

目录

1	引言	4
1.1	开发目的	4
1.2	现状及意义	4
1.3	背景	4
2	系统可行性分析	4
2.1	可行性研究的前提	4
2.1.1	要求	4
2.1.2	目标	6
2.1.3	条件、假定和限制	6
2.1.4	进行可行性研究的方法	7
2.1.5	评价尺度	8
2.2	对现有系统的分析	8
2.2.1	处理流程和数据流程	8
2.2.2	工作负荷	10
2.2.3	费用开支	11
2.2.4	人员	11
2.2.5	设备	12
2.2.6	局限性	12
2.3	所建议的系统	12
2.3.1	对所建议系统的说明	12
2.3.2	处理流程和数据流程	12
2.3.3	改进之处	13
2.3.4	影响	13
2.3.5	局限性	13
2.3.6	技术条件方面的可行性	13
2.4	可选择的其他系统方案	14
2.4.1	可选择的系统方案 1	14
2.4.2	可选择的系统方案 2	14
2.5	投资及效益分析	14
2.5.1	支出	14
2.5.2	收益	15
2.5.3	收益/投资比	15
2.5.4	投资回收周期	15
2.5.5	敏感性分析	16
2.6	社会因素方面的可行性	16

2.6.1	法律方面的可行性	16
2.6.2	使用方面的可行性	16
2.7	结论	16
3	系统开发计划	17
3.1	引言	17
3.2	项目概述	17
3.2.1	工作内容	17
3.2.2	主要参加人员	17
3.2.3	产品	17
3.2.4	完成项目的最迟期限	18
3.2.5	本计划的批准者和批准日期	18
3.3	实施计划	19
3.3.1	工作任务的分解与人员分工	19
3.3.2	接口人员	19
3.3.3	进度	19
3.3.4	预算	19
3.3.5	关键问题	20
3.4	支持条件	20
3.4.1	计算机系统支持	20
3.4.2	需由用户承担的工作	20
3.4.3	由外单位提供的条件	21
3.4.4	计算机系统支持	21
3.5	专题计划要点	21
4	系统（软件）需求规格说明	21
4.1	引言	21
4.2	任务概述	22
4.2.1	目标	22
4.3	软件的一般性描述	23
4.3.1	软件产品与其环境之间的关系	23
4.3.2	限制与约束	24
4.3.3	假设与前提条件	24
4.4	软件功能需求描述	25
4.4.1	软件功能概述	25
4.4.2	软件需求的用例模型	26
4.4.3	软件需求的分析模型	27
4.5	其它软件需求描述	30

4.5.1	性能要求	30
4.5.2	设计约束	30
4.5.3	界面要求	31
4.5.4	进度要求	31
4.5.5	交付要求	32
4.5.6	验收要求	32
4.6	软件原型	33
5	软件设计规格说明	35
5.1	引言	35
5.1.1	编写目的	35
5.1.2	读者对象	35
5.1.3	软件项目概述	35
5.1.4	文档概述	36
5.1.5	定义	36
5.2	软件设计约束	37
5.2.1	软件设计目标和原则	37
5.2.2	软件设计的约束和限制	38
5.3	软件设计	38
5.3.1	软件体系结构设计	38
5.3.2	用户界面设计	39
5.3.3	用例设计	40
5.3.4	类设计	44
5.3.5	数据设计	46
5.3.6	部署设计	52
5.3.7	系统出错处理设计	52
6	系统测试	55
6.1		55
6.2		55
6.3		55
6.4		55
6.5		55
6.6		55

1 引言

1.1 开发目的

本软件（以下均简称 MS）旨在为 Android 用户提供一个集 AI 音乐创作，作品分享，音乐鉴赏，社区互动为一体的音乐社区平台。

1.2 现状及意义

随着信息时代的逐渐发展，AI 创作以及普及到了世界各地。早在 2023 年，Google 已经发布了 MusicLM，它能够根据用户输入的文字描述（如风格、情绪、乐器、场景等），直接生成高质量、结构完整的音乐片段（最长可达数分钟），标志着 AI 音乐生成技术的重要突破。然而当前国内市场上未存在“AI 创作 + 鉴赏交流”的社交平台。本软件可以推动多模态 AI 音乐技术落地，融合生成、编辑、分析能力，实现从”工具”到”创作伙伴”的升级。打破音乐创作阶层壁垒，促进多元文化表达，为独立音乐人提供新舞台。通过社区乐评、风格解析等功能，提升大众音乐素养，传承音乐文化。

1.3 背景

为了为众多音乐爱好者提供良好便捷的交流以及自主编写音乐的平台，我们小组决定开发一个音乐生成及鉴赏社区。该项目由我们小组组长提出，由小组成员共同完成。面向群众为音乐爱好者，独立音乐人，音乐教育者等人群。本项目使用华为昇腾 MindSpore 框架，私有化部署 AI 模型进行增量化模型训练。

2 系统可行性分析

2.1 可行性研究的前提

说明对所建议的开发项目进行可行性研究的前提，如要求、目标、假定、限制等。

2.1.1 要求

A. 功能

音乐生成功能 基于 MindSpore 框架，接收文本、图像等多模态输入，实现流行、古典、电子等多风格音乐的自动生成，支持用户对生成音乐的节奏、旋律、和声等参数进行个性化调整。

音乐鉴赏功能 通过人工智能技术分析音乐的旋律结构、和声走向、节奏特点等元素，结合用户评价与社区互动数据，为用户提供音乐评分、优缺点分析、相似音乐推荐等鉴赏服务。

社区交互功能 构建用户注册登录、作品上传分享、评论点赞、私信交流、关注粉丝等社交功能，打造音乐创作者与爱好者交流互动的平台，支持用户创建或加入音乐主题小组。

B. 性能

响应时间 音乐生成请求在普通网络环境下，响应时间不超过 12 秒；社区交互操作如评论、点赞等，响应时间控制在 1.5 秒以内。

吞吐量 支持至少千级用户同时在线访问，每秒可处理 500 次以上的音乐生成与社区交互请求。

稳定性 系统全年可用性不低于 99.9%，平均无故障运行时间 (MTBF) 不低于 5000 小时，具备自动容灾与故障恢复能力。

C. 输出

生成音乐文件 格式为常见的 MP3、WAV 等，用于用户下载保存、分享传播，每次音乐生成操作即产生，用户可通过社区平台下载接口获取，分发对象为音乐生成的用户及该用户分享的其他社区成员。

音乐鉴赏报告 以文字、图表结合的形式，展示音乐的各项分析数据与评价结果，在用户触发鉴赏功能时生成，可在线查看或下载为 PDF 格式，接口位于音乐作品详情页，分发对象为发起鉴赏请求的用户。

社区数据统计报表 按日、周、月统计用户活跃度、作品上传数量、社区互动数据等，用于项目运营分析，产生频度为每周，以 Excel 格式存储，通过后台管理系统接口获取，分发对象为项目运营团队。

D. 输入

数据来源 音乐生成的多模态输入数据来源于用户手动输入；音乐鉴赏的原始音乐数据由用户上传及合作音乐平台授权提供；社区交互数据来自用户在平台的操作行为。

数据类型 包括文本（用户输入的描述、评论等）、图像（作为音乐生成灵感的图片）、音频（用户上传的音乐作品、待鉴赏的音乐）、用户操作日志等数据类型。

数据数量与组织 预计初期每日新增用户输入数据 500 条左右，音乐文件上传量 100 个，随着用户增长逐步上升；数据按用户 ID、功能模块进行分类存储与管理，采用分布式数据库架构。

提供频度 用户输入数据实时产生；音乐文件上传与社区交互数据持续产生；合作平台提供的音乐数据根据合作协议定期更新。

E. 安全与保密方面要求

数据安全 采用加密技术对用户上传的音乐文件、个人隐私数据进行加密存储与传输，防止数据泄露；定期进行数据备份，确保数据的完整性与可恢复性。

用户认证与授权 严格用户注册登录认证机制，采用账号密码、手机验证码、第三方平台授权等多种认证方式；对用户操作权限进行分级管理，不同权限用户只能访问相应功能与数据。

系统安全防护 部署防火墙、入侵检测系统等安全防护设备，实时监测系统安全状

态，防范网络攻击、恶意访问等安全威胁。

2.1.2 目标

A. 人力与设备费用的减少

通过 MindSpore 框架的高效性与自动化特性，降低音乐生成与鉴赏算法开发过程中对人力的依赖。利用框架内置的模型训练与优化功能，减少开发人员手动调参、代码优化的工作量，缩短开发周期，进而降低人力成本。MindSpore 的分布式训练与推理能力，可有效提升服务器资源利用率，避免因资源浪费而额外购置设备，降低硬件设备投入费用，实现人力与设备资源的高效配置。

B. 处理速度的提高

借助 MindSpore 强大的计算能力与优化算法，提升音乐生成与鉴赏系统的处理速度。在音乐生成环节，针对用户输入的多模态数据，系统可快速调用预训练模型进行分析与生成，确保在普通网络环境下，音乐生成请求响应时间不超过 8 秒。对于音乐鉴赏功能，通过对音乐元素的并行分析与智能算法优化，结合社区互动数据的快速检索，能在短时间内生成音乐鉴赏报告，同时保证社区交互操作如评论、点赞等响应时间控制在 1.5 秒以内，为用户提供流畅高效的使用体验。

C. 管理信息服务的改进

MusicStorm 社区通过整合用户行为数据、音乐作品数据、社区互动数据等多维度信息，构建完善的管理信息服务体系。运营团队可通过后台管理系统实时获取用户活跃度、作品上传数量、社区互动趋势等统计报表，以周为周期生成详细的数据分析报告。这些数据支持可视化展示与深度挖掘，帮助运营团队更精准地了解用户需求、评估社区运营效果，从而制定更有效的运营策略，实现管理信息服务从传统人工统计向智能化、精准化的转变。

2.1.3 条件、假定和限制

A. 所建议系统的运行寿命的最小值

假定 MusicStorm 系统的运行寿命最小值为 2 年。在这期间，需持续维护系统的稳定性与功能性，不断进行技术迭代与优化，以适应市场变化与用户需求升级，确保系统在音乐生成、鉴赏及社区交互等核心功能上保持竞争力，同时保障用户数据安全与服务质量。

B. 进行系统方案选择比较的时间

在项目启动后的第 2 天内，完成系统方案的选择与比较工作。此阶段将综合评估不同技术方案在实现音乐生成、鉴赏及社区交互功能上的可行性、效率、成本等因素，结合 MindSpore 框架特性，确定最优开发方案，为后续开发工作奠定基础，避免因方案选择延误项目整体进度。

C. 法律和政策方面的限制

音乐版权法律 在音乐生成、上传、分享及传播过程中，需严格遵守《著作权法》等相关法律法规，确保用户上传作品的版权合法性，以及生成音乐的版权归属明确。对于用户生成内容（UGC），需建立完善的版权审核与授权机制，避免侵权纠纷。

互联网信息服务政策：遵循国家互联网信息服务相关规定，对社区内容进行严格审核与管理，禁止出现违法违规、低俗有害信息。定期进行内容合规性检查，及时响应政策变化，调整运营策略，确保系统合法合规运营。

D. 可利用的信息和资源

数据资源 可利用公开的音乐数据集（如 MIR - 1K、GTZAN 等）进行模型训练与优化，同时与音乐版权方、流媒体平台合作，获取授权音乐数据用于鉴赏功能。此外，社区用户生成的内容与交互数据将成为重要的数据资源，用于改进系统功能与用户体验。

技术资源：依托 MindSpore 社区的技术文档、开源模型及开发者支持，加速项目开发进程。同时，借鉴人工智能、音乐信息检索等领域的前沿研究成果，提升系统技术水平。

2.1.4 进行可行性研究的方法

A. 技术可行性研究方法

文献研究与技术调研 广泛查阅国内外关于 MindSpore 框架应用、音乐生成算法、多模态数据处理及社区平台开发的学术文献、技术报告和行业案例。重点分析 MindSpore 在音乐领域的成功实践，研究当前主流音乐生成模型（如变分自编码器 VAE、生成对抗网络 GAN 在音乐生成中的应用）的优缺点，了解多模态数据融合技术的发展现状，为项目技术方案提供理论支撑与技术参考。

技术测试与实验 搭建小型技术验证平台，针对 MusicStorm 的核心功能进行技术测试。例如，基于 MindSpore 框架实现简单的音乐生成模块，测试其在不同风格音乐生成上的效果；开发初步的多模态输入接口，验证文本、图像等数据转换为音乐元素的可行性与准确性；构建基础的社区交互原型，测试系统在高并发访问下的性能表现。通过实验数据评估技术方案的可行性与潜在风险。

B. 社会可行性研究方法

法律与政策合规性审查 全面梳理与音乐产业、互联网服务相关的法律法规与政策文件，确保 MusicStorm 项目在音乐版权管理、用户数据保护、内容审核等方面符合法律要求。定期关注政策变化，及时调整项目运营策略，避免法律风险，保证项目的合法合规运营。

社会影响评估 分析项目对音乐产业生态、社会文化发展的影响。评估项目是否有助于推动音乐创作创新、促进音乐文化传播、培养音乐人才；研究项目对社会就业、经济增长的贡献，判断项目在社会层面的可行性与价值。

利益相关者分析 识别项目的利益相关者，包括用户、音乐版权方、投资者、合作伙伴等，分析各利益相关者的需求与期望，评估项目对他们的影响。通过建立良好的沟通机制与合作模式，协调各方利益，确保项目得到社会各界的支持与认可。

2.1.5 评价尺度

A. 各项功能的优先次序

核心功能优先 音乐生成与鉴赏功能作为项目的核心，优先保证其稳定性与创新性。在开发资源分配上，将 60%-70% 的人力和时间投入到核心功能的研发与优化中，确保生成音乐的质量达到专业水准，鉴赏结果具有权威性和参考价值。**基础功能完善** 社区交互功能为用户参与和使用项目的基础，在核心功能开发的同时，同步完善用户注册登录、作品上传分享、评论点赞等基础交互功能，投入 20%-30% 的资源，保障社区的基本使用体验。**增值功能拓展** 在核心功能和基础功能稳定运行后，利用剩余 10%-20% 的资源，逐步开发会员专属服务、音乐版权交易等增值功能，提升项目的商业价值和用户粘性。

B. 开发时间的长短

计划时间对比 以项目制定的 12 个月开发周期为基准，在项目开发过程中，定期检查各阶段任务的完成进度。若实际开发与计划时间偏差在 $\pm 10\%$ 以内，视为项目进度可控；若超过 10%，则需分析原因，判断是技术难题、资源不足还是管理问题导致，并及时调整开发计划。**市场机遇考量** 考虑到音乐市场和人工智能技术的快速发展，若项目开发时间过长，可能错失市场机遇。因此，即使项目在技术上可行，但开发时间超过 15 个月，且市场竞争格局已发生重大变化，也需重新评估项目的必要性和可行性。

C. 使用中的难易程度

界面设计友好性 通过用户测试和问卷调查，评估系统界面的布局是否简洁明了，操作流程是否清晰易懂。若 80% 以上的用户认为界面友好，操作方便，则说明系统在界面设计方面表现良好；若低于 60%，则需对界面进行优化改进。**学习成本高低** 观察新用户首次使用系统时，掌握核心功能所需的时间。若用户能在 10 分钟内独立完成音乐生成或鉴赏的基本操作，且在 30 分钟内熟悉社区的主要交互功能，说明系统的学习成本较低，易于使用；反之，则需优化功能设计，降低学习门槛。**用户反馈与投诉率** 统计用户在使用过程中的反馈意见和投诉数量，若每月用户投诉率低于 0.5%，且积极反馈建议较多，说明用户对系统的使用体验较为满意；若投诉率超过 1%，则需深入分析问题根源，及时解决用户痛点，提升系统易用性。

2.2 对现有系统的分析

2.2.1 处理流程和数据流程

A. 处理流程

音乐生成处理流程：

1. 用户在系统界面输入多模态信息，如描述音乐风格、情绪的文本，或上传作为灵感来源的图像等。

2. 输入数据进入预处理模块，对文本进行语义分析、关键词提取，对图像进行特征识别与转换，将多模态数据转化为适合音乐生成模型处理的格式。
3. 数据传输至基于 MindSpore 搭建的音乐生成模型，模型依据预设算法和训练参数，结合输入数据生成音乐的旋律、和声、节奏等元素，形成初步音乐片段。
4. 初步生成的音乐片段进入优化调整模块，用户可在此对音乐的速度、调式、乐器音色等参数进行手动调整，系统根据用户指令对音乐片段进行优化。
5. 优化后的音乐文件经格式转换，以常见的 MP3、WAV 等格式存储至系统服务器，并返回生成成功提示，用户可选择下载或直接在社区分享。

音乐鉴赏处理流程：

1. 用户上传待鉴赏的音乐文件或选择社区内其他用户分享的音乐。
2. 音乐文件进入音频解析模块，提取音乐的旋律、和声、节奏、音色等基础音频特征。
3. 提取的音频特征数据与系统音乐知识库中的标准数据进行比对分析，同时结合社区内其他用户对该音乐的评价数据、点赞收藏数据等进行综合评估。
4. 基于分析评估结果，生成包含音乐评分、优缺点分析、相似音乐推荐等内容的鉴赏报告，展示在音乐作品详情页面供用户查看。

社区交互处理流程：

1. 用户注册登录系统，完成身份验证后进入社区。
2. 用户可进行作品上传、发布动态、评论他人作品、点赞收藏、关注其他用户、加入主题小组等交互操作。
3. 系统实时接收用户交互操作指令，将相关数据存储至用户行为数据库，同时更新社区页面展示内容，如在用户个人主页更新作品列表，在动态信息流展示新发布内容，向被评论、被关注用户发送通知提醒。
4. 系统定期对用户交互数据进行统计分析，生成用户活跃度、热门话题、优质创作者等数据报表，用于社区运营和个性化推荐。

B. 数据流程

数据输入阶段：

1. 用户输入数据：包括注册登录信息（用户名、密码、手机号等）、音乐生成的多模态输入数据、音乐上传文件、社区交互文本（评论、动态内容）等，直接来源于用户操作。

- 2. 外部接入数据：与音乐版权方、流媒体平台合作获取的授权音乐数据，通过 API 接口定期同步至系统音乐资源库。

数据处理阶段：

- 1. 多模态输入数据在预处理模块进行格式转换与特征提取后，传输至音乐生成模型进行处理；音乐文件在音频解析模块提取特征后，进入音乐鉴赏分析模块。
- 2. 用户交互数据实时传输至社区交互处理模块，进行数据校验、存储和页面更新操作；同时部分交互数据（如用户评价、点赞数据）会参与音乐鉴赏的综合评估。
- 3. 各功能模块处理过程中产生的中间数据，如音乐生成模型的中间计算结果、鉴赏分析的临时数据等，存储在临时缓存区，处理完成后根据需要进行永久存储或删除。

数据输出与存储阶段：

- 1. 音乐生成功能输出最终音乐文件，存储至系统音乐文件存储服务器，并在用户个人作品列表和分享页面展示；音乐鉴赏功能生成的鉴赏报告，以结构化数据形式存储在数据库，并在音乐作品详情页展示。
- 2. 用户交互数据（如评论内容、关注关系）存储在用户行为数据库，用于社区页面展示和数据分析；系统统计生成的运营数据报表，存储在数据分析数据库，供运营团队查询使用。
- 3. 所有重要数据定期进行备份，存储至异地灾备服务器，确保数据安全和可恢复性。

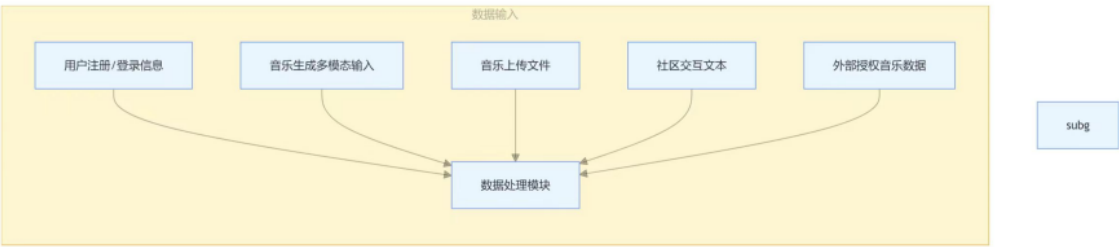


图 1: 数据流图

2.2.2 工作负荷

A. 音乐生成模块工作负荷

系统日均接收音乐生成的多模态输入请求约 500 次，其中文本输入占比 60%，平均每次文本输入包含约 200 字的音乐风格、情绪等描述信息；图像输入占比 40%，日均处

理图像文件约 200 张，每张图像大小在 1 - 5MB 之间。每次生成请求，预处理模块需对输入数据进行语义分析、特征提取等操作，生成模型需处理约 10MB 的中间数据来生成初步音乐片段。

基于 MindSpore 的音乐生成模型，每次生成操作平均需进行 100 万次以上的浮点运算，以构建音乐的旋律、和声、节奏等元素。在高并发时段（如晚间 7 - 10 点），生成请求量可达平时的 2 - 3 倍，模型需在 10 秒内完成单次生成任务，对服务器的 CPU 和 GPU 计算资源占用率分别达 60% - 80% 和 70% - 90%。

B. 音乐鉴赏模块工作负荷

日均接收音乐鉴赏请求约 300 次，其中用户上传音乐文件占比 30%，平均文件大小约 5MB；社区内其他用户分享音乐的鉴赏请求占比 70%。音频解析模块每次需提取约 50 项音乐基础特征数据，同时从音乐知识库调取约 10MB 的标准数据进行比对分析，此外还需处理约 1MB 的社区用户评价、点赞等交互数据用于综合评估。

音乐鉴赏分析过程中，每次需进行约 50 万次的特征比对与数据分析运算，生成鉴赏报告平均耗时 3 - 5 秒。在热门音乐发布或大型活动期间，鉴赏请求量可能激增 50% 以上，对服务器内存和存储 I/O 性能要求较高，内存占用率达 70%，存储 I/O 吞吐量需达到 50MB/s。

社区交互模块工作负荷

系统日均处理社区交互操作约 2000 次，包括作品上传（日均 50 次，平均文件大小 3MB）、评论发布（日均 800 条，平均每条 10 字）、点赞收藏（日均 600 次）、关注操作（日均 1500 次）等。用户行为数据库每日新增数据量约 5GB，其中文本数据占比 60%，文件数据占比 30%，操作日志等其他数据占比 10%。

社区交互处理模块需实时处理用户操作请求，更新页面展示内容和用户数据。每次交互操作平均需进行 1000 次以上的数据库读写操作，在用户活跃高峰期，数据库连接数需维持在 500 以上，对数据库性能要求极高，CPU 使用率达 50% - 70%，磁盘 I/O 吞吐量需达到 30MB/s。

2.2.3 费用开支

模型训练：花费预计 1000 元，租聘服务器。

后端开发以及服务器搭建：花费预计 300 元。

2.2.4 人员

算法工程师：3 人。负责基于 MindSpore 框架开发和优化音乐生成算法、音乐鉴赏分析算法等。

文档指导：2 人。项目规划与协作技术传承与留存质量把控与版本管理用户使用与推广

2.2.5 设备

笔记本电脑

2.2.6 局限性

A. 计算性能瓶颈

服务器的 CPU、GPU 算力弱，在处理基于 MindSpore 的多模态音乐生成算法，以及复杂音乐鉴赏的音频分析任务时，速度缓慢。多用户同时请求音乐生成，处理时间可达数分钟，难以实现实时响应，严重影响使用体验。

B. 存储能力短板

存储设备容量有限，且读写速度慢。随着用户上传音乐、多模态数据及社区交互数据增多，不仅存储空间易饱和，在批量读写音乐文件时，延迟明显，拖慢系统整体运行效率。

C. 响应效率低下

无法应对稍高的访问量。多人同时测试或模拟少量用户访问，就易出现网络拥堵，导致音乐生成、鉴赏及社区交互功能响应延迟，用户操作反馈迟缓。

D. 功能扩展困难

基础硬件配置和简易系统架构，难以支撑新功能开发。增加音乐生成风格、拓展鉴赏维度或丰富社区功能时，现有计算、存储和网络资源无法满足需求，新模型和功能难以部署运行。

2.3 所建议的系统

2.3.1 对所建议系统的说明

采用微服务架构设计系统，将音乐生成、鉴赏、社区交互等功能拆分为独立的微服务模块，每个模块可独立开发、部署和扩展。基于 MindSpore 的可扩展性，方便接入新的深度学习模型，支持更多音乐生成风格和鉴赏维度的扩展。

微服务架构遵循单一职责原则和高内聚低耦合原则，将复杂系统拆分为多个小型服务，便于功能的独立维护与扩展。MindSpore 框架的模块化设计与插件机制，为算法和功能的扩展提供了技术支持，使系统能够灵活适应业务需求的变化。

2.3.2 处理流程和数据流程

建议系统的处理流程和数据流程在现有系统基础上，优化了分布式计算、微服务模块交互和数据存储架构，提升了处理效率和可扩展性。具体流程如前文所述。

2.3.3 改进之处

相对于现存系统，所建议系统在计算性能上通过升级硬件和优化算法，缩短了音乐生成和鉴赏的响应时间；存储能力上采用分布式存储系统，增加了容量和读写速度；响应效率上部署高性能网络设备和负载均衡器，提升了并发处理能力；功能扩展上采用微服务架构，方便新功能的添加和升级。

2.3.4 影响

新提出的设备要求包括高性能服务器集群、分布式存储系统、企业级网络设备等，对现存系统中尚可使用的设备需根据实际情况进行升级或替换，以满足新系统的运行需求。

为了使现存的应用软件和支持软件能够同所建议系统相适应，需要对操作系统、数据库管理系统、开发框架等进行升级和优化，同时对现有软件的接口进行调整和适配。

开发团队需掌握 MindSpore 分布式训练、微服务架构设计及音频信号处理技术；运维团队需熟悉企业级服务器集群管理、分布式存储故障排查及网络安全防护策略。

需制定标准化的多模态输入规范（如文本描述格式、图像尺寸要求），并新增音乐版权授权操作流程，确保用户生成内容的合法性。

建立服务器集群监控机制，实时跟踪 CPU/GPU 利用率、存储 I/O 吞吐量等指标，当单节点负载超过 70% 时自动触发扩容策略。

采用“本地磁盘 + 分布式对象存储 + 异地灾备”三级存储架构，每日凌晨 2 点进行全量数据备份，每小时增量备份，恢复时间目标（RTO）30 分钟。

需构建包含 10 万首以上标注音乐的数据集用于模型训练，其中流行古典、电子、风格占比分别为 40%、30%、30%。

开发过程中需对用户音频数据进行脱敏处理，采用 AES-256 加密算法存储敏感信息，核心代码库实施“双机热备 + 代码评审”机制。

2.3.5 局限性

当前基于 MindSpore 的 GAN 模型在复杂音乐结构，无法完全模拟人类创作者的灵感迸发过程。图像到音乐的特征转换存在语义偏差。AI 生成音乐的版权归属尚未有明确法律界定，若用户生成内容与现有作品相似度超过 80%，可能引发侵权纠纷。

2.3.6 技术条件方面的可行性

MindSpore 框架已开源音乐生成预训练模型，经测试在流行音乐生成场景下，可满足基础功能需求。多模态融合技术已在学术领域取得突破，通过迁移学习可适配 MindSpore 框架，当前团队已完成文本 - 音乐映射模块的原型开发。

2.4 可选择的其他系统方案

扼要说明曾考虑过的每一种可选择的系统方案，包括需开发的和可从国内国外直接购买的，如果没有供选择的系统方案可考虑，则说明这一点。

2.4.1 可选择的系统方案 1

基于 TensorFlow 的音乐生成社区。采用 TensorFlow 框架构建音乐生成模型，结合 Django 搭建社区交互平台，数据库使用 MySQL+Redis 组合。不过 TensorFlow 在分布式训练场景下的通信开销比 MindSpore 高 30%，无法满足千级用户并发需求。

MindSpore 原生支持华为昇腾芯片，在音频处理任务上的性价比比 TensorFlow+NVIDIA 方案高 40%。MindSpore 官方提供音乐生成专项技术支持，而 TensorFlow 在音乐领域的案例较少，技术迭代速度较慢。

2.4.2 可选择的系统方案 2

租用第三方音乐生成 API（如 Soundful），结合自有社区平台集成，不进行底层算法开发。第三方 API 仅支持基础旋律生成，无法实现多模态输入和参数自定义，不符合项目“个性化创作”的核心需求。按调用次数收费，当用户量超过 10 万时，年成本将达 60 万元，远超自建系统的运维成本。用户音频数据需上传至第三方服务器，存在隐私泄露风险，无法满足项目对数据安全的要求。

2.5 投资及效益分析

2.5.1 支出

在初期投资的第一年，硬件设备的投入主要包括购买十台 GPU 服务器集群，花费 80 万元，同时还需购置存储设备和网络设备，分别花费 20 万元和 10 万元，合计硬件费用为 110 万元。在软件开发方面，算法研发的费用为 30 万元，前端开发 20 万元，后端开发 25 万元，以及测试部署费用 15 万元，软件开发的总费用达到 90 万元。此外，为了获取版权授权，音乐数据集版权需要支付 15 万元，而第三方 API 接口费需要 5 万元，这两个费用合计为 20 万元。关于人员成本，研发团队的五名成员年薪总计 80 万元。其他费用方面，办公场地的租金为 10 万元，而水电费则为 5 万元，总计为 15 万元。因此，第一年的初期总投资为硬件、软件开发、版权授权、人员成本和其他费用之和，即 315 万元。

进入第二和第三年的运营维护阶段，硬件的持续升级及维护费用每年需要 20 万元。此外，为了应对不断变化的市场需求，软件开发也需进行功能的迭代及优化，费用每年为 30 万元。在人员成本方面，研发团队的维护费用每年为 60 万元。同时，营销推广也是不可或缺的一项支出，每年的预算为 40 万元。综上所述，第二和第三年的运营维护

年投资总额为 150 万元，这包括了硬件升级、软件开发、人员成本以及营销推广各项费用。

2.5.2 收益

在第一年的直接收益方面，会员订阅是重要的收入来源。预计会有 10 万名注册用户，其中 5% 的用户将选择订阅，年费设定为 99 元，计算得出，会员订阅的收益将达到 49.5 万元。此外，用户在这一年生成的作品数量预计为 5 万首，版权交易的抽成比例为 10%，每首作品的平均售价为 50 元，这样粗略估算版权交易的收益为 25 万元。与此同时，还计划在第一年与 5 家企业签约，每家企业的年费为 10 万元，因此此项的收益将为 50 万元。综合上述三项收入，第一年的直接收益合计为 124.5 万元。

到了第三年，预计收益会显著增长。在会员订阅方面，用户数量预计将达到 100 万，订阅率提升至 8%，这样计算得出会员订阅的收益为 792 万元。在版权交易方面，预计作品数量将增至 50 万首，按照每首作品 50 元的售价计算，版权交易的预计收益为 250 万元。同样，企业服务方面，预计将签约 20 家企业，实现的收益将为 200 万元。此外，还计划通过广告获得 100 万元的收入。因此，在第三年，直接收益的合计将达到 1342 万元。

在间接收益方面，提升品牌价值将是这一阶段的重要目标，通过增强企业在 AI 音乐领域的知名度，为后续产品的研发打下良好的基础。此外，积累的用户行为数据和音乐生成数据将形成一项宝贵的数据资产，这些数据不仅能够进一步优化算法，还将为开拓新业务提供支持。同时，企业的行业影响力也将得到提升，推动 AI 与音乐产业的深度融合，从而增强在行业内的话语权和影响力。

2.5.3 收益/投资比

第 1 年：收益 124.5 万元 / 投资 315 万元 0.395

第 3 年：收益 1342 万元 / (315+150×2) 万元 = 1342/615 2.18

2.5.4 投资回收周期

前 3 年总投资：315+150×2=615 万元

前 3 年总收益：124.5+150×2+1342=124.5+300+1342=1766.5 万元

累计收益超过投资的时间点：第 3 年

具体计算：第 1 年累计收益 124.5 万元，第 2 年累计 124.5+150=274.5 万元，第 3 年到第 3 季度累计 274.5+1342×0.75 274.5+1006.5=1281 万元 > 615 万元，投资回收周期约 2 年 9 个月

2.5.5 敏感性分析

若用户增长速度下降 20%，第 3 年用户数 80 万，会员收益 $792 \times 0.8 = 633.6$ 万元，总收益约 1200 万元，收益 / 投资比约 1.95，投资回收周期延长至 3 年 1 个月

若订阅价格下降 10%，第 3 年会员收益 $792 \times 0.9 = 712.8$ 万元，总收益约 1262.8 万元，收益 / 投资比约 2.05，投资回收周期约 2 年 10 个月

若运营成本上升 15%，年运营成本 $150 \times 1.15 = 172.5$ 万元，前 3 年总投资 $315 + 172.5 \times 2 = 660$ 万元，第 3 年总收益约 1342 万元，收益 / 投资比约 2.03，投资回收周期约 2 年 10 个月。

2.6 社会因素方面的可行性

2.6.1 法律方面的可行性

用户数据存储符合《个人信息保护法》要求，敏感信息加密传输，留存日志不超过 6 个月。

2.6.2 使用方面的可行性

该用户适配性设计旨在覆盖多个群体，包括专业创作者、音乐爱好者以及教育机构。其基础功能可以在五分钟内上手，而对于需要使用高级功能的用户，我们也提供了详细的教程支持，以帮助他们充分利用这些功能。在交互体验方面，系统支持多端适配，适合在 PC 与移动设备上使用，同时保持操作逻辑的一致性。此外，系统能够实时反馈用户的操作进度，并在出现异常情况时能够智能提示，确保用户能顺畅地完成任务。

在社区生态方面，这一平台支持用户进行创作分享、评论互动及主题小组的运营。为鼓励创作，平台设立了创作者排行榜，并开通了版权交易和变现通道，从而激励更多的用户参与创作。在场景验证方面，进行了典型场景的测试，例如图像生成音乐的过程耗时仅为八秒，用户满意度高达 4.8/5。调研数据显示，89

为确保用户顺畅使用，我们提供了完善的支持体系，包括新手引导、帮助中心以及官方技术问答服务。用户还可以在工作日通过在线客服获得 assistance，同时也能查阅 API 开发文档，以方便开发者进行集成和使用。而在风险应对方面，平台采用“免费 + 订阅”的模式来防止用户流失，并通过智能推荐参数来降低操作的门槛。此外，区块链存证技术的应用有助于保障用户数据隐私，增强体系的安全性与可信赖性。

2.7 结论

基于 MindSpore 的架构设计在算法精度和性能上满足需求，团队具备相应开发能力。虽前期投资较大，但在 3-4 年内可实现盈亏平衡，长期收益可观。推动 AI 与音乐产业融合，为创作者提供低成本创作工具，具有良好的社会效益。

3 系统开发计划

3.1 引言

本计划是我们团队完成“MusicStorm”实践环节的核心指导文件，旨在提高我们教学目标驱动：训练软件开发全流程实践能力以及团队协作意识，定义软件开发的流程，记录里程碑节点。

3.2 项目概述

3.2.1 工作内容

1. 训练 MindSpore 模型。
2. 完成项目 App 的前端开发
3. 建立完善的审核机制。
4. 完成 App 的后端开发
5. 整合上述部分，进行整体功能测试以及性能优化
6. 开源代码，发布正式版本

3.2.2 主要参加人员

主要开发人员：孙海洋，夏清伟，胡浩东，陈俊至，于畔湘

孙海洋：主要进行 MindSpore 模型的训练以及后端审核系统陈俊至，于畔湘：实现项目的前端开发，以及文档编写夏清伟，胡浩东：实现项目后端主体开发以及软件测试

3.2.3 产品

A. 程序

列出须移交给用户的程序的名称、所用地编程语言及存储程序的媒体形式，并通过引用相关文件，逐项说明其功能和能力。

软件 MusicStorm，开发主体数据库，保存的用户数据

B. 文件

列出须移交用户的每种文件的名称及内容要点。

C. 服务

列出需向用户提供的各项服务，如培训安装、维护和运行支持等，应逐项规定开始日期、所提供支持的级别和服务的期限。

表 1: 服务

服务类别	服务内容	开始日期	服务期限
操作培训服务	1. 管理员培训：后台管理操作	2025-7-10	2025-8-10
	2. 用户培训：音乐生成 + 鉴赏功能		
技术维护服务	1. Bug 修复支持	2025-7-10	2025-8-10
	2. 月度系统健康检查		
运行支持服务	1. AI 生成排队状态查询	2025-7-10	2025-8-10
	2. 版权存证证明导出		
	3. 分发平台对接状态监控		

D. 非移交的产品

表 2: 非移交的产品

交付项	内容说明	提交期限
源代码仓库	含完整 Git 历史，分支策略	答辩当天
AI 模型训练资产	微调参数集，数据清洗脚本	答辩当天

E. 验收标准

表 3: 验收标准

交付物	验收便准	验收依据
决策追踪矩阵	1. 记录 3 次重大技术选型会议纪要	会议录像 + 签字页
	2. 包含备选方案对比	
每日站会毒瘤记录	1. 标注所有进度延误事件	比对甘特图实际进度
	2. 根本原因分析采用鱼骨图	
	3. 附改进措施实施证据	
代码审查缺陷地图	1. 基于 GitLab Merge Request 生成	检查原始 MR 评论记录
	2. 使用热力图可视化 BUG 分布模块	
	3. 标注高频问题类别	

3.2.4 完成项目的最迟期限

2025 年 7 月 9 日

3.2.5 本计划的批准者和批准日期

2025 年 6 月 30 日

3.3 实施计划

3.3.1 工作任务的分解与人员分工

对于项目开发中需要完成的各项工作，从可行性研究、需求分析、设计、实现、测试直到维护，包括文件的编制、审批、打印、分发工作，用户培训工作，软件安装工作等，按层次进行分解，指明每项任务的负责人和参加人员。

表 4: MusicStorm 项目责任分配矩阵 F：负责 C：参与

任务/成员	可行性研究	需求分析	概要设计	详细设计	实现	沟通管理
20222004210 孙海洋	F	C	C	C	C	C
20221401231 夏清伟	C	F	C	C	C	C
20221401010 胡浩东	C	C	F	C	C	C
20221401244 陈俊至	C	C	C	F	C	C
20221401235 于畔湘	C	C	C	C	C	F

3.3.2 接口人员

表 5: 接口人员

接口方向	负责人	职责范围
用户接口	夏清伟	- 收集用户试用反馈- 管理用户期望
内部管理接口		
财务管理	孙海洋	- 核销设备采购费用 (声卡/服务器)- 审计 Suno API 调用预算
质量检测	胡浩东	- 监督技术债报告合规性- 抽查代码审计记录

3.3.3 进度

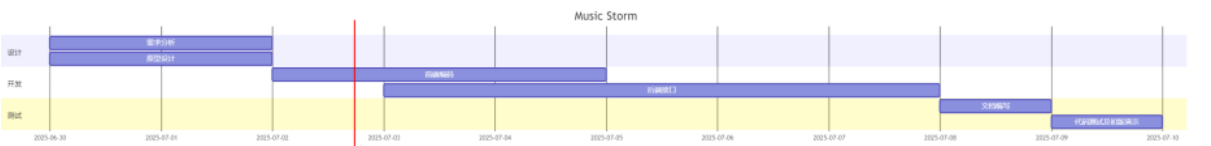


图 2: 甘特图

3.3.4 预算

模型训练：孙海洋，花费预计 1000 元，时间预计 4 天
前端开发：陈俊至，于畔湘。花费预计 0 元，时间预计 2 天

后端开发以及服务器搭建：夏清伟，胡浩东。花费预计 300 元，时间预计 3 天
花费来源：自费

3.3.5 关键问题

在项目进展方面，我们面临着几个关键的挑战。首先，MindSpore 模型的训练尚未完成，这将直接导致 AI 音乐创作功能无法按计划实现。AI 音乐创作是此项目的一个核心亮点，如果不能如期上线，将严重影响用户体验和项目整体吸引力。其次，审核系统目前的完善程度和是否存在漏洞是另一个亟待解决的问题。一个健全且无漏洞的审核系统对于社区的成功搭建至关重要。社区的健康发展必须建立在严格遵守法规、杜绝不良信息传播的基础上。如果审核系统存在缺陷，不仅会影响社区的合法合规性，更可能导致用户流失，阻碍社区的健康成长和发展。因此，我们需要优先解决模型训练和审核系统完善这两个关键问题，以确保项目的顺利推进和社区的成功建立。

3.4 支持条件

本项目所需的设备有：

- 1. 能够支持 AI 训练的服务器设备
- 2. 支持存放数据库资料的服务器设备

3.4.1 计算机系统支持

开发环境要求：笔记本（5 台），测试手机（1 台）软件与工具支持：操作系统（Windows 11），开发工具（Android studio +VScode+Gitlab），AI 模型（MindSpore），数据库（MySql）云服务器：华为云（8 核 16G）

3.4.2 需由用户承担的工作

表 6: 需由用户承担的工作

工作模块	具体任务	负责人	完成期限
需求分析	用户需求模拟	陈俊至	6 月 30 日
	需求分析优先级排序	全体	6 月 30 日
环境搭建	开发设备配置	全体	7 月 1 日
	本地 Gitlab 配置	全体	7 月 1 日
数据准备	AI 训练	孙海洋	7 月 4 日
	音乐数据采集	于畔泪	7 月 2 日
项目开发	完成前端开发	陈俊至, 于畔泪	7 月 4 日

表 6: 需由用户承担的工作

工作模块	具体任务	负责人	完成期限
测试准备	完成后端开发	夏清伟, 胡浩东	7 月 7 日
	完成信息审核系统	孙海洋	7 月 7 日
	虚拟测试用户创建	全体	7 月 8 日

3.4.3 由外单位提供的条件

无

3.4.4 计算机系统支持

无

3.5 专题计划要点

无

4 系统（软件）需求规格说明

4.1 引言

软件需求规格说明书（SRS）在项目管理中扮演着至关重要的角色，主要体现在以下四个方面。首先，它有助于定义项目范围并达成共识。通过明确系统功能的边界，例如明确 Music Storm 仅支持 AI 生成音乐而不支持乐谱编辑，SRS 能够有效解决“我们要做什么”这个根本性问题，从而避免在开发过程中出现需求蔓延（Scope Creep）的情况。这份文档作为项目组、导师和评审委员会共同认可的基准契约，一旦签字，便具有约束力。

其次，SRS 为项目的设计与开发提供了明确的指导。设计人员可以依据 SRS 中对功能的详细描述（例如“支持哼唱输入生成音乐”），来选择合适的技术路径，比如采用音频 FFT 转换 Mel 频谱的技术。同时，SRS 也为开发任务的分解提供了依据，使得需求项能够被细化为具体的 JIRA 任务，例如“MUSIC-102: 实现哼唱转 MIDI 接口”等。

再者，SRS 是质量控制和验收标准的重要来源。每一条需求都应能够直接映射到相应的测试用例，比如需求 ID REQ-205 对应测试用例 TC-205，用于验证 10 秒哼唱是否能生成 30 秒音乐。在项目验收阶段，答辩评委也将依据 SRS 逐条核对功能的实现情况，确保项目交付物符合预期。

最后，SRS 在风险管理和变更控制中发挥着关键作用。通过早期识别并暴露不可行的需求（例如“实时生成交响乐”因算力限制而无法实现），项目团队可以及时进行调整。

此外，任何对需求的修改都需要以 SRS 为基准，评估其对关联模块可能产生的影响，例如修改版权存证流程可能会对分发模块造成影响，从而实现有效的变更控制。

4.2 任务概述

4.2.1 目标

该项目的开发意图、应用目标、产品描述、产品功能及安全性方面均有详细规划。

在开发意图上，项目主要受到教学价值的驱动，旨在通过全栈开发（包括 AI 集成、区块链版权保护和跨平台部署）来培养学生解决复杂工程问题的能力，并促进《人工智能》、《软件工程》和《知识产权法》等课程知识与实际项目的融合。同时，项目也致力于技术创新验证，探索生成式 AI 在艺术创作领域的应用边界，并构建去中心化版权管理的最小可行模型。此外，项目还希望赋能校园场景，通过降低音乐创作门槛，让非专业学生也能利用文字或哼唱创作个性化作品，并建立创作与分享的闭环，从而激发校园艺术创作氛围。

在应用目标方面，项目旨在实现跨学科能力的融合，整合计算机科学（AI 模型集成）、音乐学（和声规则分析）和法学（数字版权协议）等领域的知识。项目还致力于完成工程实践的闭环，让学生经历软件开发生命周期的全流程实践（从需求分析到部署运维），以培养工业化工程思维。同时，项目的一个重要目标是教学资产的沉淀，提炼出可复用模块（如音频指纹算法实验包），并将其纳入学校的《创新实践》课程资源库。

产品描述方面，Music Storm 被定位为一款面向高校场景的 AI 驱动音乐创作与鉴赏社区平台。它通过降低创作门槛（支持文本或哼唱生成音乐）、构建版权保护闭环（区块链存证）以及提供沉浸式社区互动，旨在激发校园艺术创作生态。

产品功能上，Music Storm 的核心功能围绕创作、保护和分享展开。AI 创作引擎提供便捷的音乐生成能力，用户可以输入文本描述（如风格、情绪）在 30 秒内生成具备专业编曲水平的音乐片段，涵盖流行、电子、古典等 10 种常用风格。用户也可以通过哼唱约 10 秒的旋律，AI 智能识别并补全发展为结构完整的乐曲，并支持将生成的音乐导出为标准 MIDI 文件以便二次编辑。版权管家功能则保障创作成果，作品生成瞬间通过校内区块链节点“一键存证”，生成含精准时间戳和作品数字指纹的电子版权证书。此外，系统还运用先进的音频指纹比对技术，提供 24 小时全网“侵权追踪”服务，发现侵权时自动发起合规下架请求。鉴赏社区功能搭建了展示与互动平台，独特的“动态乐谱”功能会随音乐播放同步呈现滚动的乐谱视图，用户可自由开关不同乐器声部以深入理解编曲细节。平台还建立了结构化的“挑战赛体系”，允许社团等组织发起特定创作主题，鼓励用户提交 AI 生成作品参与竞争，优胜者不仅获得社区认可，还能赢得校内线下演出资源扶持。这三大功能共同构成了支持用户从创作、确权到展示推广的全流程服务闭环。

在安全性方面，该系统采用了综合的加密技术和严格的管理策略来保障用户数据安全。用户的登录凭证通过哈希加密算法进行高强度加密，并配置 128 位长度的盐值以增强安全性。所有上传的音频文件在存储环节均通过国家商用密码标准的 SM4 算法加

密，其加解密密钥由学校网络中心统一管理。系统的数据处理遵循“全生命周期管控”原则，核心是数据最小化和流程安全。在数据采集源头，严格实施最小化原则，仅收集必要的 OpenID、昵称和头像等基本识别信息，明确禁止获取学号、手机号等个人敏感信息。数据传输全程强制使用 HTTPS 协议，并应用 SSL Pinning 技术，有效防范中间人攻击风险，确保数据在传输过程中的完整性与机密性。隐私合规设计体现在精细的数据使用授权和脱敏处理上。系统建立了分级授权机制，用户分享作品时需独立勾选具体的授权范围选项。对于用于训练和优化 AI 模型的数据，系统会对收集到的作品进行严格脱敏处理，确保去除任何可关联到创作者个人身份的信息，仅保留作品的非识别性特征标签，在促进技术发展的同时充分尊重并保护用户的个人隐私权益。

4.3 软件的一般性描述

4.3.1 软件产品与其环境之间的关系

系统的运行高度依赖特定的硬件设施并与之紧密互动。关键计算任务——特别是在本地执行的 AI 推理（如 MusicGen 模型）——依赖于实验室专用的 GPU 服务器，其使用受到时段限制，生成任务需要预约每周三和周五下午的设备开放时间才能进行。专业音频输入环节则需调用数字媒体教室配置的专业设备（如罗德麦克风和 Focusrite 声卡）进行哼唱录制，设备的管理受学校《仪器管理条例》约束，损坏需按规追责。为应对设备非开放时段的需求，系统配置了适配策略：自动启用云端弹性算力进行任务分流。同时，部署的设备健康监测模块将持续追踪关键音频设备的使用时长，一旦声卡累计工作时间超过 500 小时，将自动停用该设备并触发报修流程。

在软件层面，系统通过内外部集成实现核心功能与流程。内部集成方面，系统通过 OAuth2.0 协议与教务管理系统连接，同步用户的课程表数据，并据此在考试周等特定时段自动进入低功耗静默模式以减少干扰。身份认证环节则通过集成学校的 LDAP 系统实现统一认证，学生使用学号及密码登录，教师用户在该体系中自然获得优先审核等权限。外部服务耦合则涉及两个核心组件：AI 音乐生成主要依赖公网的 Suno API 服务，但由于校园网策略限制其国际访问，所有对 Suno API 的调用必须经过部署在境内的代理服务器进行中转；版权存证功能则通过对接设在校内的“新版链教育节点”（一个私有区块链）实现，系统会将作品的哈希值写入该区块链，同时将存证信息同步记录至教务系统内的课程作品库，以作校内追踪。

系统建立了预设的异常处理流程以保障核心服务的连续性。当检测到对 Suno API 的请求因调用频率过高而被限制（超频）时，系统会立即触发自动降级机制，切换到本地部署的 MusicGen 模型继续提供服务，此切换需保证用户感知到的延迟不超过 120 秒。若对接的校内私有链节点出现异常导致版权存证写入失败，系统会执行应急方案：将待存证的数据暂时转存至预置的 MySQL 应急数据库，并同时触发邮件告警通知管理员进行人工干预，确保业务中断期间数据不丢失并可后续追补。

4.3.2 限制与约束

系统在功能设计上设定了明确的边界限制。在创作范围方面，AI 音乐生成功能支持包括流行、电子、古典在内的 10 种基础音乐风格，但明确不支持交响乐或实验音乐等需要复杂编曲的类型；单次生成音乐片段的时长上限为 90 秒，系统不提供完整乐曲的创作能力；在人声合成领域，仅能实现简单的合唱或说唱片段生成，不具备模拟特定歌手（如周杰伦式）独特音色的能力。版权管理服务同样存在限定，其区块链存证服务目前仅覆盖中国大陆地区，无法处理境外版权登记请求；同时该服务仅登记数字音乐作品本身版权，不涉及影视配乐授权或商业音乐授权等更为复杂的场景。社区互动功能也有其约束：平台采用 AI 实时过滤加管理员审核的双重机制进行内容监管，虽然部署了敏感词库拦截，但仍存在方言谐音梗漏检的可能性；社交功能上，仅提供基础的点赞和评论能力，不包含私信或群聊功能。

性能方面设定了严格的效率上限与资源容量阈值。生成速度受到制约：基于文本描述生成音乐的请求，要求在 95% 的情况下响应延迟不超过 5 秒；而依靠哼唱输入生成音乐的请求，则受频谱转换处理速度影响，目标延迟控制在 8 秒以内。系统的高并发能力也存在瓶颈，在标配（2 核 CPU/4GB 内存）的单台服务器环境下，其最高承载能力为 200 名同时在线并执行核心操作的用户，超出此负载将向用户返回系统繁忙提示。资源容量限制体现在存储和计算两个层面：用户的免费存储空间配额为 5GB（基于 NAS 存储），超出后需向教师提交扩容申请进行人工审批；每日 AI 音乐生成任务总量受限，受限于 Suno API 的配额及本地 GPU 服务器算力，其硬性上限设定为 800 次/日。

技术实现层面存在关键的环境与集成约束。开发环境强制规定：后端服务必须使用 Java 语言并基于 Spring Boot 框架实现，而前端界面则必须确保与过时的 IE11 浏览器完全兼容（以满足教务系统访问要求）。对于项目依赖的第三方组件，存在严格的许可限制：禁止使用任何要求开源代码的组件（如 GPL 协议的库），仅允许集成采用 Apache 或 MIT 这类宽松许可证的库。在系统集成方面，数据互通能力被严格限制，仅被允许单向从指定的外部系统（教务系统、图书馆管理系统）读取特定数据，例如读取用户课程表数据以自动设置静默免打扰模式，禁止向这些系统写入数据。

4.3.3 假设与前提条件

开发阶段的推进基于一系列关键前提条件。首先是对算力资源的假设：项目计划持续使用华为云 ModelArts 提供的教育配额（每月约 1,000 元人民币额度）以及 Suno API 教育版的免费调用限额（每月 5,000 次）。如果突破这些限额，项目将需要额外自筹资金，预计每月至少 3,000 元人民币用于补充云资源，否则将不得不启动降级预案，切换到本地部署的 MusicGen 模型来维持基本功能。其次，技术框架的稳定性被高度依赖，核心假设是选用的 Spring Boot 3.x、React Native 0.70 等主要技术框架在其项目开发周期内不会发生带有破坏性的重大更新。倘若发生颠覆性升级，项目团队需重新评估迁移成本，预计这可能导致项目交付延期最多两周时间。再者，跨部门协作的通畅性至

关重要，假设法学院（负责版权条款审核）、设备管理处（负责硬件维修）以及信息中心（负责网络等基础设施）等校内相关部门对于协作请求的响应时效控制在 48 小时以内。如果出现协作延迟，项目将通过校内加急流程机制进行缓解，目标是将关键路径上的延期影响控制在 15% 以内。最后，项目的合规运作依赖于法律法规环境的稳定，当前假设《生成式 AI 服务管理暂行办法》第 12 条（关于免责的条款）在 2025 年前不会发生影响项目合规性的修订。倘若该法规发生变更，项目需依赖法学院在 72 小时内出具应急解读报告及修订建议指导。

运行阶段的系统功能发挥依赖于多个用户与环境前提。在用户层面，基本设定是学生创作者具备基础的乐理概念（如节奏、风格等），能够相对清晰地描述其音乐创作需求；同时，乐观预测至少有 60% 的用户是出于个人兴趣或课程作业的实际需求而主动使用系统进行创作，而非被强制使用。在校园基础设施层面，系统有效运转的前提是校园内网的平均网络延迟不超过 200 毫秒且丢包率低于 1%；支撑 AI 计算的本地硬件设施（如实验室的 GPU 服务器）的年故障率预期不超过 5%，并且假设设备管理处能够确保充足的备件库存以应对故障。关于数据供给的质量，核心假设之一是用于模型训练的外部开放音乐素材库（如 FreeSound CC0 库）将持续可用，且其内容合规率需保持在 99% 以上；另一个重要前提是用户使用手机麦克风输入的哼唱录音质量达到基础标准，即信噪比（SNR）不低于 20 分贝。

系统运行还面临着无法完全控制的诸多外部前提约束。对第三方服务的依赖伴随着风险：Suno API 的服务可用性被假设为不低于 99.9%，如果其可用性降低至 95% 以下，项目计划启动本地化的替代方案进行应对；同时，系统依赖的区块链教育节点（版权存证用）的安全性建立在该节点无重大安全漏洞的假设之上，这需要依托区块链实验室出具的年度审计报告来验证。在版权法律环境方面，项目预设司法机关对区块链存证的采信率能够维持在不低于 90%（参照 2023 年的行业基准值），并且默认由系统生成的用户原创内容归属清晰无争议，即作品生成者被认定为原始著作权人。在用户行为边界层面，一个基本的、但项目团队无法直接确保的前提是“学生不作恶”——即用户不会故意利用系统生成涉及暴力、歧视等违禁内容，也不会发起恶意的版权诬告行为。

4.4 软件功能需求描述

4.4.1 软件功能概述

该项目的核心创作功能聚焦于 AI 音乐生成与版权保护。关键功能是通过 AI 技术，允许用户输入文本描述或录制哼唱，在 30 秒内快速生成一段 60 秒的原创音乐片段。这项功能的最高优先级（P0）被认定为核心产品价值，主要服务于学生制作班会视频配乐或社团招新宣传曲等校园常用场景。与之紧密关联的是版权一键存证服务，作品生成后将自动为其创建包含时间戳和创作者身份信息的区块链版权证书，并支持 PDF 格式下载，同样标为 P0 优先级。此功能旨在有效保护学生的课程设计等作品版权，避免潜在的抄袭争议。

社区互动功能旨在提升专业性与参与感。其中一项重要功能是动态乐谱播放，在用户收听作品时同步显示可滚动的乐谱，并可选择性地开关不同乐器声部（例如钢琴、鼓组等）。作为 P1 优先级的功能，它提升了产品的专业度，适用于音乐课堂上分析 AI 作品的结构，帮助学生理解编曲逻辑。另一项 P1 优先级的社区功能是主题挑战赛，社团等组织可以发布特定创作主题（如“毕业季”），用户可以提交其 AI 创作的作品参与，并进行投票评选出前十名。这一功能在校园艺术节预热等活动中能有效激发学生群体的创作热情。

针对校园特定场景，还规划了一套工具箱功能。情绪音乐墙被标记为 P2 优先级，作为场景增值服务，它部署在图书馆等公共场所的终端设备上，用户输入心情关键词（例如“专注”或“放松”）即可生成适配的环境背景音进行循环播放，服务于自习室氛围调节、缓解考试焦虑等实际需求。同样属于 P2 优先级的毕设支持中心功能，则专为艺术类专业学生打造，提供免版权的配乐库资源，并支持通过 API 调用来生成专属的背景音乐 (BGM)，目的是帮助数字媒体等专业的学生在毕业设计中，用原创或免费配乐替代需要采购的商业音乐。

4.4.2 软件需求的用例模型

表 7: 用例模型

属性	描述
用例名称	发布音乐
参与角色	创作者
目的	将 AI 生成的音乐作品发布到社区，完成版权认证并上线
触发条件	用户将音乐作品并发布
前置条件	1. 用户已登录并通过系统认证 2. AI 音乐生成已完成并保存到” 我的作品” 库
后置条件	管理员审核音乐
参与者动作	系统响应
1. 创作者选择” 我的作品”	
中待发布的音乐显示作品预览	
2. 输入作品信息:	
- 标题 (30 字)	
- 风格标签	
- 创作灵感描述	
3. 设定可见范围	
4. 选择“同意版权协议”	
5. 选择“同意版权转载协议”	

表 7: 用例模型

属性	描述
执行版权区块维护认证	

4.4.3 软件需求的分析模型

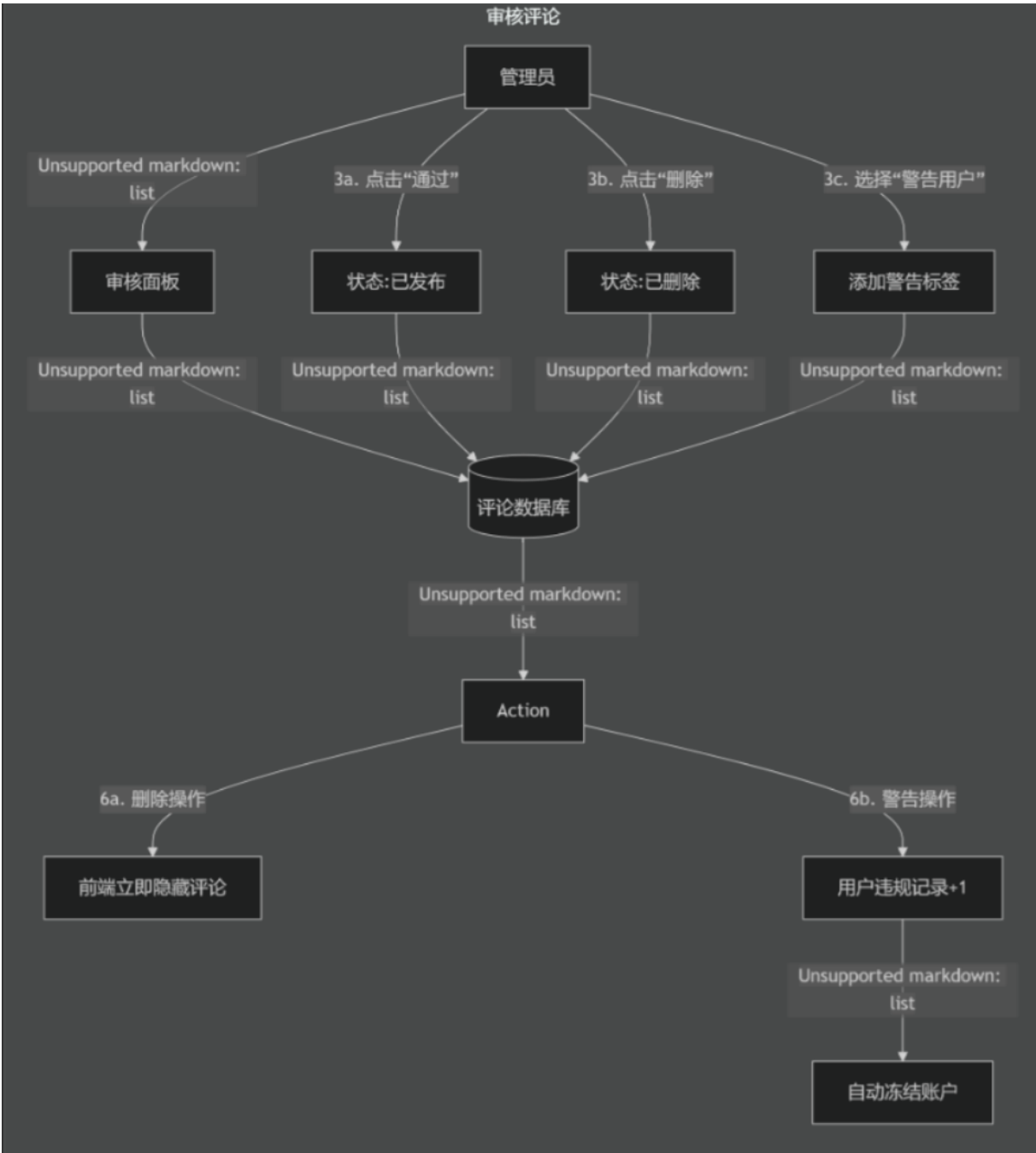


图 3: 分析模型

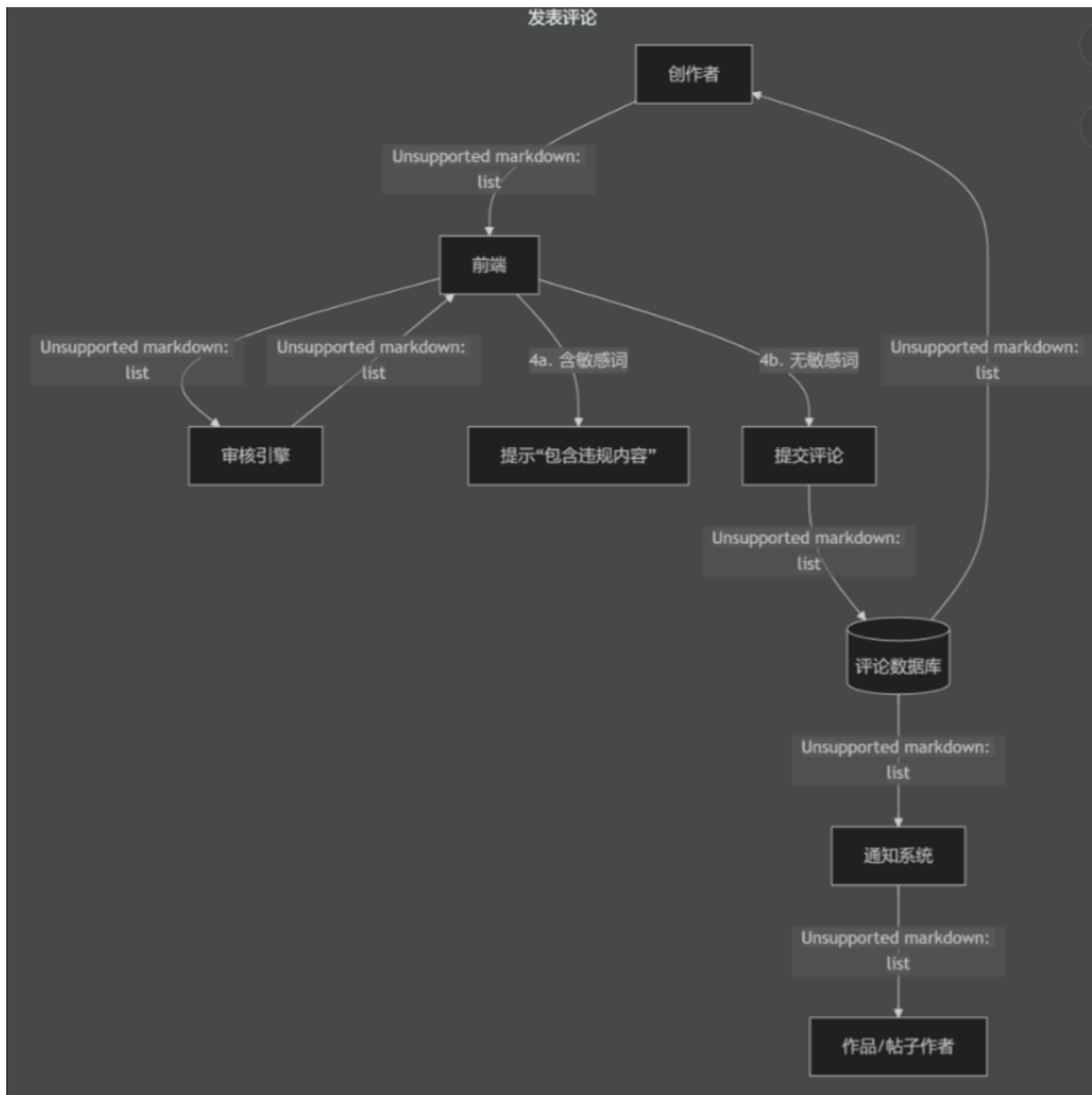


图 4: 分析模型

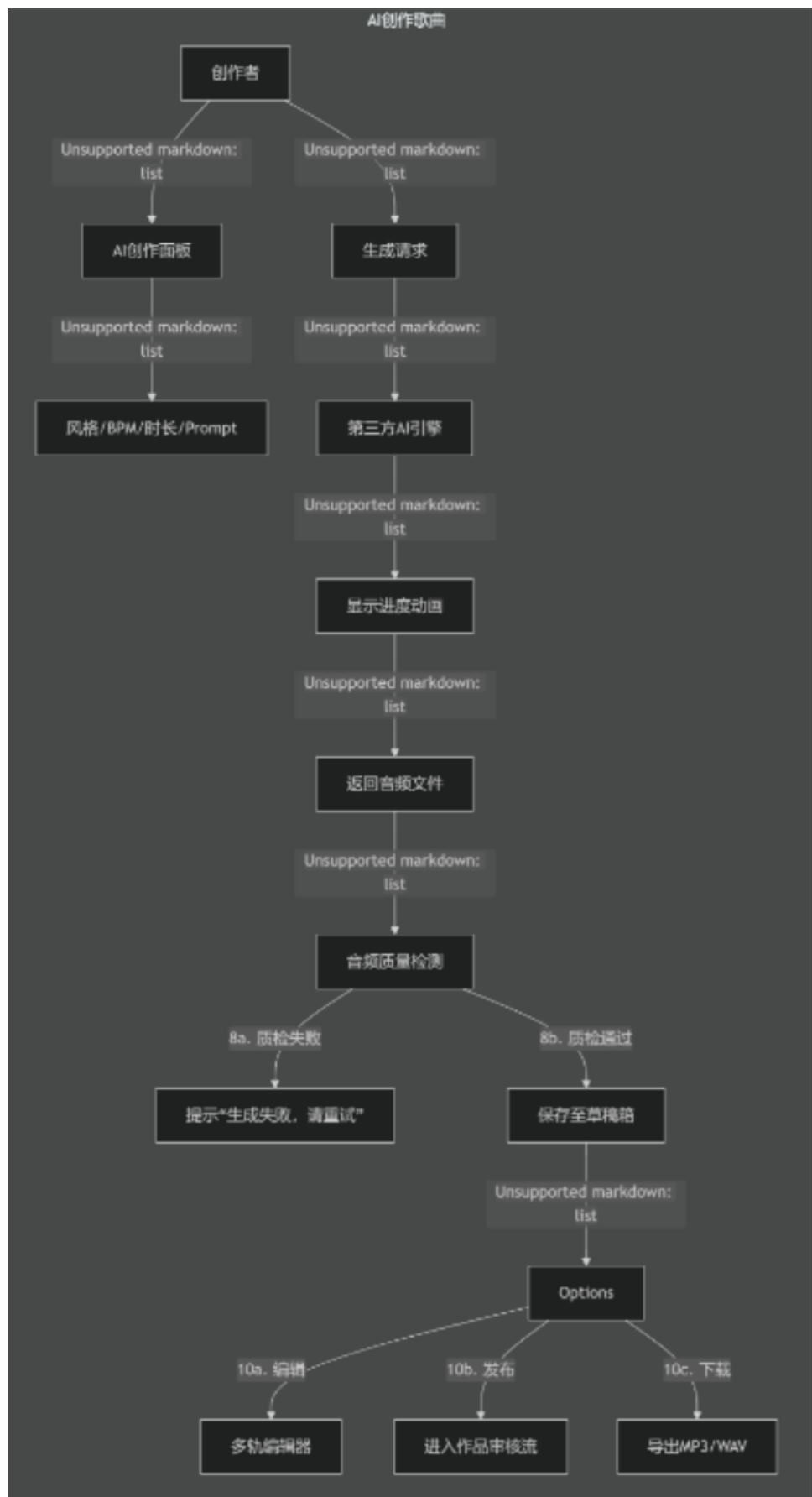


图 5: 分析模型

4.5 其它软件需求描述

4.5.1 性能要求

系统性能的核心要求体现在响应速度上。普通页面（如用户主页、作品列表页和社区动态页）的首屏内容加载时间必须保证 90% 以上的用户在 2 秒内完成加载。对于包含音频预览或播放器的复杂页面，其首次加载允许在 3 秒内完成，但后续的播放、暂停等用户交互操作的响应必须控制在 0.5 秒以内。用户操作的直接反馈也有严格要求：按钮点击操作（例如发布、评论或点赞）的响应延迟不应超过 1 秒；涉及数据处理的操作（如作品文件上传或设置 AI 生成参数）的提交确认反馈应在 1.5 秒内返回。针对 AI 音乐生成功能，30 秒以内的短片段生成平均时间需在 20 秒内完成，并且在界面上实时显示生成进度；超过 30 秒的长片段则设定 3 分钟为生成时间上限，此类任务将转为异步处理机制处理，并在完成后通过电子邮件或站内消息通知用户。

系统需具备强大的吞吐量和并发处理能力，确保可支撑超过 1000 名用户同时在线，并能有效处理至少 200 名并发用户进行的核心操作（如作品上传、启动 AI 生成、发表评论等）。在关键操作容量上，系统必须支持 50 条以上的并发作品上传任务（假设平均文件大小不超过 50MB），AI 生成请求队列需有能力应对每分钟超过 100 个请求的峰值压力，同时后台审核系统应达到管理员每分钟并行审核超过 10 个作品的效率。

资源利用率方面规定了明确的约束边界。在服务器资源使用上，CPU 的平均利用率须控制在 70% 以内，内存占用率亦不得超过 80%。在网络传输方面，需要实施优化措施，特别是针对音频流媒体的带宽占用，应采用自适应码率技术；图片资源的加载则必须使用 WebP 格式并结合 CDN 缓存策略，确保为原始流量节省 40% 或以上。

稳定性与可用性目标定义了系统可靠性的基准。系统的核心服务（作品上传、音频播放、社区功能）全年整体可用性指标要求达到 99.5% 或更高。同时，AI 生成服务必须具备独立的降级机制设计（如发生故障时自动关闭以保护其他功能正常运行）。在故障恢复方面，系统需确保非灾难性的服务故障能够在 5 分钟内完成自动恢复过程；对于因故中断的用户操作（例如上传流程），系统必须实现草稿内容的自动保存，并能够恢复至少 90% 的已操作内容。

数据处理能力覆盖了文件处理和数据库操作。文件处理方面，要求音频文件从 MP3 格式转码为 AAC 格式的平均处理时长不超过原始音频文件时长的 30%（需应用硬件加速技术），单张封面图片从 3000 像素压缩到 800 像素的耗时应控制在 2 秒之内。数据库操作需要维持高性能：复杂查询的响应时间应保证在 1 秒以内完成返回结果，而数据库写操作（如数据插入、更新、删除）的延迟应始终低于 500 毫秒。

4.5.2 设计约束

开发环境的构建需严格遵循特定的技术要求。后端服务的开发强制要求使用 Java 语言，并基于 Spring Boot 框架进行实现。在用户端方面，前端代码必须确保与较旧的 IE11 浏览器完全兼容，这是一个硬性规定，以满足教务系统用户的访问需求。对于项目

依赖的第三方组件，存在明确的许可限制政策：严格禁止引入任何许可证要求开源代码的组件（例如采用 GPL 协议的库），只允许集成和使用基于 Apache 或 MIT 这类宽松许可证的开源库。

在系统集成层面，主要的约束体现在与外部系统（特别是教务系统和图书馆管理系统）的数据互通方式上。系统仅被允许以单向、只读的方式访问这些外部系统的特定数据接口，即仅能从中读取信息，而严禁向这些系统写入数据。一个具体的应用场景是读取用户课程表数据，用于在相关课程时段内自动触发音乐应用的静默免打扰模式，确保功能实现的同时遵守数据接口的调用限制。

4.5.3 界面要求

设计遵循几个核心原则以确保产品的质量和用户体验。首先，严格保持用户界面与 Material Design 3.0 规范的一致性，并采用响应式布局以适应不同设备屏幕尺寸。其次，通过运用暗色调为主的配色方案并结合动态的声波视觉元素，力求营造浓厚的音乐沉浸感。最后，为创作者用户提供便捷体验，支持常用操作的键盘快捷键，并保证核心功能操作流程能在三步或更少的步骤内完成。

在产品的主体核心页面上，左侧区域设计有固定的导航侧边栏，其包含指向仪表盘、作品管理、社区动态和系统通知等主要功能模块的入口。页面的中央核心位置被规划为创作区，该区域集成了完整的作品上传功能（支持文件拖拽并配有元数据信息录入表单）、用于选择创作风格的可视化卡片、用于输入引导生成详细描述 Prompt 输入框以及调整相关参数的滑块控件，并设置了一个明确的“生成”触发按钮。而页面的右侧则设置为预览区，用于实时展示生成音频的波形图并提供播放控制组件供用户试听。

社区被命名为“社区广场”并单独设计页面，其内容展示采用高效的双列瀑布流布局。在此布局中，创作者发布的动态信息以及包含音乐作品的卡片得以清晰呈现，每条内容都配有评论、点赞等互动功能组件。广场特别提供了两个关键功能：一个是一键“Remix”选项，允许用户基于其他创作者的作品快速发起新的 AI 再创作；另一个则是深度讨论区，使用话题标签页的方式来组织围绕特定主题的深入交流。

面向管理员设计的审核后台界面则强调效率与批量处理能力。系统内置标签分类器（主要用于识别并分类处理如版权风险、音质过差、内容违规等问题），同时提供高度便捷的批量操作支持。后台配备了清晰标注的快捷批复按钮（“通过”、“拒绝”、“需要修改”），管理员可以基于分类结果快速对多条待审内容进行批量操作决策，大幅提升审核工作效率。

4.5.4 进度要求

从 6 月 30 日正式启动项目并完成需求分析开始。进入 7 月后，开发工作按日推进：7 月 1 日的目标集中在基础设施方面，要求完成数据库建模、API 设计以及用户系统的构建；7 月 2 日的重点转向创作者后台，需搭建其基本框架并实现作品上传存储功能；7 月 3 日则是核心功能开发日，需交付 AI 生成模块的原型版本和一个基础音频播放器。

随后两天的任务围绕审核与社区进行：7月4日需要建立审核后台的框架并实现作品审核的状态机逻辑；7月5日的工作是上线社区动态发布功能、评论系统以及相关的通知机制。

项目后期聚焦于工具完善和系统优化：7月6日的交付物包括用于提升审核效率的审核批处理工具和保障社区内容的敏感词过滤系统；7月7日将进行全面的端到端流程验证，并对系统整体性能进行优化打磨。

4.5.5 交付要求

可运行的系统将提供为 Docker 镜像格式，镜像是名为 `musicstorm/prod:v1.0.0` 的完整微服务容器包，其中封装了 Web 前端（基于 React 框架）、后端 API 服务（使用 Node.js 构建）、AI 集成适配层（采用 Python 开发）以及必要的数据库初始化脚本。

为了满足本地测试环境的部署需求，系统同时提供独立部署包，它是一个 VMware OVA 格式的虚拟机模板，可以直接导入使用。

系统的源代码资产将以代码仓库的快照形式提供，文件命名为 `musicstorm-src-20250709.zip`。这个压缩包包含了使用 React 结合 Redux 状态管理开发的完整前端工程及其单元测试，基于 Node.js 实现的微服务后端（其 API 文档由 Swagger 工具自动生成），以及用于 AI 系统集成的 Python 适配层代码。所有源代码均附带一个名为 `LICENSE.md` 的开源协议声明文件，其中详细注明了项目依赖的第三方库及其许可信息。

数据库相关资产主要包括 PostgreSQL 数据库的表结构定义文件 `schema.sql`，该文件内含有对索引优化的注释说明，以及用于系统初始化的关键数据，如审核规则库和音乐标签分类表的 CSV 格式数据文件。

4.5.6 验收要求

核心创作流程确保创作者能够顺利完成从上传作品、AI 生成处理到提交审核并查看状态的全部步骤，要求该完整流程的成功率不低于 95%，该结论基于 10 组样本的测试验证结果；同时明确规定了必现 Bug 的判定标准，即在流程中出现元数据丢失、文件损坏或流程中断任意一种情况，均视为该流程未成功通过。

审核管理部分要求管理员具备高效处理能力，需在 1 分钟内完成对 5 条待审作品的处理操作，处理内容包含播放音频片段及完整填写驳回原因；一旦作品被驳回，系统必须确保相应的驳回信息在 10 秒内实时推送至创作者的平台通知中心。

社区交互功能重点关注响应速度和内容安全。在模拟千人同时在线的压力测试场景下，用户发布动态或评论的响应时间必须控制在 1 秒之内；系统配备的敏感词过滤机制要求具有高准确性，使用包含 500 多个违规词的专用词库进行验证时，其拦截准确率需达到 90% 以上（或表述为漏报率不超过 10%）。

AI 系统对其生成效率及质量有严格要求。该系统的音频生成任务，对于一段 30 秒的音频生成请求，其处理耗时不能超过 25 秒，且在连续发出 10 次请求的场景下，满

足此耗时的达标率需为 100%；在生成音频的质量方面，通过专业的音频分析软件（如 FFT）检测，必须确保生成结果无任何音频断裂或爆音现象，检测到的信号信噪比 (SNR) 需大于或等于 30 分贝。s

4.6 软件原型



图 6: 音乐社区

MusicStorm 提供了一系列核心功能来满足用户的听歌、探索和交流需求。“为我推荐”功能通过分析用户的个人偏好，专门生成个性化的歌单和歌曲推荐。“发现音乐”则是用户探索广阔音乐世界的入口，可以在此浏览新歌、热门榜单以及按不同类别归类的音乐内容。

应用内设有一个专门的“音乐社区”区域，作为用户互动交流的平台，支持音乐分享、发表评论以及其他社交功能。“最近播放”功能会自动记录并展示用户近期听过的歌曲列表，方便快速找回。“我喜欢的音乐”等同于用户的收藏夹，点击即可集中查看所有特别标记为喜爱的歌曲。

对于歌单管理，用户可以通过“创建的歌单”功能入口来制作并管理自己定制的歌单列表。而“收藏的歌单”则服务于保存和便捷访问其他用户创建、您感兴趣并保存下来的歌单内容。整体功能架构清晰，致力于提供全面且便捷的音乐体验。



图 7: 音乐社区后台审核

MusicStorm 的后台管理界面为管理员提供了一个清晰、高效的集中管控平台。整个界面采用简洁的灰白配色，布局分明。左侧设置了核心的功能导航栏，包含指向各个管理模块的入口：管理员可以通过“首页”返回主视图，“主面板”可能是核心数据仪表盘，“帖子管理”及细分出的“帖子审核”用于内容管理，“我的帖子”允许管理员查看自己发布的公告或内容状态，“信息管理”及其下的“个人信息”和“使用说明”则涵盖了账户设置和平台操作指南。

界面的主体区域集中在右侧，是综合的数据展示与操作区。最上方区域设计为关键数据概览卡片，直观地显示全局统计数据及其动态变化：包括当前总“帖子数量”以及自上次访问后显著增加的新帖、总评论数、总点赞数；以及系统消息状态这些增量数据是监控平台活跃度和内容增长的核心指标。

紧接着下方是“我的所有帖子详情”区，此处以表格形式详细列出了管理员关联帖子的信息，包含帖子的预览图（帖图）、名称（标题）和产生的收益（收益）。每个帖子条目旁应提供操作按钮（如“详情”）以进行管理。

右侧区域则主要用于可视化分析，展示了一个名为“发帖收益与内容长度”的图表（类型可能是折线图或柱状图，示例时间为“今日”），该图表旨在分析发布内容的长度（如文字或视频时长）与其所带来收益之间的关联趋势，为内容策略优化提供依据。

最底部设有“消息详情”区，用于展示具体的信息内容片段，例如图中示例显示了一条用户的评论：“我觉得周杰伦这首歌挺一般的，你这评价有些失真”。这一区域便于管理员快速查看和响应具体的用户反馈或系统通知。整体而言，该界面将导航、核心数据监控、内容管理、收益追踪、趋势分析和消息处理等功能集于一体，设计旨在提升平台管理的效率和洞察力。

5 软件设计规格说明

5.1 引言

5.1.1 编写目的

Music Storm 软件设计规格说明旨在为系统从需求到实现搭建清晰桥梁，为开发工作提供精准且全面的指导。首先，它将需求规格说明中的功能需求与限制条件转化为具体的系统架构、模块设计与技术方案，明确系统各组成部分的功能划分、交互逻辑和接口定义，帮助开发团队理解系统整体设计思路，避免因设计模糊导致的开发混乱与返工，确保开发工作高效推进。

其次，该文档对系统各模块的算法实现、数据结构设计等细节进行规范，为软件开发工程师编写代码提供统一标准，保证代码质量与可维护性。同时，通过详细说明系统的性能优化策略、安全防护设计等非功能需求的实现方式，确保最终产品在性能、安全性等方面达到预期目标。

此外，软件设计规格说明也是项目团队成员之间、团队与利益相关者之间沟通的重要工具。它使不同角色人员对系统设计达成共识，便于测试人员依据设计文档制定测试策略，管理人员进行项目进度把控与资源协调，从而促进项目各环节紧密协作，保障 Music Storm 系统高质量落地。

5.1.2 读者对象

本文档的读者对象主要包括系统架构师、软件开发工程师、测试工程师、项目管理人员以及其他利益相关者。系统架构师需要依据文档内容完善和优化系统架构设计，把控整体技术方向；软件开发工程师以文档中的模块设计、算法实现规范等为指导，开展代码编写工作；测试工程师通过阅读文档，了解系统设计细节，制定合理的测试策略与用例；项目管理人员则借助文档进行项目进度规划、资源协调和风险管控；确保系统设计符合业务需求与预期目标。

5.1.3 软件项目概述

A. 项目名称、简称和代号

项目全称为“Music Storm——基于 MindSpore 的音乐生成与鉴赏多模态社区系统”，简称为“Music Storm 系统”，项目代号可设定为“MS-Music”，方便在项目管理与沟通中快速识别。

B. 用户单位

用户单位面向广泛的音乐爱好者群体，包括但不限于专业音乐创作者、音乐研究人员、普通音乐发烧友等，同时也适用于音乐教育机构、文化娱乐企业等组织用户，满足其在音乐创作、教学、传播等方面的需求。

C. 开发单位

D. 功能需求

Music Storm 系统集成音乐生成、鉴赏与社交功能于一体。音乐生成支持文本、图像、音频等多模态输入，能快速生成古典、流行、电音等多种风格的音乐作品；音乐鉴赏模块配备智能推荐算法，可精准推送符合用户品味的音乐，并提供评论、分享等互动功能；社交功能构建活跃社区生态，用户可在其中分享创作心得、交流音乐见解，还可通过版权交易、会员服务等商业功能实现创作价值。

E. 性能需求

系统需具备高效的响应速度，确保音乐生成在普通配置设备上能在数秒内完成；智能推荐算法要快速准确，根据用户行为数据实时更新推荐结果；支持高并发访问，满足大量用户同时在线创作、鉴赏和社交的需求；同时，系统要保证数据安全与稳定，通过加密存储、定期备份等措施保护用户个人信息与音乐作品数据，在不同网络环境下均可稳定运行。

5.1.4 文档概述

大致内容：本文档聚焦将 Music Storm 系统需求转化为可落地的设计方案。开篇阐述编写目的与读者对象，明确文档价值与适用人群；随后深入解析系统架构设计，涵盖整体架构、模块划分，说明各模块如何协同实现音乐生成、鉴赏与社交功能。在详细设计部分，对各功能模块的算法逻辑、数据结构、接口设计等进行细化，如音乐生成模块的多模态数据处理算法、智能推荐算法的具体实现方式等。同时，针对性能优化、安全防护等非功能需求，提出对应的设计策略与技术方案，保障系统高效、稳定、安全运行。此外，还会包含对设计方案的评估与验证方法，确保设计满足需求。

组织结构：文档采用总分式结构，先总述编写目的、适用范围等基础信息，让读者快速把握文档核心。接着分模块详细阐述系统架构设计、各功能模块设计、非功能需求设计等内容，各部分逻辑清晰，层层递进。在系统架构设计之后，按照音乐生成、鉴赏、社交等功能模块依次展开详细设计，便于开发人员针对性查阅。最后以设计验证收尾，形成完整闭环，确保从整体架构到细节设计，再到最终验证的全流程覆盖，为系统开发提供全面、有序的指导。

5.1.5 定义

Music Storm 系统：指基于 MindSpore 的音乐生成与鉴赏多模态社区系统，是本软件项目开发的核心产品，集成音乐生成、鉴赏与社交功能，为用户提供多模态交互的音乐体验平台。

MindSpore：昇思 MindSpore，是一个全场景 AI 计算框架，具备高效计算性能与分布式训练能力，Music Storm 系统基于此框架实现音乐生成算法的开发与运行。

多模态：在本系统中表示结合文本、图像、音频等多种数据模态，用户可通过多种形式输入内容，系统据此生成相应音乐或提供交互服务，实现多样化的音乐创作与鉴赏

体验。系统架构设计：对 Music Storm 系统的整体结构、组成模块、模块间关系以及运行机制进行规划设计，明确系统各部分如何协同工作以实现音乐生成、鉴赏与社交等核心功能。

模块划分：将 Music Storm 系统按功能拆分为不同的子模块，如音乐生成模块、音乐鉴赏模块、社交模块等，每个模块负责特定功能，通过接口相互通信，共同完成系统目标。

算法逻辑：指各功能模块实现具体功能所采用的算法及运算流程，例如音乐生成模块中多模态数据处理算法如何将输入内容转化为音乐特征并生成音乐，智能推荐算法如何根据用户行为数据计算推荐结果。

数据结构：为高效存储、管理和处理数据，对系统中数据的组织形式、存储方式及操作方法进行设计，如用户信息、音乐作品数据、推荐算法相关数据等在系统中的存储与调用结构。

接口设计：定义系统各模块之间、系统与外部环境（如其他软件系统、硬件设备）之间进行数据交互和功能调用的规范与标准，确保不同部分能够准确、稳定地通信与协作。非功能需求：除系统核心功能外的其他需求，包括性能优化（如响应速度、并发处理能力）、安全防护（数据加密、用户认证、防攻击）、兼容性（不同操作系统、设备适配）等方面，用于保障系统的整体质量与用户体验。

设计验证：通过一定的方法和流程，对系统设计方案进行评估和测试，检查设计是否满足需求规格说明中的功能与性能要求，确保设计方案可有效指导系统开发并实现预期目标。

5.2 软件设计约束

5.2.1 软件设计目标和原则

A. 软件设计目标

功能实现目标：完成音乐生成、鉴赏与社交功能的设计与整合，确保音乐生成模块支持多模态输入，能够生成多样化风格音乐；音乐鉴赏模块实现精准智能推荐与流畅的互动体验；社交模块构建活跃的社区生态，支持用户间高效交流与合作。

性能优化目标：保障系统具备快速响应能力，音乐生成时间控制在合理范围内，满足高并发用户访问需求，确保在不同网络环境和硬件配置下，系统性能稳定，用户体验流畅。

兼容性目标：实现对 Windows、Mac、Linux 等主流操作系统，以及各类移动设备的良好适配，保证系统功能在不同平台上均可正常使用，降低用户使用门槛。

安全可靠目标：通过完善的数据加密、用户认证、权限管理等措施，保障用户数据和音乐作品安全；建立系统稳定性保障机制，确保系统 7×24 小时不间断运行，具备数据备份与灾难恢复能力。

B. 软件设计原则

技术选型原则：以 MindSpore 为核心技术框架，充分发挥其计算性能优势；选用成熟、稳定且社区活跃的开源技术与工具，降低开发风险，便于后期维护与功能扩展；确保技术栈与系统功能和性能需求高度匹配。

架构设计原则：采用模块化、分层化架构设计，提高系统的可维护性和可扩展性，各模块职责明确，通过标准化接口进行交互；遵循高内聚低耦合原则，减少模块间依赖，方便单个模块的升级与替换。

安全设计原则：将安全性融入系统设计的每个环节，遵循数据最小化收集、加密传输与存储原则；建立完善的安全防护体系，抵御网络攻击和数据泄露风险，确保系统符合相关数据安全法规要求。

用户体验原则：以用户需求为导向进行功能设计，界面简洁直观，操作流程便捷；注重交互细节，提供及时反馈，根据用户反馈持续优化系统功能和界面设计，提升用户满意度。成本控制原则：在满足系统需求的前提下，合理规划资源使用，避免过度设计；优先选择性价比高的技术方案和硬件配置，降低开发和运维成本，提高项目经济效益。

5.2.2 软件设计的约束和限制

硬件平台：服务器需具备高性能 CPU、GPU 和充足的内存、存储资源，以支持音乐生成模型的训练和实时运算；客户端需兼容主流计算机和移动设备。

操作系统：服务器端支持 Linux 等主流操作系统，客户端支持 Windows、Mac、Linux 及主流移动操作系统。

开发语言：主要采用 Python 语言进行开发，结合 MindSpore 框架的 API。

标准规范：遵循软件开发的相关标准和规范，如代码规范、接口规范等，确保系统的可维护性和可扩展性。

开发工具：使用 MindSpore 开发工具、IDE（如 PyCharm）、版本控制工具（如 Git）等。

容量和性能要求：随着用户量和数据量的增长，系统需具备良好的可扩展性，以满足存储和计算需求。

灵活性和配置要求：系统应具备灵活的配置选项，方便根据不同需求进行调整和优化。

5.3 软件设计

5.3.1 软件体系结构设计

表 8: 系统包说明

包名	说明	负责人
MUSIC.PLAYER.CORE	播放器主控模块，管理音频解码、播放控制和设备初始化	孙海洋
MUSIC.PLAYER.UI	用户界面管理模块，实现播放器界面、控制面板及可视化效果	夏清伟
MUSIC.COMMUNITY.FEED	社区动态模块，处理内容发布、信息流展示和用户互动	陈俊至
MUSIC.DATA.PLAYLIST	歌单管理模块，负责歌单创建、歌曲管理和个性化推荐	胡浩东
MUSIC.SERVICE.API	服务接口模块，对接后台 API 及第三方音乐平台服务	于畔湘

5.3.2 用户界面设计

A. 基础架构类

1. IView/BaseView: 定义视图渲染、更新和事件绑定的基础接口与实现
2. IViewModel/BaseViewModel: 管理视图数据和交互逻辑的基础接口与实现
3. Router: 负责页面路由导航，处理 URL 与视图的映射关系
4. Route: 单个路由配置项，关联路径、视图和视图模型

B. 核心视图类

1. AppView: 应用主视图，管理全局 Header 和 Footer
2. HomeView: 首页视图，展示推荐音乐和热门内容
3. MusicPlayerView: 音乐播放视图，包含播放控制和歌词展示
4. SongListView: 歌单详情视图，展示歌单内容和信息
5. UserProfileView: 用户个人主页视图，展示个人信息和收藏
6. DiscoverView: 发现音乐视图，按风格和新发布分类展示

C. 视图模型类

各视图对应 ViewModel (如 HomeViewModel): 处理数据加载和业务逻辑 MusicPlayerViewModel: 管理播放状态、歌曲信息和播放控制

D. 数据模型类

User: 用户信息模型 Song: 歌曲信息模型, 包含歌手、专辑关联 Playlist: 歌单模型, 包含歌曲列表和创建者信息 Artist/Album/Genre: 音乐相关实体模型

5.3.3 用例设计

A. 播放歌曲

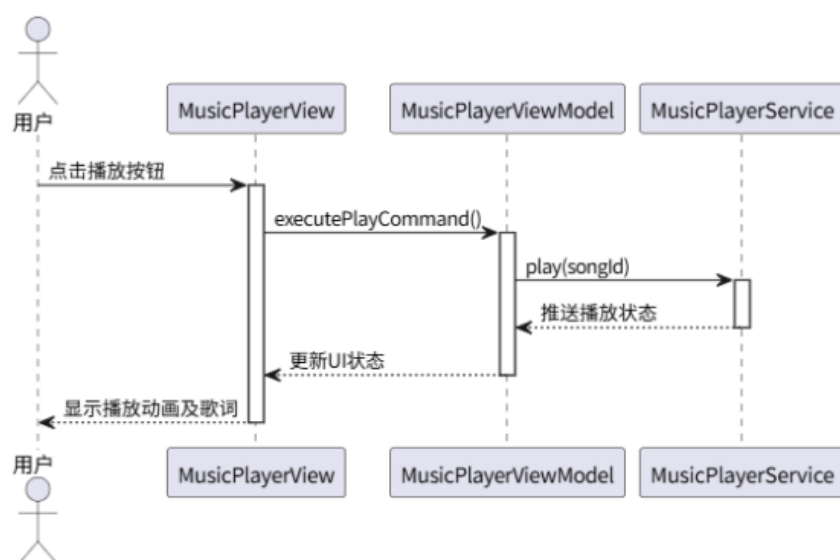


图 8: 播放歌曲-顺序图

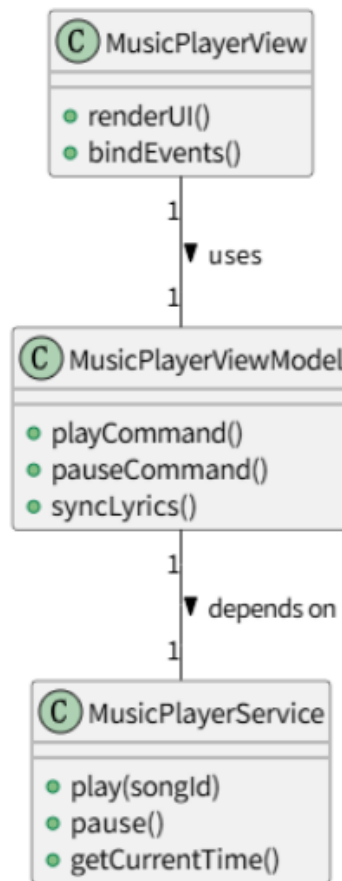


图 9: 播放歌曲-类图

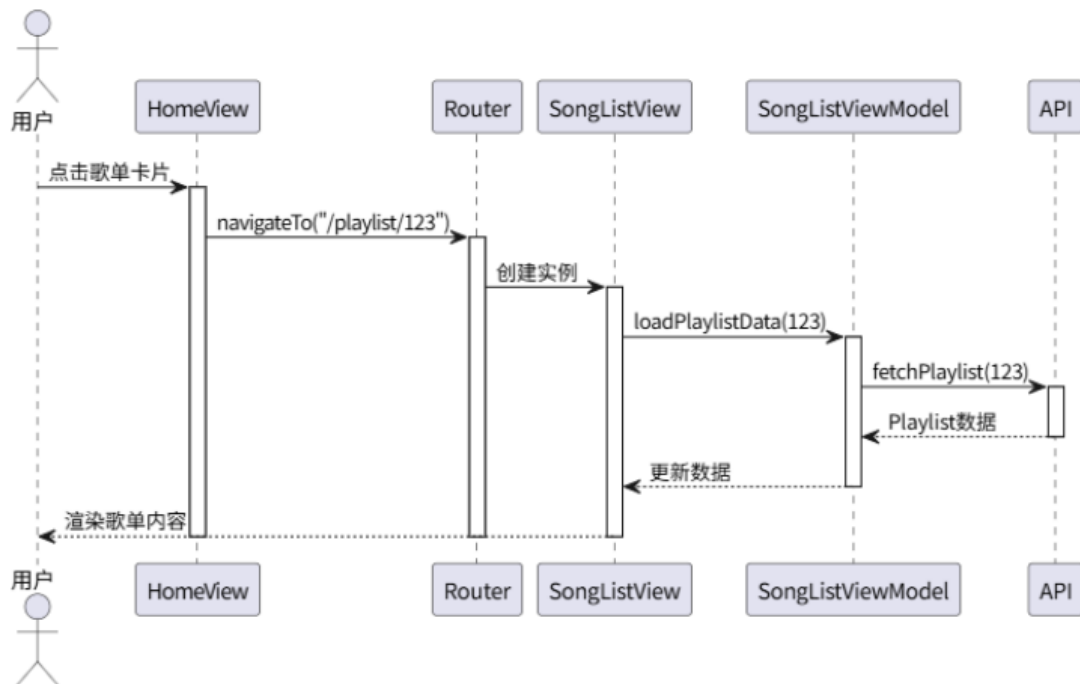


图 10: 歌单浏览-顺序图

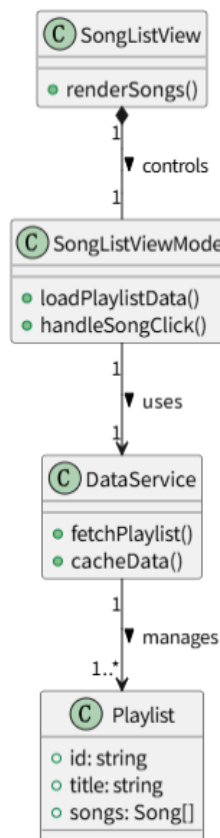


图 11: 歌单浏览-类图

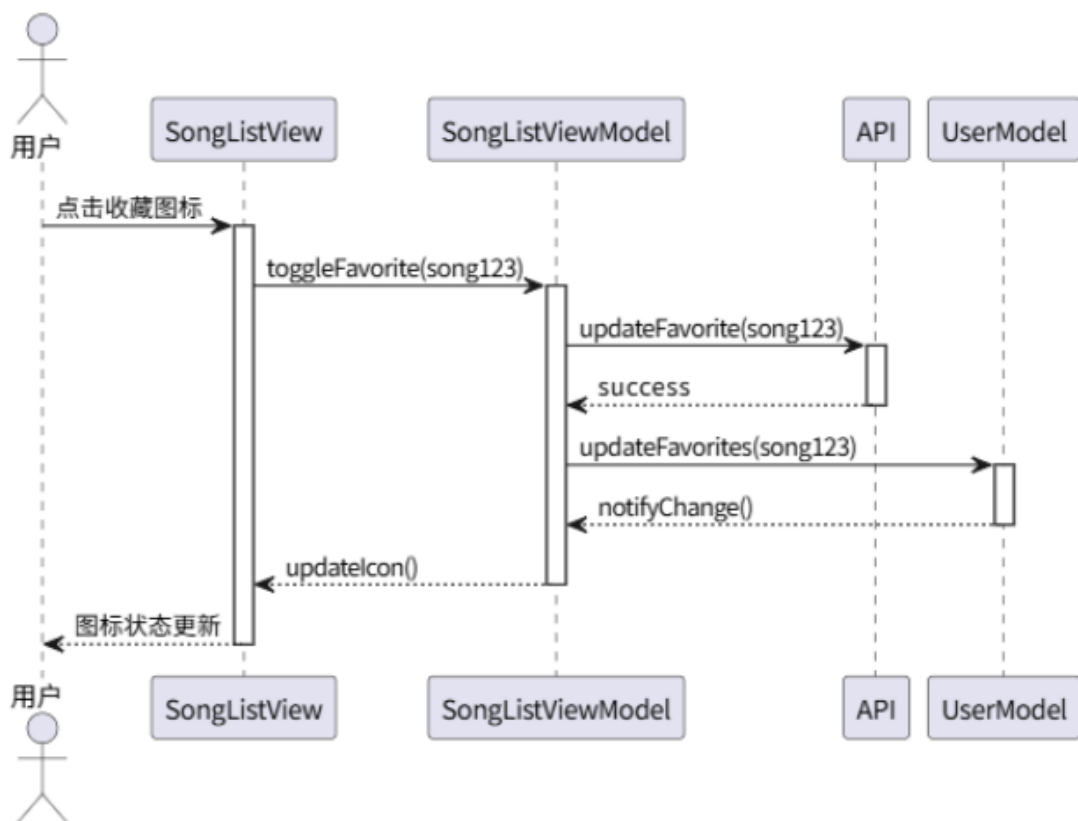


图 12: 歌曲收藏-顺序图

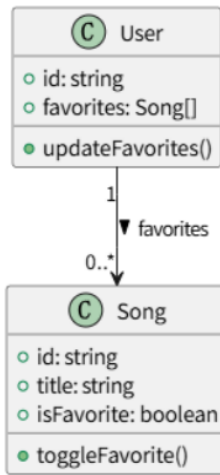


图 13: 歌曲收藏-类图

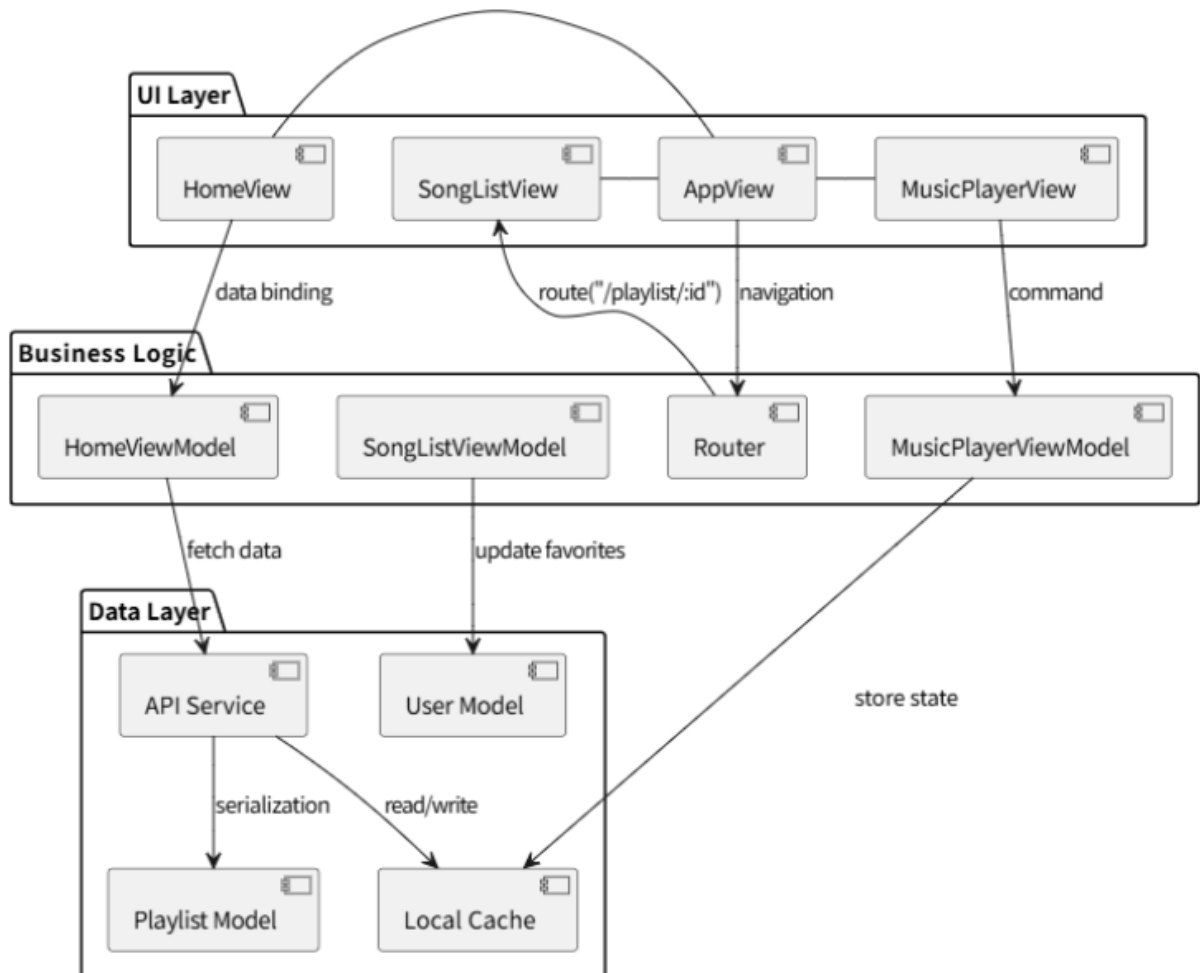


图 14: 歌曲收藏-类图

5.3.4 类设计

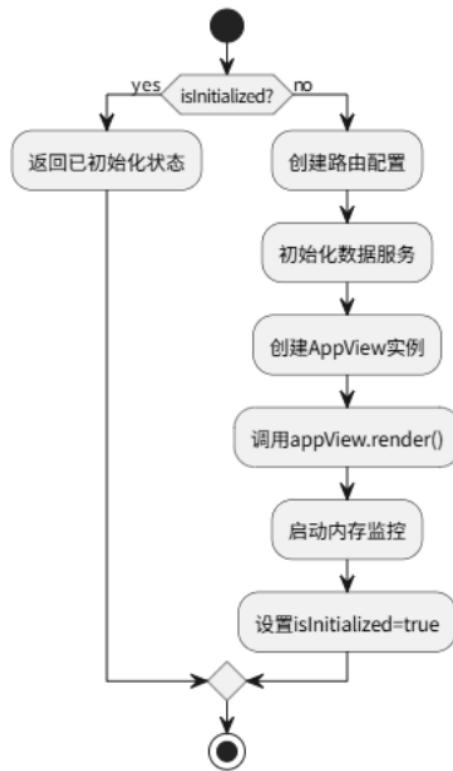


图 15: 歌曲收藏-类图

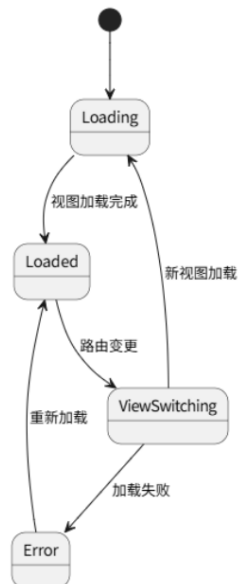


图 16: 歌曲收藏-类图

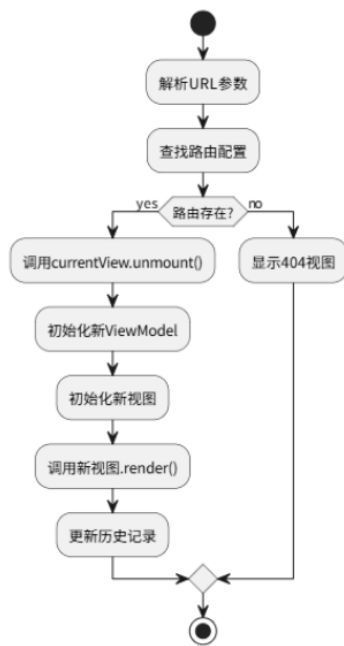


图 17: 歌曲收藏-类图

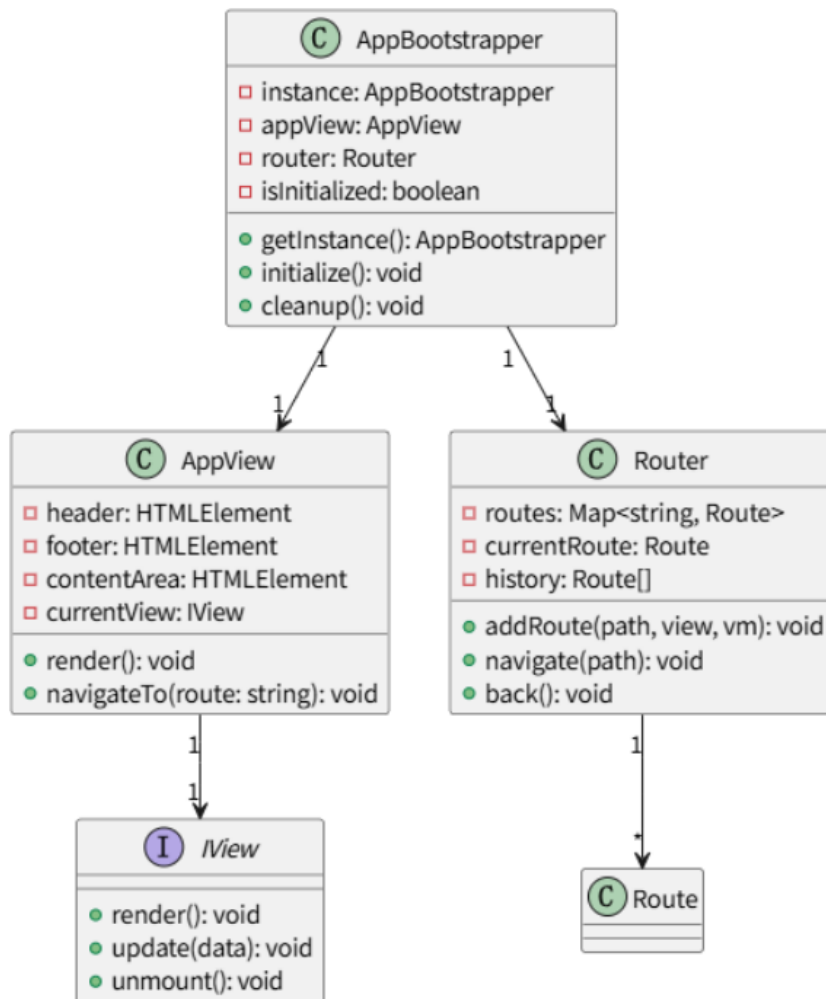


图 18: 歌曲收藏-类图

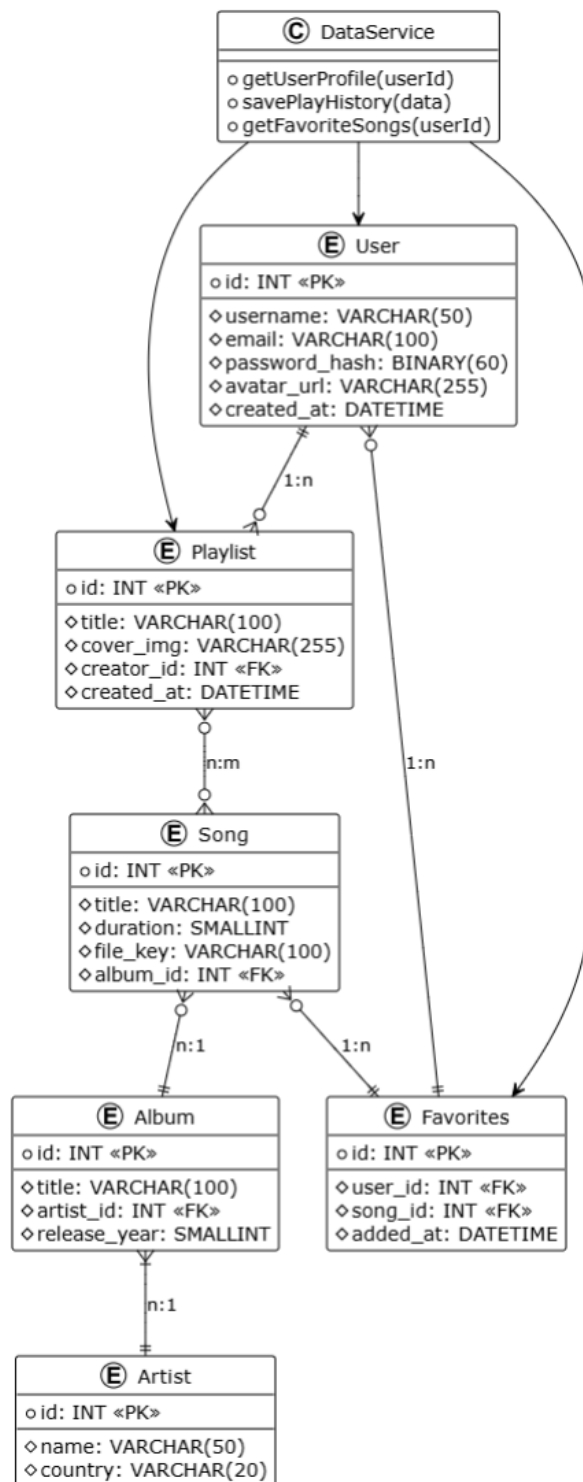
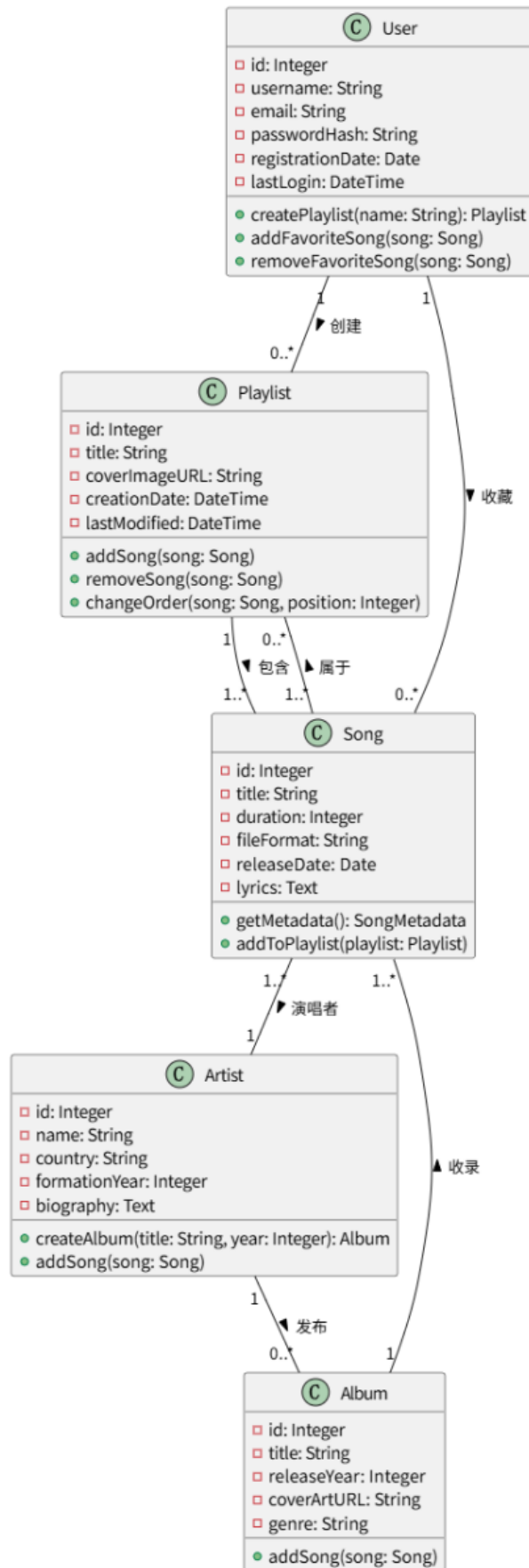


图 19: 歌曲收藏-类图

5.3.5 数据设计

A. 概念结构设计 (类图)



B. 逻辑结构设计

表 9: 逻辑结构设计

关系表名	字段	类型	约束
users	user_id	INT	PK, AUTO_INC
	username	VARCHAR(50)	NOT NULL, UNIQUE
	email	VARCHAR(100)	NOT NULL, UNIQUE
	password_hash	CHAR(60)	NOT NULL
	registration_date	DATETIME	DEFAULT NOW()
	last_login	DATETIME	
artists	artist_id	INT	PK, AUTO_INC
	name	VARCHAR(50)	NOT NULL
	country	VARCHAR(30)	
	formation_year	YEAR	
	biography	TEXT	
albums	album_id	INT	PK, AUTO_INC
	title	VARCHAR(100)	NOT NULL
	release_year	YEAR	
	cover_art_url	VARCHAR(255)	
	genre	VARCHAR(30)	
	artist_id	INT	FK (artists.artist_id)
songs	song_id	INT	PK, AUTO_INC
	title	VARCHAR(100)	NOT NULL
	duration	SMALLINT	
	file_key	VARCHAR(100)	NOT NULL
	release_date	DATE	
	lyrics	TEXT	
	album_id	INT	FK (albums.album_id)
playlists	playlist_id	INT	PK, AUTO_INC
	title	VARCHAR(100)	NOT NULL
	cover_image	VARCHAR(255)	
	created_at	DATETIME	DEFAULT NOW()
	last_modified	DATETIME	
	user_id	INT	FK (users.user_id)
song_artist	song_id	INT	PK, FK (songs.song_id)
	artist_id	INT	PK, FK (artists.artist_id)
	is_primary	BOOLEAN	DEFAULT TRUE
user_favorites	user_id	INT	PK, FK (users.user_id)

表 9: 逻辑结构设计

关系表名	字段	类型	约束
playlist_songs	song_id	INT	PK, FK (songs.song_id)
	added_at	DATETIME	DEFAULT NOW()
	playlist_id	INT	PK, FK (playlists.playlist_id)
	song_id	INT	PK, FK (songs.song_id)
	position	SMALLINT	

C. 物理结构设计

表 10: 用户信息表表

序号	列名	数据类型	长度	说明	允许为空
1	uUserId	文本	10	用户唯一编号，主键（如 UUID 简化格式），用于身份识别与关联业务数据。	否
2	uUserName	文本	20	用户名，登录及展示用，需符合平台命名规范。	否
3	uPassword	文本	15	密码（加密存储，如 MD5/BCrypt 哈希后），保障账户安全。	否
4	uEmail	文本	10	用户邮箱，用于找回密码、系统通知。	是
5	uPhone	文本	20	手机号码，支持快捷登录（需验证。	是
6	uAge	文本	整数	年龄，0 - 100 有效范围，用于个性化推荐（如音乐风格偏好关联。	是
7	uRole	货币	文本	用户角色（普通用户 / 创作者 / 管理员等），控制功能权限。	否

表 11: 音乐作者关系表

序号	列名	数据类型	长度	说明	允许为空
1	mWorkID	文本	12	音乐作品唯一编号，主键，关联生成记录、鉴赏数据。	否
2	mTitle	文本	50	作品标题，展示及检索用。	否
3	mStyle	文本	20	音乐风格（流行 / 古典 / 电子等），辅助推荐算法。	是

表 11: 音乐作者关系表

序号	列名	数据类型	长度	说明	允许为空
4	mGenerator	文本	10	创作者 ID (关联 tb_user.uUserId), 明确作品归属。	否
5	mInputType	文本	20	生成输入类型 (文本 / 图像 / 多模态等), 用于分析创作习惯。	是
6	mFilePath	文本	255	音乐文件存储路径 (服务器 / 云存储地址), 用于播放、下载。	否
7	mCreateTime	日期时间	默认	作品生成时间, 记录创作节点。	否

表 12: 音乐帖子关系表

序号	列名	数据类型	长度	说明	允许为空
1	aAnalysisID	文本	15	鉴赏记录唯一编号, 主键。	否
2	aWorkID	文本	12	关联音乐作品 ID (tb_music_work.mWorkID), 绑定分析对象。	否
3	aFeatureData	文本	200	音乐特征数据 (如旋律复杂度、节奏 BPM 等), JSON 格式存储, 用于推荐算法。	是
4	aRecommendTag	文本	50	推荐标签 (如“治愈系”“高能量”等), 辅助社区内容分发。	是
5	aAnalysisTime	日期时间	默认	鉴赏分析时间, 记录系统处理节点。	否

表 13: 用户交互关系表

序号	列名	数据类型	长度	说明	允许为空
1	iInteractID	文本	15	交互记录唯一编号, 主键。	否
2	iUserId	文本	10	操作用户 ID (关联 tb_user.uUserId), 明确交互主体。	否
3	iWorkID	文本	12	关联音乐作品 ID (tb_music_work.mWorkID), 若无则为社区动态 (如纯文字评论)。	是
4	iType	文本	10	交互类型 (评论 / 点赞 / 收藏 / 私信等), 区分业务逻辑。	否

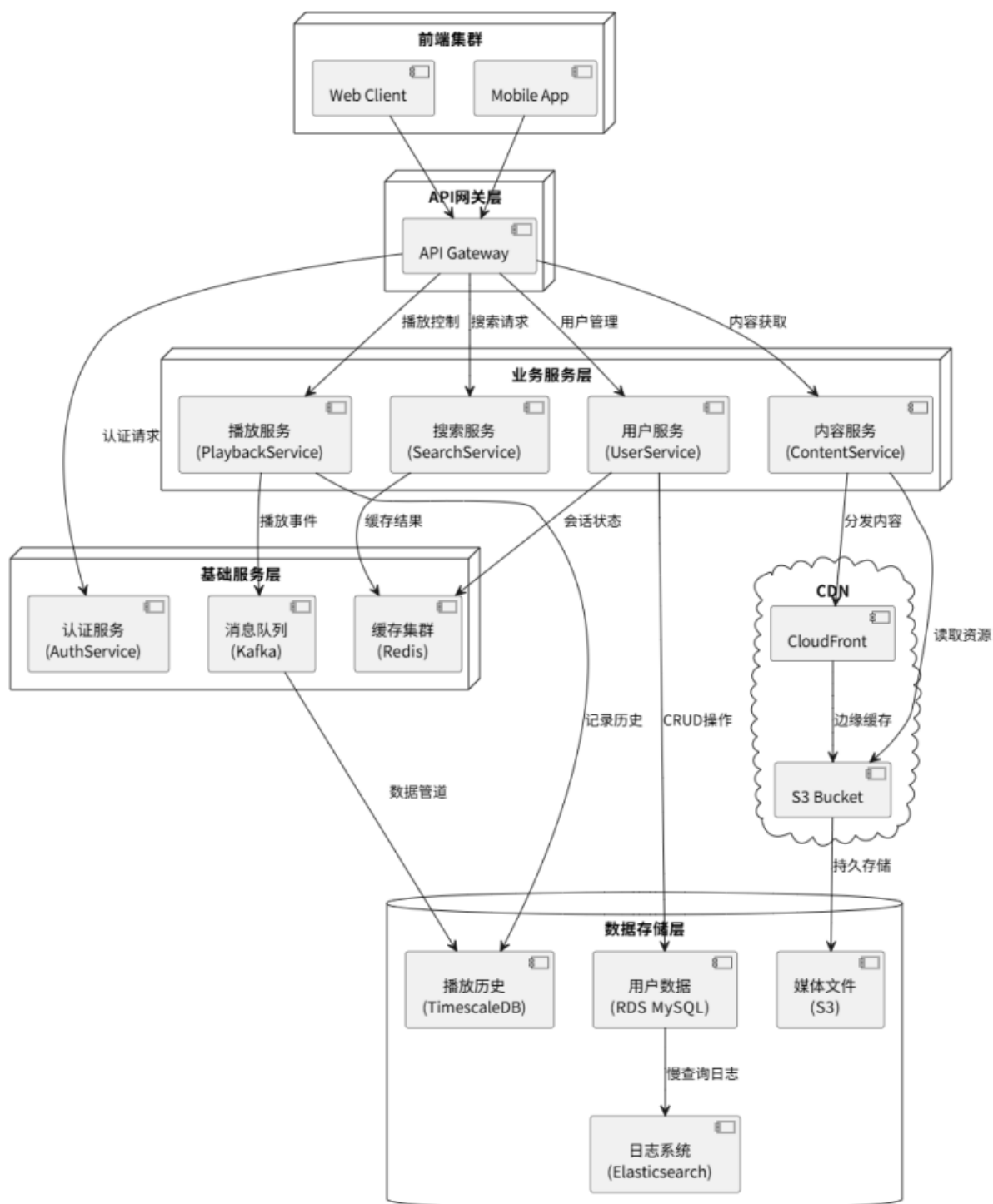
表 13: 用户交互关系表

序号	列名	数据类型	长度	说明	允许为空
5	iContent	文本	200	交互内容（评论文字、私信消息等），需过滤敏感词。	是
6	iCreateTime	日期时间	默认	交互发生时间，记录社区活跃节点。	否

表 14: 配置关系表

序号	列名	数据类型	长度	说明	允许为空
1	cConfigID	文本	10	配置项唯一编号，主键。	否
2	cConfigKey	文本	50	配置键	否
3	cConfigValue	文本	200	配置值	是
4	cDescription	文本	100	配置说明，解释配置项作用（如“音乐生成超时时间，单位秒”）。	是

5.3.6 部署设计



5.3.7 系统出错处理设计

错误名城	解决方法
音乐生成相关错误	
多模态输入不完整 (Incomplete multimodal input)	提示用户补充缺失的输入类型 (如文本、图像、音频至少选其一)，返回生成界面重新操作。Prompt the user to supplement the missing input type (e.g., at least one of text, image, or audio must be selected), and return to the generation interface to re - operate.
生成参数超出范围 (Generation parameters out of range)	高亮显示超出范围的参数 (如风格权重、节奏复杂度等)，提示合理取值范围，引导重新设置。Highlight the parameters that are out of range (such as style weight, rhythm complexity, etc.), prompt the reasonable value range, and guide to reset.
音乐生成超时 (Music generation timeout)	提示生成耗时过长，提供“取消重试”“继续等待”选项；若频繁超时，建议检查设备性能或联系技术支持。Prompt that the generation takes too long, provide options of "Cancel and Retry" and "Continue Waiting"; if frequent timeouts occur, suggest checking device performance or contacting technical support.

音乐鉴赏与推荐错误	
音乐特征解析失败 (Music feature analysis failure)	提示音乐解析异常，建议更换音乐文件或稍后重试；记录错误日志便于后台排查。Prompt that the music analysis is abnormal, suggest replacing the music file or retrying later; record the error log for background troubleshooting.
推荐算法无结果 (Recommendation algorithm no results)	显示“暂无匹配推荐”，引导用户调整推荐条件 (如扩大风格范围、刷新推荐)，或推荐热门通用内容。Display "No matching recommendations for now", guide the user to adjust the recommendation conditions (such as expanding the style range, refreshing the recommendation), or recommend popular general content.
社区交互错误	
作品发布内容违规 (Illegal content in work release)	拦截违规内容 (涉黄、侵权等)，提示违规类型并告知平台规则，拒绝发布；可申诉反馈。Intercept illegal content (pornographic, infringing, etc.), prompt the type of violation and inform the platform rules, and refuse to publish; appeal feedback is available.
评论 / 私信含敏感词 (Comments/private messages contain sensitive words)	屏蔽敏感内容，提示用户修改后重新提交；多次违规触发账号警告。Block sensitive content, prompt the user to modify and resubmit; multiple violations trigger an account warning.
关注 / 点赞操作失败 (Follow/like operation failure)	提示操作异常，检查网络或重试；若因权限问题 (如被对方拉黑)，明确告知权限限制。Prompt for abnormal operation, check the

	network or retry; if due to permission issues (such as being blocked by the other party), clearly inform the permission restriction.
--	--

用户管理与权限错误	
身份认证失效 (Identity authentication failure)	强制跳转登录界面，提示“登录过期 / 身份验证失败”，重新完成认证。Force jump to the login interface, prompt "Login expired/Identity authentication failed", and complete the authentication again.
权限不足访问功能 (Insufficient permissions to access function)	提示“当前账号无此功能权限”，引导升级会员 / 申请对应权限，或展示权限说明页面。Prompt "The current account has no permission for this function", guide to upgrade membership/apply for corresponding permissions, or display the permission description page.
个人信息修改格式错误 (Personal information modification format error)	(如昵称含特殊字符、年龄超范围、联系方式格式错等)，逐项高亮错误字段，提示正确格式要求，重新编辑。(e.g., nickname contains special characters, age is out of range, contact information format is wrong, etc.), highlight the error fields one by one, prompt the correct format requirements, and re - edit.
系统与数据错误	
数据库连接失败 (Database connection failure)	提示“系统数据加载异常，请稍后重试”，后台自动尝试重连；频繁失败触发运维报警。Prompt "System data loading exception, please try again later", the background automatically attempts to reconnect; frequent failures trigger operation and maintenance alarms.
文件上传 / 下载失败 (File upload/download failure)	显示失败原因（网络中断、存储不足、文件损坏等），支持断点续传或重新操作，记录失败日志。Display the failure reason (network interruption, insufficient storage, file corruption, etc.), support resumable upload or re - operation, and record the failure log.
跨设备同步冲突 (Cross - device synchronization conflict)	提示“多设备操作有冲突”，展示冲突内容差异，让用户选择保留哪端数据或合并处理（如作品修改记录）。Prompt "There is a conflict in multi - device operation", display the difference in conflicting content, and let the user choose which end of the data to keep or merge (such as work modification records).

6 系统测试

6.1

6.2

6.3

6.4

6.5

6.6

六、23 1. 引言 23 2. 测试概要 23 3. 测试结果及发现 23 3.1 测试 1（标识符）23 3.2 测试 2（标识符）23 4. 对软件功能的结论 23 4.1 功能 1（标识符）23 4.2 功能 2（标识符）24 5. 分析摘要 24 5.1 能力 24 5.2 缺陷和限制 24 5.3 建议 24 5.4 评价 24 6. 测试资源消耗 24 七、系统用户操作手册 25 1. 引言 25 1.1 编写目的 25 1.2 适用范围 25 1.3 版权声明 25 2. 安装与卸载 25 3. 使用说明 25 3.1 xx 功能 25 八、总结与展望 27 参考文献 28 附录（可选，项目实施部分）团队总结及评分 1

参考文献

- [1] P. Domingos, “A few useful things to know about machine learning,” *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012, doi: 10.1145/2347736.2347755.
- [2] Q. Wang, “Analysis of the incidence and influencing factors of frailty among the elderly in Chinese communities,” *Journal of Southern Medical University*, vol. 41, no. 11, pp. 1719–1724, Nov. 2021, doi: 10.12122/j.issn.1673-4254.2021.11.18.
- [3] S. V. Wilson, B. Ceber, J. Myatt, and S. Wilson, AnotherSamWilson/miceforest: Release for Zenodo DOI. (Dec. 12, 2022). Zenodo. doi: 10.5281/zenodo.7428632.
- [4] S. Wen, “Calculation of Care Needs and Care Costs of Disabled Elderly in China - Based on 2013 CHARLS National Baseline Survey”.
- [5] “CHARLS database.” Accessed: Apr. 04, 2024. [Online].
- [6] N. Shrestha, “Detecting Multicollinearity in Regression Analysis,” *American Journal of Applied Mathematics and Statistics*, vol. 8, no. 2, Art. no. 2, Jun. 2020, doi: 10.12691/ajams-8-2-1.
- [7] He N., “Detection of cognitive impairment among the elderly in China and its relationship with sleep duration - an empirical analysis based on the 2018 CHARLS data,” *Chinese Journal of Gerontology*, doi: 10.3969/j.issn.1005-9202.2023.07.056.

- [8] Z. Arvanitakis, R. C. Shah, and D. A. Bennett, “Diagnosis and Management of Dementia: Review,” *JAMA*, vol. 322, no. 16, pp. 1589–1599, Oct. 2019, doi: 10.1001/jama.2019.4782.
- [9] J. Lee Ray, “Explaining Interstate Conflict and War: What Should Be Controlled for?,” *Conflict Management and Peace Science*, vol. 20, no. 2, pp. 1–31, Sep. 2003, doi: 10.1177/073889420302000201.
- [10] X. Zhao et al., “Longitudinal Relationship Between Frailty and Cognitive Impairment in Chinese Older Adults: A Prospective Study,” *J Appl Gerontol*, vol. 41, no. 12, pp. 2490–2498, Dec. 2022, doi: 10.1177/07334648221118352.
- [11] R. Guo, “Master ’s thesis on prediction of preeclampsia based on SAM-Voting,” 2022, doi: 10.27009/d.cnki.gdblu.2022.001069.
- [12] F. E. Harrell, *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. in Springer Series in Statistics. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-19425-7.
- [13] X. Zhou, “Research on the demand forecast of China ’s elderly care workers based on Markov model”.
- [14] Wang tao and Guo Z., “Research progress in diagnosis and treatment of mild cognitive impairment,” *Western Medicine*, vol. 31, no. 9, 2019.