台的所有业务逻辑处理和数据存储。

软件特征

1. 音乐资源管理

- 用户可以登录平台后上传歌曲并管理自己上传的歌曲。
- 所有歌曲均为公开，平台展示歌曲并提供推荐、检索、分类功能。
- 用户可以定制和导出歌词文件，并收到平台定期发送的创作周报。

2. 主页建设和信息管理

- 每个用户有个人主页，能够编辑个人信息和管理喜欢的歌曲及收藏夹。
- 用户可以将收藏夹分享为歌单，并查看和关注其他用户的主页。
- 平台会提醒用户关注的用户的更新，用户可以从歌单批量导入歌曲到收藏夹。

3. 歌曲播放

- 访客可以查看歌曲信息、播放歌曲，并进行各种播放控制。
- 登录用户可以在歌曲播放页面编辑评论，查看历史播放记录及听歌周报。

4. 审核与管理

- 用户可以投诉平台中的歌曲和歌单，平台管理端会收到消息并进行审查。
- 被投诉者可以申诉，管理端可以删除歌曲、取消分享歌单并进行消息通知。

2.2.3 测试项

本次测试涵盖“iMusic”在线音乐分享平台的所有关键功能模块和相关特性。每个测试项均标识其版本和修订级别，并在测试开始之前明确影响范围及相应要求。以下是详细的测试项列表：

1. 音乐资源管理

版本/修订级别: v1.0

特性和要求:

- 用户登录后可以上传和管理自己的歌曲。
- 平台展示所有公开的歌曲，支持推荐、检索和分类功能。
- 提供定制和导出歌词文件功能。
- 平台定期发送创作周报。

逻辑/物理变换要求:

- 歌曲展示页面需根据用户喜好推荐。
- 歌词文件的导出格式需符合行业标准（如LRC）。

2. 主页建设和信息管理

版本/修订级别: v1.0

特性和要求:

- 每个注册用户有自己的个人主页，可以编辑个人信息。
- 用户能管理喜欢的歌曲和收藏夹，分享为歌单。
- 用户可以查看和关注其他用户的主页，并收到关注用户的更新提醒。
- 从歌单中批量导入歌曲到收藏夹的功能。

逻辑/物理变换要求:

- 个人主页的编辑在保存后应即时更新。
- 分享功能需支持多种社交媒体平台。
- 消息提醒功能需确保及时准确。

3. 歌曲播放

版本/修订级别: v1.0

特性和要求:

- 访客可以查看歌曲信息、播放歌曲并进行播放控制。
- 登录用户可以在歌曲播放页面编辑评论。
- 查看历史播放记录及听歌周报。

逻辑/物理变换要求:

- 播放器需支持常见的播放控制（如播放、暂停、跳转）。
- 听歌周报的生成需基于用户的实际播放数据。

4. 审核与管理

版本/修订级别: v1.0

特性和要求:

- 平台中的歌曲和歌单可以被投诉。
- 管理端收到投诉消息后进行审查。
- 被投诉者可以进行申诉。
- 管理端可执行删除歌曲、取消分享歌单的操作及其逆操作，并通知相关用户。

逻辑/物理变换要求:

- 投诉流程需简洁明了，并确保所有投诉记录完整。
- 管理端的操作需记录日志，以备审计。

影响范围

1. **功能实现**: 确保所有功能按照需求文档正确实现，并且没有功能缺失或错误。
2. **用户体验**: 确保用户界面友好，操作便捷。
3. **数据安全**: 确保用户数据和隐私得到有效保护。
4. **兼容性**: 确保系统在不同浏览器和设备上的兼容性。

2.2.4 要测试的特征

本次测试将涵盖“iMusic”在线音乐分享平台的所有主要功能特征及其组合，确保每个特征在独立和组合情况下均能正常运行。以下是详细的测试特征列表：

1. 用户注册与登录

测试特征:

- 用户可以进行注册，并通过邮箱验证激活账号。
- 用户可以使用注册的邮箱和密码登录平台。

组合特征:

- 测试在不同浏览器和设备上的注册和登录功能。

2. 音乐资源管理

测试特征:

- 用户上传歌曲功能。
- 用户管理已上传歌曲功能，包括修改、删除和查看歌曲详细信息。
- 平台展示所有公开歌曲功能，包括推荐、检索、分类等。

组合特征:

- 测试歌曲上传后能否正确展示，并能被其他用户检索到。
- 测试歌曲管理功能在不同操作系统和浏览器上的表现。

4. 个人主页和信息管理

测试特征:

- 用户可以编辑个人主页和个人信息。
- 用户可以管理喜欢的歌曲和收藏夹。
- 用户可以将收藏夹中的歌曲分享为歌单。

组合特征:

- 测试个人主页编辑后信息的实时更新情况。
- 测试收藏夹和歌单管理功能在不同浏览器上的一致性。

5. 社交互动

测试特征:

- 用户可以查看其他用户的主页，并进行关注。
- 用户可以收到所关注用户的更新提醒。
- 用户可以将歌单中的歌曲批量导入自己的收藏夹。

组合特征:

- 测试关注功能和更新提醒功能在高并发情况下的性能表现。

6. 歌曲播放

测试特征:

- 访客可以查看歌曲信息并播放歌曲。
- 用户可以进行播放控制（播放、暂停、跳转等）。
- 登录用户可以在播放页面编辑评论。
- 用户可以查看历史播放记录和听歌周报。

组合特征:

- 测试歌曲播放功能在不同网络环境下的流畅性。
- 测试评论功能在不同浏览器和设备上的表现。

7. 审核与管理

测试特征:

- 用户可以对歌曲和歌单进行投诉。
- 管理端收到投诉后进行审查。
- 被投诉者可以进行申诉。
- 管理端可以执行删除歌曲和取消分享歌单的操作，并通知相关用户。

组合特征:

- 测试投诉处理流程的完整性和消息通知的及时性。
- 测试管理端操作在不同权限级别下的表现。

8. 安全性与隐私保护

测试特征:

- 确保用户可以进行隐私设置。
- 确保用户隐私得到有效保护，防止未经授权的访问。

组合特征:

- 测试在不同网络环境下的数据加密和传输安全性。

2.2.5 不要测试的特征

在本次测试计划中，以下特征和特征组合将不纳入测试范围，具体理由如下：

1. 第三方插件及工具的功能

特征描述:

- 使用第三方插件或工具：jieba分词、哈工大停用词库用于处理搜索关键词，colorthief处理歌曲背景渐变色

理由:

- “iMusic”平台的设计目标是独立提供所有必要的功能，不依赖于第三方插件或工具。因此，测试重点将放在平台本身的功能实现和稳定性上，而不是外部工具的兼容性。

2. 支付和交易功能

特征描述:

- 用户打赏等交易相关功能。

理由:

- 目前平台暂未涉及任何形式的商业交易或支付功能，主要以免费分享和交流为主。因此，这部分内容不在本次测试范围内。

3. 非常规使用场景

特征描述:

- 平台在极端或非常规使用场景下的表现，例如长时间高负荷连续播放、同时进行大量并发上传操作等。

理由:

- 本次测试将以常规使用场景为主，确保平台在正常使用情况下的功能和性能表现。极端或非常规使用场景的测试将在后续阶段视情况而定。

4. 多语言支持

特征描述:

- 平台的多语言支持和本地化功能。

理由:

- 当前版本的“iMusic”平台仅支持简体中文，暂未计划推出其他语言版本。因此，多语言支持和本地化功能不在本次测试范围内。

2.3测试方法

本次测试将采用综合测试方法，包括单元测试、集成测试和用户验收测试，以确保“iMusic”在线音乐分享平台的各个特征组得到充分测试。以下是详细的测试方法描述：

1. 单元测试

主要活动、技术和工具:

- **活动:** 针对平台的各个独立功能模块，编写和执行单元测试，确保每个模块功能的正确性。
- **技术:** 使用Django的内置单元测试框架，对后端逻辑和数据库操作进行测试。
- **工具:** Django测试框架。

测试任务及时间估计:

- 编写单元测试用例（1周）
- 执行单元测试（1周）

测试充分性和结束准则:

- 确保每个函数和方法至少执行一次，并覆盖所有关键路径。
- 测试所有关键功能均无重大缺陷。

2. 集成测试

主要活动、技术和工具:

- **活动:** 对各功能模块进行集成测试，验证模块之间的接口和交互。
- **技术:** 逐步集成各模块，执行集成测试用例，确保模块间的协调工作。
- **工具:** Django测试框架。

测试任务及时间估计:

- 编写集成测试用例（1周）
- 执行集成测试（1周）

测试充分性和结束准则:

- 确保所有模块间的接口均被调用并正确响应。
- 所有集成测试用例通过，且无重大集成缺陷。

3. 系统测试

主要活动、技术和工具:

- **活动:** 对整个系统进行全面测试, 验证系统在真实环境下的表现。
- **技术:** 执行全系统测试用例, 覆盖功能测试、性能测试、安全性测试等。
- **工具:** JMeter (性能测试)、Burp Suite (安全性测试)、Selenium (功能测试)。

测试任务及时间估计:

- 编写系统测试用例 (1周)
- 执行系统测试 (1周)

测试充分性和结束准则:

- 确保所有系统功能均被测试并无重大缺陷。
- 系统性能达到预期标准, 在高并发情况下稳定运行。
- 安全性测试无重大漏洞。

4. 验收测试

主要计划:

- **计划:** 模拟实际用户进行测试, 收集反馈并进行改进。

测试任务及时间估计:

- 准备用户验收测试环境 (1周)
- 执行用户验收测试 (1周)

测试充分性和结束准则:

- 根据反馈进行必要的调整和改进。

测试充分性和结束准则

最低测试充分性:

- 确保每个功能模块和关键路径至少执行一次。
- 通过覆盖率工具确保测试覆盖率达到90%以上。

补充结束准则:

- 出错频率控制在合理范围内, 单个测试环节的错误率不超过5%。
- 所有高优先级和中优先级缺陷均已修复并通过回归测试。

需求跟踪技术

技术:

- 使用需求跟踪矩阵, 将测试用例与需求文档中的功能点进行对应。
- 使用JIRA或类似的项目管理工具, 记录需求、测试用例和缺陷, 并进行追踪。

2.4测试通过准则

本部分规定用于确定每个测试项是否通过测试或测试失败的准则。为了确保“iMusic”在线音乐分享平台的高质量和稳定性，每个测试项必须满足以下通过准则。

1. 功能测试通过准则

- **所有功能按预期工作**：测试用例中的所有功能操作均按需求文档的描述正确执行，无任何功能缺失或错误。
- **无重大缺陷**：测试过程中未发现任何高优先级或阻塞性缺陷。所有发现的中优先级和低优先级缺陷均已修复或记录在案，并计划在后续版本中解决。
- **测试用例全部通过**：功能测试用例的通过率达到100%，所有关键路径和场景均被覆盖。

2. 用户体验测试通过准则

- **界面友好**：用户界面美观，操作简单，符合用户使用习惯。
- **交互流畅**：所有用户交互操作响应迅速，无明显延迟或卡顿现象。

3. 性能测试通过准则

- **响应时间**：在预期负载下，系统的平均响应时间小于2秒，高峰负载下不超过5秒。
- **资源利用率**：CPU、内存、网络等资源利用率在合理范围内，无资源瓶颈。

4. 集成测试通过准则

- **模块接口正常**：所有模块接口按预期工作，无数据传输错误或异常。
- **无集成缺陷**：集成测试过程中未发现任何高优先级或中优先级缺陷。

5. 系统测试通过准则

- **整体功能正常**：系统所有功能模块均能正常工作，互相之间协调运行，无冲突或不兼容现象。
- **无重大系统缺陷**：系统测试过程中未发现任何高优先级或阻塞性缺陷。

补充结束准则

- **出错频率**：测试过程中，每100个测试用例中的错误数不超过5个。
- **回归测试通过**：所有修复的缺陷均已通过回归测试，确保没有引入新的缺陷。
- **覆盖率**：代码覆盖率达到90%以上，关键业务逻辑覆盖率达到100%。

2.5测试任务

1. 需求分析与测试计划制定

- **任务描述**：分析需求文档，制定详细的测试计划，明确测试目标和范围。
- **依赖关系**：无，作为整个测试过程的起点。
- **特殊技能**：需求分析、测试计划制定。

2. 测试用例设计

- **任务描述:** 根据需求文档设计详细的测试用例，涵盖所有功能和特征组合。
- **依赖关系:** 需求分析与测试计划制定完成后开始。
- **特殊技能:** 测试用例设计、需求理解。

3. 测试环境搭建

- **任务描述:** 搭建测试所需的硬件和软件环境，确保测试环境与实际使用环境一致。
- **依赖关系:** 测试用例设计阶段进行，以便及时验证环境配置。
- **特殊技能:** 环境配置、系统管理。

4. 单元测试编写与执行

- **任务描述:** 编写并执行单元测试，验证每个独立模块的功能正确性。
- **依赖关系:** 测试环境搭建完成后开始。
- **特殊技能:** 编程、单元测试编写、Django测试框架使用。

5. 集成测试编写与执行

- **任务描述:** 编写并执行集成测试，验证模块间的接口和交互。
- **依赖关系:** 单元测试完成后开始。
- **特殊技能:** 集成测试设计、接口测试、Django测试框架使用。

6. 系统测试编写与执行

- **任务描述:** 编写并执行系统测试，全面验证系统在真实环境下的表现。
- **依赖关系:** 集成测试完成后开始。
- **特殊技能:** 系统测试设计、Selenium使用、JMeter使用、Burp Suite使用。

7. 用户体验测试

- **任务描述:** 模拟实际用户进行测试，评估体验效果。
- **依赖关系:** 系统测试完成后开始。
- **特殊技能:** 研究用户习惯、反馈收集与分析。

8. 回归测试

- **任务描述:** 对修复的缺陷进行回归测试，确保修复有效且未引入新缺陷。
- **依赖关系:** 每轮测试发现并修复缺陷后进行。
- **特殊技能:** 回归测试设计、缺陷管理。

9. 测试报告编写

- **任务描述:** 编写详细的测试报告，总结测试过程和结果，为项目团队提供反馈和改进建议。
- **依赖关系:** 所有测试任务完成后进行。
- **特殊技能:** 测试报告编写、数据分析、技术文档撰写。

2.6环境要求

为了确保“iMusic”在线音乐分享平台的测试过程全面、高效且准确地进行，本部分详细说明测试环境的必要性和希望特性，包括硬件、通信、系统软件、使用方式及其他支持测试所需的软件或设备。同时，规定了这些测试设施、系统软件和专有组成部分所需的安全等级。

必要的特性

1. 硬件特性

- **服务器:**
 - 高性能服务器，用于部署和运行测试环境，包括开发服务器和生产模拟服务器。
 - 配置: 至少具备2核CPU，2GB内存，20GB SSD存储空间。
- **客户端设备:**
 - 多种设备类型以覆盖不同用户场景，包括桌面电脑、笔记本电脑、平板电脑。
 - 配置: 主流品牌和型号，分别运行不同操作系统（Windows、macOS、Linux、Android）。

2. 通信特性

- 高速互联网连接，确保测试过程中数据传输的稳定性和速度。
- 安全的VPN接入，以保证远程测试人员能够安全访问测试环境。

3. 系统软件特性

- **操作系统:**
 - 服务器端: Ubuntu 20.04 LTS 或其他企业级Linux发行版。
 - 客户端: 最新版本的Windows 10/11, macOS, 各主流Linux发行版，以及Android和iOS。
- **数据库:**
 - 使用MySQL 8或以上版本，用于存储和管理平台数据。
- **Web服务器:**
 - Nginx 或 Apache，作为反向代理和静态资源服务器。
- **应用服务器:**
 - 配置Django应用运行环境，包括必要的Python依赖库和包管理工具。

4. 支持软件

- **开发与测试工具:**
 - IDE: PyCharm、Visual Studio Code、WebStorm，用于编写和调试代码。
 - 版本控制系统: Git，使用华为CodeHub进行代码托管和协作。
 - 自动化测试工具: Selenium，用于Web界面自动化测试。
 - 性能测试工具: JMeter，用于负载和性能测试。

- 安全测试工具: Burp Suite, 用于安全性和渗透测试。
- **监控与日志系统:**
 - ELK Stack (Elasticsearch, Logstash, Kibana), 用于日志收集、存储和分析。
 - Prometheus 和 Grafana, 用于系统性能监控和指标展示。

5. 使用方式

- 测试环境需可独立使用, 以确保测试过程不受开发或生产环境的影响。
- 支持多用户同时访问和操作, 确保测试团队能够并行工作。

6. 安全等级

- 数据安全:
 - 所有测试数据应进行脱敏处理, 防止敏感信息泄露。
 - 测试环境中的数据传输应采用SSL/TLS加密。
- 访问控制:
 - 使用基于角色的访问控制 (RBAC), 确保只有授权人员能够进行特定操作。

希望的特性

1. 高可用性

- 测试环境应具备高可用性, 确保在测试期间内系统持续可用, 减少因环境问题导致的测试中断。

2. 自动化配置

- 使用容器化技术 (如Docker) 和配置管理工具 (如Ansible) 进行环境配置和管理, 提升环境搭建和维护的效率。

3. 备份和恢复

- 配置定期备份机制, 确保测试数据和配置的安全性, 并具备快速恢复的能力, 以应对意外情况。

3 测试设计说明

3.1 目的

1. 详细规定测试方法

通过测试设计, 我们将详细规定测试的具体方法, 以确保每个测试项都得到充分验证。测试方法包括但不限于以下几种:

- **单元测试:** 验证各个独立模块的功能正确性。使用Django的内置单元测试框架, 编写和执行针对具体函数和方法的测试用例。
- **集成测试:** 验证各模块之间的接口和交互。使用ApiFox和Django内置Client进行API测试, 确保各模块之间的数据传输和交互正常。
- **系统测试:** 对整个系统进行全面测试。使用Selenium进行自动化功能测试, 使用JMeter进行性能测试, 使用Burp Suite进行安全测试, 确保系统在各种场景下的表现。
- **用户体验测试:** 模拟、邀请实际用户进行测试, 收集反馈并进行分析, 评估用户界面的友好性和操作的便捷性。

- **兼容性测试**: 验证平台在不同浏览器和设备上的兼容性。使用多种设备和浏览器进行测试，确保平台在各种环境下的表现一致。
- **回归测试**: 在缺陷修复后进行测试，确保修复有效且未引入新缺陷。

2. 标识要测试的特征

测试设计将明确标识所有要测试的特征，确保每个功能模块和特征组合都得到充分验证。要测试的特征包括但不限于以下内容：

- **用户注册与登录**: 验证用户注册和登录功能，确保用户可以顺利注册、激活账号并登录平台。
- **音乐资源管理**: 验证用户上传、管理和展示歌曲的功能，确保平台支持多种音频格式，并能正确展示和分类歌曲。
- **歌词文件定制与导出**: 验证用户定制和导出歌词文件的功能，确保歌词文件符合行业标准。
- **个人主页和信息管理**: 验证用户编辑个人主页、管理喜欢的歌曲和收藏夹的功能，确保用户可以分享歌单并关注其他用户。
- **歌曲播放**: 验证歌曲播放功能，确保访客和登录用户均可顺利播放歌曲，并进行评论和查看历史记录。
- **审核与管理**: 验证平台的投诉处理和管理功能，确保投诉流程完整，相关消息通知及时。
- **安全性**: 验证平台在数据传输和存储过程中的安全性，确保用户数据和隐私得到有效保护。
- **性能和稳定性**: 验证平台在高并发访问情况下的性能和稳定性，确保系统响应迅速且运行稳定。
- **兼容性**: 验证平台在不同浏览器和设备上的兼容性，确保功能和界面的一致性。

3.2 详细说明

3.2.1 测试设计说明标识符

为了确保测试设计说明在项目管理过程中的唯一性和可追溯性，现为“iMusic”在线音乐分享平台的测试设计说明分配以下唯一标识符：

IMUSIC-TEST-DESIGN-2024-V1.0

- **IMUSIC**: 表示该测试设计说明所属的项目名称为“iMusic”在线音乐分享平台。
- **TEST-DESIGN**: 表示该文档类型为测试设计说明。
- **2024**: 表示测试设计说明制定的年份。
- **V1.0**: 表示测试设计说明的版本号，当前为第一个正式版本。

3.2.2 要测试的特征

本部分旨在标识“iMusic”在线音乐分享平台的测试项，并描述作为该设计说明对象的主要特征和特征组合。

1. 用户注册与登录

测试项: 验证用户注册和登录功能

特征和特征组合:

- 用户可以通过邮箱注册新账号，并接收激活邮件进行账号激活。
- 用户可以使用邮箱和密码登录平台。
- 密码重置功能，用户可以重置密码。

2. 音乐资源管理

测试项: 验证音乐上传、管理和展示功能

特征和特征组合:

- 用户可以管理已上传的歌曲，包括修改、删除和查看详细信息。
- 平台可以合理展示所有公开的歌曲，并提供推荐、检索和分类功能。
- 用户可以定制和导出歌词文件，格式符合行业标准（如LRC）。

3. 个人主页和信息管理

测试项: 验证个人主页编辑和信息管理功能

特征和特征组合:

- 用户可以编辑个人主页，包括个人信息、头像等。
- 用户可以管理自己喜欢的歌曲和收藏夹。
- 用户可以将收藏夹中的歌曲分享为歌单。
- 用户可以查看和关注其他用户的主页，并收到关注用户的更新提醒。

4. 歌曲播放

测试项: 验证歌曲播放功能

特征和特征组合:

- 访客可以查看歌曲信息，并进行播放控制（播放、暂停、跳转）。
- 登录用户可以在播放页面编辑评论。
- 用户可以查看自己的历史播放记录及听歌周报。

5. 审核与管理

测试项: 验证投诉处理和管理功能

特征和特征组合:

- 用户可以对平台中的歌曲和歌单进行投诉。
- 管理端可以接收投诉消息，并进行进一步审查。
- 被投诉者可以进行申诉。
- 管理端可以执行删除歌曲、取消分享歌单的操作，并通知相关用户。

6. 安全性

测试项: 验证平台的安全性

特征和特征组合:

- 确保用户数据在传输和存储过程中的安全性，所有敏感数据均加密处理。
- 验证平台的防护措施，确保无信息泄露和未经授权的访问。
- 进行安全性测试，发现并修复潜在的安全漏洞。

7. 性能和稳定性

测试项: 验证平台的性能和稳定性

特征和特征组合:

- 进行负载测试，确保系统在高并发情况下的响应速度和稳定性。
- 监控系统资源利用率，确保无性能瓶颈。
- 在不同网络环境下验证系统的稳定性，确保持续可用。

8. 兼容性

测试项: 验证平台的兼容性

特征和特征组合:

- 确保平台在主流浏览器（如Chrome、Firefox、Safari、Edge）上的表现一致。
- 验证平台在不同设备（如桌面电脑、笔记本电脑、平板电脑）上的兼容性。

3.2.3方法细化

本部分细化测试计划中描述的方法，包括具体测试技术、测试结果分析方法、选择测试用例的依据及共性测试用例的归纳。

1. 具体测试技术

1. 单元测试

- **技术:** 使用Django内置的单元测试框架。
- **目标:** 验证各个独立模块的功能正确性。
- **方法:** 编写针对每个函数和方法的单元测试用例，确保每个模块的业务逻辑正常。
- **结果分析:** 比较预期输出与实际输出，使用assert语句验证功能。

2. 集成测试

- **技术:** 使用Postman和Django内置Client进行API测试，使用Selenium进行界面交互测试。
- **目标:** 验证模块之间的接口和交互。
- **方法:** 编写API测试用例，模拟真实用户操作场景进行测试。
- **结果分析:** 检查API响应码、响应信息、响应时间和数据完整性；通过可视化审查确认界面交互的正确性。

3. 系统测试

- **技术:** 使用Selenium进行自动化功能测试，使用JMeter进行性能测试，使用Burp Suite进行安全测试。
- **目标:** 全面验证系统在真实环境下的表现。
- **方法:** 编写系统测试用例，覆盖所有功能模块和场景。
- **结果分析:** 通过日志和报告分析功能、性能和安全测试结果，比较预期与实际系统行为。

4. 用户体验测试

- **技术:** 使用问卷调查和用户反馈收集工具。

- **目标:** 评估用户界面的友好性和操作的便捷性。
- **方法:** 邀请实际用户进行测试, 收集用户反馈并进行分析。
- **结果分析:** 综合用户反馈, 提取共性问题和改进建议。

5. 兼容性测试

- **技术:** 使用多种设备和浏览器进行手动测试和自动化测试。
- **目标:** 验证平台在不同浏览器和设备上的兼容性。
- **方法:** 编写兼容性测试用例, 在多种环境下执行测试。
- **结果分析:** 比较不同环境下的测试结果, 确保一致性。

6. 性能测试

- **技术:** 使用JMeter进行负载测试。
- **目标:** 评估系统在高并发情况下的稳定性和响应速度。
- **方法:** 模拟大量用户访问场景, 进行负载和压力测试。
- **结果分析:** 通过性能监控工具分析系统资源利用率、响应时间和吞吐量。

7. 安全性测试

- **技术:** 使用Burp Suite进行安全扫描和渗透测试。
- **目标:** 发现并修复潜在的安全漏洞。
- **方法:** 进行常见安全测试, 如SQL注入、XSS、CSRF等。
- **结果分析:** 分析安全测试报告, 识别并修复高风险漏洞。

2. 测试结果分析方法

- **比较程序:** 使用自动化工具比较预期输出与实际输出, 生成对比报告。
- **可视化审查:** 通过日志和报告的可视化审查, 确认系统行为是否符合预期。
- **性能指标分析:** 使用监控工具分析性能测试结果, 确保各项指标在可接受范围内。
- **安全报告分析:** 通过分析安全扫描报告, 识别系统中的安全漏洞, 并制定修复计划。

3. 选择测试用例的依据

- **功能覆盖:** 确保所有功能模块和业务逻辑均有对应的测试用例。
- **边界条件:** 包含边界值分析, 验证系统在极端条件下的表现。
- **错误处理:** 测试系统对各种异常输入和操作的处理能力, 确保系统健壮性。
- **用户场景:** 基于用户需求和场景设计测试用例, 确保测试的实用性。

4. 测试用例的共性属性

- **输入约束:** 针对组相关测试用例的所有输入必须符合系统需求中的有效输入范围。
- **共享环境:** 测试用例需在统一的测试环境中执行, 确保环境的一致性和可复现性。
- **共享特殊规程:** 测试用例需遵循统一的测试流程和规范, 确保测试过程标准化。
- **依赖关系:** 确保测试用例按依赖关系顺序执行, 例如, 集成测试需在单元测试通过后进行, 系统测试需在集成测试通过后进行。

3.2.4特征通过准则

本部分规定了用于判定“iMusic”在线音乐分享平台各个特征或特征组合是否通过或失败的准则。通过这些准则，确保测试过程具有明确的评价标准，以保证平台的高质量交付。

1. 用户注册与登录

通过准则:

- 用户能够成功注册并收到激活邮件，激活链接有效且可以激活账号。
- 用户可以使用注册的邮箱和密码成功登录平台。
- 密码重置功能正常。

失败准则:

- 注册或激活过程中出现错误，用户无法完成注册或激活。
- 用户无法使用正确的邮箱和密码登录平台。
- 密码重置功能无法正常使用，重置密码失败。

2. 音乐资源管理

通过准则:

- 用户可以成功上传歌曲，并在平台上展示。
- 用户能够管理已上传的歌曲，包括修改、删除和查看详细信息。
- 平台能够正确推荐、检索和分类所有公开歌曲。
- 用户可以定制和导出符合行业标准的歌词文件。

失败准则:

- 用户无法上传歌曲或上传过程中出现错误。
- 用户无法管理已上传的歌曲，或管理操作（修改、删除、查看）失败。
- 平台推荐、检索和分类功能不准确或存在错误。
- 歌词文件无法定制或导出，或导出文件不符合标准。

3. 个人主页和信息管理

通过准则:

- 用户可以成功编辑个人主页和个人信息，并实时保存更新。
- 用户能够管理喜欢的歌曲和收藏夹，并将收藏夹中的歌曲分享为歌单。
- 用户可以查看和关注其他用户的主页，并收到关注用户的更新提醒。

失败准则:

- 个人主页和信息编辑过程中出现错误，用户无法保存更改。
- 用户无法管理喜欢的歌曲和收藏夹，或分享歌单失败。
- 用户无法查看或关注其他用户的主页，或未收到更新提醒。

4. 歌曲播放

通过准则:

- 访客可以成功查看歌曲信息，并正常播放歌曲。
- 登录用户可以进行播放控制（播放、暂停、跳转），并在播放页面编辑评论。
- 用户可以查看自己的历史播放记录及听歌周报。

失败准则:

- 访客无法查看歌曲信息或播放歌曲时出现错误。
- 登录用户的播放控制功能无法正常使用，或评论编辑失败。
- 用户无法查看历史播放记录或听歌周报。

5. 审核与管理

通过准则:

- 用户可以成功对平台中的歌曲和歌单进行投诉。
- 管理端能够接收投诉消息并进行审查。
- 被投诉者可以进行申诉，管理端可以执行删除歌曲、取消分享歌单的操作，并通知相关用户。

失败准则:

- 用户无法对歌曲和歌单进行投诉，或投诉消息无法提交。
- 管理端未能接收投诉消息或无法进行审查。
- 被投诉者无法进行申诉，管理端的删除或取消分享操作失败，或未通知相关用户。

6. 安全性

通过准则:

- 用户数据在传输和存储过程中均经过加密处理，无信息泄露。
- 平台通过安全性测试，未发现高风险安全漏洞。
- 所有发现的安全漏洞均已修复，并通过回归测试验证。

失败准则:

- 用户数据在传输或存储过程中存在泄露风险。
- 平台存在未修复的高风险安全漏洞。
- 修复的安全漏洞未通过回归测试。

7. 性能和稳定性

通过准则:

- 系统在预期负载下的平均响应时间小于2秒，高峰负载下不超过5秒。
- 在高并发访问情况下，系统运行稳定，无崩溃或严重性能下降现象。
- 系统资源利用率（CPU、内存等）在合理范围内，无资源瓶颈。

失败准则:

- 系统在预期负载下的响应时间超过2秒，高峰负载下超过5秒。
- 在高并发访问情况下，系统出现崩溃或严重性能下降。
- 系统资源利用率超出合理范围，存在资源瓶颈。

8. 兼容性

通过准则:

- 平台在主流浏览器（如Chrome、Firefox、Safari、Edge）上的表现一致，无样式或功能差异。
- 平台在不同设备（如桌面电脑、笔记本电脑、平板电脑）上的表现一致，功能正常。

失败准则:

- 平台在不同浏览器上的表现不一致，存在样式或功能差异。
- 平台在不同设备上的表现不一致，功能无法正常使用。

4 测试用例说明

4.1 目的

本部分定义由测试设计说明所标识的测试用例。通过详细设计和执行这些测试用例，确保“iMusic”在线音乐分享平台的各项功能和特征达到预期的质量标准。测试用例旨在全面验证平台的功能性、性能、兼容性、安全性和用户体验，确保系统在各种场景下的稳定性和可靠性。

1. 功能性测试用例

目的: 验证平台的各项功能是否按需求文档的描述正确实现。确保用户在实际使用过程中，各功能模块均能正常工作，无功能缺失或错误。

- 用户注册与登录
- 音乐资源管理
- 个人主页和信息管理
- 歌曲播放
- 审核与管理

2. 性能测试用例

目的: 评估平台在不同负载情况下的性能表现。确保系统在高并发访问时的响应速度和稳定性，确认系统资源利用率在合理范围内，无性能瓶颈。

- 负载测试
- 压力测试
- 高峰负载测试

3. 兼容性测试用例

目的: 验证平台在不同浏览器和设备上的兼容性。确保平台在各种环境下的表现一致，无功能或样式差异，用户体验良好。

- 多浏览器兼容性测试
- 多设备兼容性测试

4. 安全性测试用例

目的: 发现并修复平台的安全漏洞，确保用户数据和隐私的安全。验证平台在数据传输和存储过程中的安全性，防止信息泄露和未经授权的访问。

- SQL注入测试
- XSS（跨站脚本攻击）测试
- CSRF（跨站请求伪造）测试
- 数据加密和传输安全测试

5. 用户体验测试用例

目的: 评估平台的用户界面友好性和操作便捷性。通过实际用户的测试和反馈，改进用户体验，确保平台易用且满足用户需求。

- 用户界面交互测试
- 用户操作流程测试
- 用户反馈收集和分析

6. 回归测试用例

目的: 验证在修复缺陷后，平台功能的正确性和稳定性。确保修复的缺陷不影响其他功能，且未引入新缺陷。

- 缺陷修复验证测试
- 关键功能回归测试

4.2详细说明

4.2.1 测试用例说明标识符

为了确保测试用例说明在项目管理过程中的唯一性和可追溯性，现为“iMusic”在线音乐分享平台的测试用例说明分配以下唯一标识符：

IMUSIC-TEST-CASES-2024-V1.0

- **IMUSIC:** 表示该测试用例说明所属的项目名称为“iMusic”在线音乐分享平台。
- **TEST-CASES:** 表示该文档类型为测试用例说明。
- **2024:** 表示测试用例说明制定的年份。
- **V1.0:** 表示测试用例说明的版本号，当前为第一个正式版本。

4.2.2 输入说明

本部分规定执行测试用例所需要的每种输入。输入可以用具体的值（必要时允许适当的容差）或名称来规定，包括数据库、文件、终端消息、内存驻留区域及操作系统传送的各个值。

1. 用户注册与登录

输入:

- **邮箱地址:** 用户注册时输入的邮箱地址，例如 `user@example.com`。
- **密码:** 用户设置的密码，例如 `Password123!`。

2. 音乐资源管理

输入:

- **音频文件:** 用户上传的音频文件，例如 `song.mp3`。
- **歌曲元数据:** 用户输入的歌曲信息，包括标题、艺术家等，例如 `{title: "可惜没如果", singer: "林俊杰"}`。
- **歌词文件:** 用户上传或定制的歌词文件，例如 `lyrics.lrc`。

3. 个人主页和信息管理

输入:

- **个人信息:** 用户输入的个人信息，包括用户名、头像、简介等，例如 `{username: "User123", avatar: "avatar.jpg", bio: "This is a bio"}`。
- **收藏夹:** 用户添加的收藏夹名称和歌曲ID，例如 `{songlist_title: "学习", song_id: [1, 2, 3]}`。

4. 歌曲播放

输入:

- **歌曲ID:** 用户选择播放的歌曲ID，例如 `song_id: 123`。
- **评论内容:** 用户在歌曲播放页面编辑的评论，例如 `comment: "好听!"`。
- **播放控制指令:** 用户发送的播放控制指令，例如 `{action: "play"}, {action: "pause"}, {action: "skip", time: 30}`。

5. 审核与管理

输入:

- **投诉内容:** 用户提交的投诉信息，包括投诉类型和详细描述，例如 `{complaint_type: "内容不合适", content: "歌词包含不合适内容"}`。
- **申诉内容:** 被投诉者提交的申诉信息、申诉理由，例如 `{appeal_reason: "歌词内容为艺术化表达，我认为不算违规"}`。

6. 安全性

输入:

- **测试脚本:** 安全测试使用的脚本和攻击载荷, 例如 `SQLi_payload: "' OR '1'='1';`, `XSS_payload: "<script>alert('XSS')</script>"`。
- **用户数据:** 系统中的敏感用户数据, 用于验证加密和保护措施, 例如 `{email: "user@example.com", password: "hashedPassword123"}`。

7. 性能和稳定性

输入:

- **并发用户数:** 性能测试中模拟的并发用户数量, 例如 `concurrent_users: 1000`。
- **请求频率:** 每个用户发送请求的频率, 例如 `requests_per_minute: 60`。

8. 兼容性

输入:

- **浏览器类型和版本:** 兼容性测试使用的不同浏览器及其版本, 例如 `{browser: "Chrome", version: "89.0"}`, `{browser: "Firefox", version: "78.0"}`。
- **设备类型:** 兼容性测试使用的不同设备类型, 例如 `{device: "Desktop", os: "Windows 11"}`, `{device: "Pad", os: "Android 11"}`。

数据库、文件和系统消息

- **用户数据库:** 包含用户信息的数据库表, 例如 `users` 表, 字段包括 `user_id`, `email`, `password`, `username` 等。
- **歌曲数据库:** 包含歌曲信息的数据库表, 例如 `songs` 表, 字段包括 `id`, `title`, `singer`, `introduction` 等。
- **投诉数据库:** 包含投诉信息的数据库表, 例如 `complaint` 表, 字段包括 `id`, `complained_id`, `complainer_id`, `content`, `type`。
- **日志文件:** 记录系统操作和事件的日志文件, 例如 `system.log`, `access.log`。
- **终端消息:** 用户界面和系统之间的消息交互, 例如 `欢迎消息`, `错误提示`, `成功通知`。
- **内存驻留区域:** 用于缓存和临时存储的数据区域, 例如 `Redis` 缓存。

4.2.3 输出说明

本部分规定测试项所有要求的输出和特征, 包括响应时间等。为每个要求的输出或特征提供准确的值, 必要时允许适当的容差。

1. 用户注册与登录

输出:

- **登录成功消息:** 用户成功登录后的提示信息, 例如:

```
{
  "success": true,
```

```
"message": "登录成功",
"data": {
  "user_id": 16,
  "username": "sivenlu",
  "email": "test@buaa.edu.cn",
  "bio": "test",
  "avatar": "http://182.92.100.66:5000/media/avatars/IMG_20240114_1825591-01.jpeg",
  "follower_count": 2,
  "following_count": 1,
  "role": "admin",
  "registration_date": "2024-04-22 19:04:47"
},
"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoxiwiidXNlcm5hbWUiOiJzaxZlbnx1Iiwicm9sZSI6ImFkbWl1IiwiaXNze3NDA4MDE5fQ.RJJnGmB_vMQOC_2YbIgCen1T8QgBGZ1Ln5a852741Tw"
}
```

- **响应时间:** 所有注册、激活、登录和密码重置操作的响应时间不超过2秒。

2. 音乐资源管理

输出:

- **上传成功消息:** 用户成功上传歌曲后的提示信息, 例如 `{"success": true, "message": "歌曲上传成功"}`
- **歌曲展示列表:** 平台展示的所有公开歌曲列表, 包括歌曲的标题、歌手等信息, 例如:

```
{
  "success": true,
  "message": "获取所有歌曲信息成功",
  "data": [
    {
      "id": 8,
      "title": "betty",
      "singer": "Taylor Swift",
      "cover": "http://182.92.100.66:5000/media/covers/betty_cover.jpg",
      "gradient": "background: rgb(160, 160, 160);background: linear-gradient(135deg, rgb(174, 174, 174), rgb(68, 68, 68));",
      "introduction": "《贝蒂》（英语：\"Betty\"，风格化为全小写）是美国创作歌手泰勒·斯威夫特的一首歌曲，于2020年8月17日由联众唱片发行，作为她第八张录音室专辑《民间故事》的第三支单曲。这首歌由斯威夫特及乔·阿尔文（笔名威廉·鲍里）共同创作，由斯威夫特、杰克·安东诺夫及亚伦·戴斯纳共同制作[1]。《贝蒂》是一首乡村及民谣摇滚的歌曲，并有着由吉他及口琴交织而成的乐器。歌词方面，它叙述了詹姆斯的道歉，詹姆斯是《民间故事》中的三角恋核心人物之一。这首歌于发行首周在公告牌的《公告牌》百强单曲榜及热门乡村歌曲榜分别获得第42名及第6名，也成为斯威夫特在热门乡村歌曲榜的第22首前10名歌曲。",
      "audio": "http://182.92.100.66:5000/media/audios/betty_audio.mp3",
      "duration": "294.556734",
      "lyric": "http://182.92.100.66:5000/media/lyrics/betty_lyric.lrc",
      "tag_theme": null,
      "tag_scene": "旅行",
    }
  ]
}
```

```

        "tag_mood": "宣泄",
        "tag_style": "流行",
        "tag_language": "英语",
        "uploader": "sivenlu",
        "like": 2,
        "upload_date": "2024-04-22 19:19:08",
        "user_like": false
    },
    {
        "id": 9,
        "title": "exile",
        "singer": "Taylor Swift",
        "cover": "http://182.92.100.66:5000/media/covers/exile_cover.jpg",
        "gradient": "background: rgb(160, 160, 160);background: linear-
gradient(135deg, rgb(174, 174, 174), rgb(68, 68, 68));",
        "introduction": "《Exile》描绘了两个疏远恋人进行无声对话的想象故事，是一首极简但具
有电影感的独立民谣，带有福音、管弦乐和室内流行元素，将斯威夫特甜蜜的歌声和弗农低沉的男中音结合成一首
忧郁的二重唱，沉重的钢琴、旋转的弦乐和高亢的和声。 《Exile》一经发行就获得了广泛的好评，重点是二人组
合的声乐化学反应、令人痛苦的抒情、丰富的器乐和阴郁的气氛。音乐评论家将其视为《Folklore》中的佼佼者，
并将其称为斯威夫特迄今为止最好的合作作品之一，并将其与她 2011 年的歌曲“ Safe & Sound ”进行比较。
多家出版物将《Exile》列为 2020 年最佳歌曲之一。",
        "audio": "http://182.92.100.66:5000/media/audios/exile_audio.mp3",
        "duration": "285.634172",
        "lyric": "http://182.92.100.66:5000/media/lyrics/exile.lrc",
        "tag_theme": null,
        "tag_scene": "旅行",
        "tag_mood": "宣泄",
        "tag_style": "流行",
        "tag_language": "英语",
        "uploader": "sivenlu",
        "like": 2,
        "upload_date": "2024-04-22 19:21:37",
        "user_like": false
    },
    ...
]
}

```

- **响应时间:** 上传、展示和导出操作的响应时间不超过3秒。

3. 个人主页和信息管理

输出:

- **个人信息更新成功消息:** 用户成功更新个人信息后的提示信息，例如 `{"success": true,"message": "用户信息修改成功"}`
- **收藏夹和歌单:** 用户成功创建的收藏夹和歌单信息，例如 `{'success': True, 'message': '歌单创建成功', 'id': 123}`
- **关注成功消息:** 用户成功关注其他用户后的提示信息，例如 `{"success": true,"message": "取消关注成功"}`

- **响应时间:** 个人信息更新、收藏夹管理和关注操作的响应时间不超过2秒。

4. 歌曲播放

输出:

- **歌曲信息:** 用户查看的歌曲详细信息，包括标题、艺术家、专辑、时长等，例如：

```
{
  "success": true,
  "message": "获取歌曲信息成功",
  "data": {
    "id": 9,
    "title": "exile",
    "singer": "Taylor Swift",
    "cover": "http://182.92.100.66:5000/media/covers/exile_cover.jpg",
    "gradient": "background: rgb(160, 160, 160);background: linear-gradient(135deg, rgb(174, 174, 174), rgb(68, 68, 68));",
    "introduction": "《Exile》描绘了两个疏远恋人进行无声对话的想象故事，是一首极简但具有电影感的独立民谣，带有福音、管弦乐和室内流行元素，将斯威夫特甜蜜的歌声和弗农低沉的男中音结合成一首忧郁的二重唱，沉重的钢琴、旋转的弦乐和高亢的和声。《Exile》一经发行就获得了广泛的好评，重点是二人组合的声乐化学反应、令人痛苦的抒情、丰富的器乐和阴郁的气氛。音乐评论家将其视为《Folklore》中的佼佼者，并将其称为斯威夫特迄今为止最好的合作作品之一，并将其与她 2011 年的歌曲“ Safe & Sound ”进行比较。多家出版物将《Exile》列为 2020 年最佳歌曲之一。",
    "audio": "http://182.92.100.66:5000/media/audios/exile_audio.mp3",
    "duration": "285.634172",
    "lyric": "http://182.92.100.66:5000/media/lyrics/exile.lrc",
    "tag_theme": null,
    "tag_scene": "旅行",
    "tag_mood": "宣泄",
    "tag_style": "流行",
    "tag_language": "英语",
    "uploader": "sivenlu",
    "like": 2,
    "upload_date": "2024-04-22 19:21:37",
    "user_like": false
  }
}
```

- **播放状态:** 用户进行播放控制后的状态信息，例如 `{"status": "playing", "current_time": "00:30"}`，`{"status": "paused"}`
- **评论成功消息:** 用户成功编辑评论后的提示信息，例如 `{"success": true,"message": "评论成功"}`
- **历史播放记录:** 用户查看的历史播放记录列表，例如：

```
{
  "success": true,
  "message": "查询最近播放歌曲成功",
  "data": [
    {
```

```
    "song": {
      "id": 6,
      "title": "海阔天空",
      "singer": "beyond",
      ...
    },
    "last_play": "2024-04-26 10:16:45",
    "play_count": 3
  }
]
```

- **响应时间:** 歌曲播放、评论和历史记录查看操作的响应时间不超过2秒。

5. 审核与管理

输出:

- **投诉成功消息:** 用户成功提交投诉后的提示信息, 例如 `{"success": true, "message": "投诉成功"}`
- **投诉处理状态:** 管理端处理投诉后的状态信息, 例如 `{"complaint_id": 654, "status": "待处理"}`
- **下架成功消息:** 管理端成功下架歌曲或取消分享歌单后的提示信息, 例如 `{"message": "下架成功.", "song_id": 456}`
- **响应时间:** 投诉、申诉和管理操作的响应时间不超过3秒。

6. 安全性

输出:

- **安全扫描报告:** 安全测试后的扫描报告, 包括所有发现的漏洞及其详细描述, 例如 `{"vulnerabilities": [{"type": "SQL Injection", "description": "SQL injection found in login form."}]}`
- **数据加密验证:** 验证数据传输和存储过程中的加密状态, 例如 `{"data_encrypted": true}`
- **响应时间:** 安全扫描和验证操作的响应时间不超过5秒。

7. 性能和稳定性

输出:

- **响应时间报告:** 性能测试后的响应时间报告, 包括平均响应时间和高峰响应时间, 例如 `{"average_response_time": "1.5s", "peak_response_time": "4.0s"}`
- **系统资源利用率:** 性能测试后的系统资源利用率报告, 包括CPU、内存使用情况, 例如 `{"cpu_usage": "70%", "memory_usage": "60%"}`
- **稳定性报告:** 高并发访问测试后的稳定性报告, 包括系统是否出现崩溃或严重性能下降现象, 例如 `{"stable": true}`

8. 兼容性

输出:

- **兼容性报告:** 兼容性测试后的报告, 包括在不同浏览器和设备上的测试结果, 例如

```
{"browser_compatibility": {"Chrome": "pass", "Firefox": "pass", "Safari": "pass", "Edge": "pass"}, "device_compatibility": {"Desktop": "pass", "Pad": "pass"}}
```

- **响应时间:** 兼容性测试操作的响应时间不超过2秒。

通过上述输出说明, 确保每个测试项的预期输出和特征均已明确, 保证测试结果的准确性和可验证性。

5 测试规程说明

5.1 目的

5.2 详细说明

5.2.1 测试规程说明标识符

5.2.2 特殊要求

5.2.3 规程步骤

6 测试日志

6.1 后端测试

包括单元测试和集成测试, 使用 Django 内置测试模块。

一共有 234 个测试用例, 基本涵盖可能出现的各种情况, 详细测试代码见各个 Application 的 test.py 文件。

经过测试, 所有样例均通过。下面分别展示测试结果。

测试 user application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test user
Found 33 test(s).
System check identified no issues (0 silenced).
None
..valid_code
.....123456
..123456
..123456
123456
123456
.
-----
Ran 33 tests in 10.460s

OK
```

测试 song application:


```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test song
Found 47 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 47 tests in 29.840s

OK
```

测试 songlist application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test songlist
Found 34 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 34 tests in 19.159s

OK
```

测试 comment application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test comment
Found 8 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 8 tests in 3.350s

OK
```

测试 follow application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test follow
Found 22 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 22 tests in 13.956s

OK
```

测试 like application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test like
Found 6 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 6 tests in 2.379s

OK
```

测试 message application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test message
Found 23 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 23 tests in 22.038s

OK
```

测试 recommend application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test recommend
Found 3 test(s).
System check identified no issues (0 silenced).
..test Joyful
test Joyful
test Joyful
test Joyful
test Joyful
test Joyful
'
-----
Ran 3 tests in 2.267s

OK
```

测试 search application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test search
Found 7 test(s).
System check identified no issues (0 silenced).
..['Happy']
..['sad']
....
-----
Ran 7 tests in 3.384s

OK
```

测试 `timedtask` application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test timedtask
Found 2 test(s).
System check identified no issues (0 silenced).
生成2024-05-24至2024-05-31的周报
周报生成完毕
..
-----
Ran 2 tests in 0.885s

OK
```

测试 `feature` application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test feature
Found 2 test(s).
System check identified no issues (0 silenced).
..
-----
Ran 2 tests in 1.074s

OK
```

测试 `ishare` application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test ishare
Found 18 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 18 tests in 12.007s

OK
```

测试 `complaint` application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test complaint
Found 23 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 23 tests in 19.807s

OK
```

测试 `singer` application:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test singer
Found 6 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 6 tests in 3.370s

OK
```

总测试:

```
PS D:\HuaweiMoveData\Users\29824\Desktop\SE\Project\Imusic backend > python manage.py test
Found 234 test(s).
System check identified no issues (0 silenced).
.....test Joyful
test Joyful
test Joyful
test Joyful
...['Happy']
..['sad']
.....生成2024-05-24至2024-05-31的
周报
周报生成完毕
..None
..valid_code
.....123456
..123456
..123456
123456
123456
.
-----
Ran 234 tests in 147.160s

OK
```

6.2 前端测试

对于较为复杂的逻辑，使用unittest创建和运行测试，使用Selenium模拟用户在网页上的各种操作。测试文件见分支 `frontend-test` 中 `test` 文件夹下 `test*.py`。

经过测试，所有样例均通过。下面分别展示测试结果。

主体功能测试

```
✓ 测试已通过: 3 共 3 个测试 - 23秒 689毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:17 ...
Launching unittests with arguments python -m unittest testHome.TestHome in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test

Ran 3 tests in 23.692s

OK

进程已结束，退出代码为 0
```

用户登录注册测试

```
✓ 测试已通过: 2 共 2 个测试 - 21秒 31毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:18 ...
Launching unittests with arguments python -m unittest testLogin.TestLogin in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test

Ran 2 tests in 21.034s

OK

进程已结束，退出代码为 0
```

管理员测试

```
✓ 测试已通过: 2 共 2 个测试 - 35秒 216毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:24 ...
Launching unittests with arguments python -m unittest testAdmin.TestAdmin in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test

Ran 2 tests in 35.218s

OK

进程已结束，退出代码为 0
```

消息测试

```
✓ 测试 已通过: 1共 1 个测试 - 24秒 456毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:32 ...
Launching unittests with arguments python -m unittest testMessage.TestLogin.test_complaint_message in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusi

Ran 1 test in 24.457s

OK
```

```
✓ 测试 已通过: 1共 1 个测试 - 17秒 120毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:32 ...
Launching unittests with arguments python -m unittest testMessage.TestLogin.test_system_message in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-r

Ran 1 test in 17.121s

OK
```

```
✓ 测试 已通过: 1共 1 个测试 - 19秒 402毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:33 ...
Launching unittests with arguments python -m unittest testMessage.TestLogin.test_comment_message in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-

Ran 1 test in 19.403s

OK
```

```
✓ 测试 已通过: 1共 1 个测试 - 27秒 33毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:34 ...
Launching unittests with arguments python -m unittest testMessage.TestLogin.test_private_message in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-

Ran 1 test in 27.035s

OK
```

```
✓ 测试 已通过: 1共 1 个测试 - 25秒 999毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:35 ...
Launching unittests with arguments python -m unittest testMessage.TestLogin.test_private_message_send in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\im

Ran 1 test in 26.000s

OK
```

```
✓ 测试 已通过: 1共 1 个测试 - 47秒 706毫秒

C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-router\test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm 2023.3.3/plugins/python/hel
Testing started at 11:36 ...
Launching unittests with arguments python -m unittest testMessage.TestLogin.test_appeal_message in C:\Users\28109\Desktop\i4\Imusic\imusic_frontend\imusic-r

Ran 1 test in 47.707s

OK
```

均通过测试

对于简单逻辑，使用手动测试方法测试页面是否正确展示。

注销模块

测试对象	注销功能
前置条件	已经登录
后置条件	退出至未登录状态
交互行为	用户点击注销按钮
测试步骤	点击注销按钮，用户状态变为未登录
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.29

用户信息修改模块

测试对象	用户信息修改模块
前置条件	已登录用户
后置条件	用户信息修改成功
交互行为	用户修改个人信息
测试步骤	登录后点击个人信息编辑按钮，修改信息，保存修改
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.29

用户关注和取消关注模块

测试对象	用户关注和取消关注模块
前置条件	已登录用户
后置条件	成功关注或取消关注
交互行为	用户关注或取消关注其他用户
测试步骤	登录后点击关注按钮关注未关注用户，点击取消关注按钮取消对已关注用户的关注
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.29

播放歌曲模块

测试对象	播放歌曲模块
前置条件	无
后置条件	歌曲成功播放，展示详细信息
交互行为	用户点击目标歌曲的播放按钮
测试步骤	点击播放按钮，观察歌曲是否开始播放，并查看歌曲信息的展示
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.30

上传歌曲模块

测试对象	上传歌曲模块
前置条件	用户已登录
后置条件	歌曲成功上传至平台，在用户已创建的歌曲界面可以看到详细信息
交互行为	用户填写歌曲详细信息，包括歌曲名、歌手、歌曲文件、歌曲封面、标签、介绍、歌词，点击上传歌曲按钮
测试步骤	登录后填写歌曲信息，点击上传按钮
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.30

修改歌曲模块

测试对象	修改歌曲模块
前置条件	用户已登录，修改的歌曲为用户曾经上传的歌曲
后置条件	歌曲信息成功修改
交互行为	用户修改自己上传的歌曲信息

测试对象	修改歌曲模块
测试步骤	登录后查找已上传的歌曲，点击修改按钮，编辑歌曲信息并保存
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.30

删除歌曲模块

测试对象	删除歌曲模块
前置条件	用户已登录，删除的歌曲为用户曾经上传的歌曲
后置条件	歌曲成功从平台删除
交互行为	用户删除自己上传的歌曲
测试步骤	登录后查找已上传的歌曲，点击删除按钮，确认删除操作
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.30

创建歌单模块

测试对象	创建歌单模块
前置条件	用户已登录
后置条件	歌单成功创建
交互行为	用户创建新的歌单
测试步骤	登录后点击创建歌单按钮，填写歌单信息并保存
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.30

查看歌单模块

测试对象	查看歌单模块
前置条件	用户已登录
后置条件	成功查看歌单信息

测试对象	查看歌单模块
交互行为	用户查看自己创建的歌单信息
测试步骤	登录后点击查看歌单按钮，查看歌单详细信息
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.30

修改歌单模块

测试对象	修改歌单模块
前置条件	用户已登录，且修改的歌单为用户曾经创建的歌单
后置条件	歌单信息成功修改
交互行为	用户修改自己创建的歌单信息
测试步骤	登录后查找已创建的歌单，点击修改按钮，编辑歌单信息并保存
测试结果	通过
测试人	牛梓涵
测试时间	2024.5.30

删除歌单模块

测试对象	删除歌单模块
前置条件	用户已登录，且删除的歌单为用户曾经创建的歌单
后置条件	歌单成功从平台删除
交互行为	用户删除自己创建的歌单
测试步骤	登录后查找已创建的歌单，点击删除按钮，确认删除操作
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

添加歌曲至歌单模块

测试对象	添加歌曲至歌单模块
前置条件	用户已登录，且歌单为用户创建的歌单
后置条件	歌曲成功添加至歌单
交互行为	用户将歌曲添加至歌单
测试步骤	登录后查找歌曲，点击添加至歌单按钮，选择歌单并确认添加操作
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

收藏歌单模块

测试对象	收藏歌单模块
前置条件	用户已登录，且歌单为用户未收藏的歌单
后置条件	歌单成功添加至收藏
交互行为	用户点击收藏感兴趣的歌单
测试步骤	登录后进入歌单页面，点击收藏按钮，确认收藏操作
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

取消收藏歌单模块

测试对象	取消收藏歌单模块
前置条件	用户已登录，且歌单为用户收藏的歌单
后置条件	歌单从收藏中移除
交互行为	用户点击取消收藏的歌单
测试步骤	登录后进入收藏的歌单页面，点击取消收藏按钮，确认取消收藏操作
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

添加喜欢的歌曲模块

测试对象	添加喜欢的歌曲模块
前置条件	用户已登录，且歌曲为用户未喜欢的歌曲
后置条件	歌曲成功添加至喜欢
交互行为	用户点击喜欢按钮，将歌曲添加至喜欢
测试步骤	登录后播放歌曲页面，点击喜欢按钮，确认添加至喜欢操作
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

取消喜欢的歌曲模块

测试对象	取消喜欢的歌曲模块
前置条件	用户已登录，且歌曲为用户喜欢的歌曲
后置条件	歌曲从喜欢中移除
交互行为	用户点击取消喜欢按钮，取消喜欢的歌曲
测试步骤	登录后进入喜欢的歌曲页面，点击取消喜欢按钮，确认取消喜欢操作
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

关注模块

测试对象	关注模块
前置条件	用户已登录，且未关注另一用户
后置条件	成功关注用户，被关注者收到相关通知
交互行为	用户选择关注其他用户
测试步骤	登录后查看其他用户，点击关注按钮，确认关注操作
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

更新信息模块

测试对象	更新信息模块
前置条件	被关注者更新了信息
后置条件	关注者收到相关更新通知
交互行为	被关注者更新了歌曲或歌单
测试步骤	被关注者更新了歌曲或歌单信息，关注者收到更新通知
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

上传信息模块

测试对象	上传信息模块
前置条件	被关注者上传了歌曲/创建了歌单
后置条件	关注者收到相关上传通知
交互行为	被关注者上传了歌曲/创建了歌单
测试步骤	被关注者上传了歌曲/创建了歌单，关注者收到上传通知
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

听歌周报和创作周报模块

测试对象	听歌周报和创作周报模块
前置条件	用户已登录
后置条件	用户成功收到听歌周报和创作周报
交互行为	系统定时发送周报
测试步骤	用户在系统消息中查看听歌周报和创作周报
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

评论消息模块

测试对象	评论消息模块
前置条件	用户已登录，上传的歌曲受到其他用户评论
后置条件	上传者成功收到评论通知
交互行为	用户在歌曲页面发表评论
测试步骤	登录后在歌曲页面发表评论，上传者收到评论通知
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

投诉歌曲或歌单模块

测试对象	投诉歌曲或歌单模块
前置条件	用户已登录
后置条件	投诉成功并通知相关用户
交互行为	用户发起投诉请求
测试步骤	用户在歌曲或歌单详情页面发起投诉请求，被投诉者和管理员收到投诉通知
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

申诉模块

测试对象	申诉模块
前置条件	用户已登录，用户上传的歌曲或歌单被管理员下架
后置条件	申诉成功并通知管理员
交互行为	被投诉用户发起申诉
测试步骤	被投诉用户在系统中发起申诉请求，管理员收到申诉通知
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

查看和处理投诉信息模块

测试对象	查看和处理投诉信息模块
前置条件	管理员已登录
后置条件	成功查看和处理投诉信息
交互行为	管理员查看和处理投诉信息
测试步骤	管理员登录后进入投诉信息界面，查看和处理投诉请求
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

查看和处理申诉信息模块

测试对象	查看和处理申诉信息模块
前置条件	管理员已登录
后置条件	成功查看和处理申诉信息
交互行为	管理员查看和处理申诉信息
测试步骤	管理员登录后进入申诉信息界面，查看和处理申诉请求
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

添加评论模块

测试对象	添加评论模块
前置条件	用户已登录
后置条件	成功发表评论并通知相关用户
交互行为	用户在歌曲播放页面发表评论
测试步骤	登录后在歌曲播放页面发表评论，上传者收到评论通知
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

查看评论模块

测试对象	查看评论模块
前置条件	用户已登录
后置条件	成功查看歌曲播放页面的评论
交互行为	用户在歌曲播放页面查看评论
测试步骤	登录后进入歌曲播放页面，查看评论列表
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

删除评论模块

测试对象	删除评论模块
前置条件	用户已登录，且此评论为用户发表的评论
后置条件	成功删除自己发表的评论
交互行为	用户在歌曲播放页面删除评论
测试步骤	登录后在歌曲播放页面查找自己发表的评论，点击删除按钮
测试结果	通过
测试人	牛梓涵
测试时间	2024.6.1

7 测试总结报告

在本次测试活动中，我们对项目进行了全面的测试，包括单元测试、集成测试、系统测试和验收测试。我们使用了自动化测试工具以及手动测试方法，以确保覆盖所有的测试用例。

7.1 测试结果

所有的测试用例都已经执行。我们将发现的所有问题都及时地报告并修复。修复后通过了所有的测试用例。

7.2 性能测试

性能测试的结果显示，系统在高负载下仍然能够保持良好的性能。响应时间、吞吐量和资源利用率都达到了需求中预期的标准。

7.3 安全性测试

通过安全性测试，我们确认了系统的安全性措施，如数据加密和用户认证，都能够有效地防止潜在的攻击。

7.4 兼容性测试

兼容性测试表明，我们的应用在主要的浏览器和操作系统上都能够正常工作。

7.5 测试评价

我们的测试覆盖了所有的功能和需求，确保了产品的质量。虽然我们发现了一些问题，但是这些问题都已经被及时地解决。

在未来，我们将继续进行测试活动，以确保产品的质量和满足用户的需求。我们也将向及时处理用户的建议以改进我们的测试过程和方法。