

**移动应用软件开发综合实践报告**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指导老师意见   |  |  | | --- | --- | | 演 示 答 辩 （50%）：□优 □良 □中 □及格 □差 | 简评 | | 面向对象分析（10%）：□优 □良 □中 □及格 □差 |  | | 面向对象设计（10%）：□优 □良 □中 □及格 □差 | | 面向对象实现（10%）：□优 □良 □中 □及格 □差 | | 面向对象测试（10%）：□优 □良 □中 □及格 □差 | | 文 字 格 式 （10%）：□优 □良 □中 □及格 □差 |   课题等级: 签名: 日期: |

题 目： 仿网易云app

学生姓名： 刘志铭 、 李睿奇 、 唐伟铭

学 号： 202114060406、202114060416 、 202114060412

专业班级： 软工21104班

指导教师： 罗赞、彭易波

设计时间： 2024年上学期第15—17周 Git地址：<https://github.com/twmtest/fangwangyiyun.git>

评分明细表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课题名称 | 仿网易云app | | | 总分 |  |
| 技术栈：  ArkUI、arkTS  主要功能：  用户登录，主题更换，音乐播放，歌词查看，动态发放和查看，点赞。  歌单查看。 | | | | | |
| 姓名 | 刘志铭 | 学号 | 202114060406 | 得分 |  |
| 主要工作：歌曲播放界面，歌词查看，音乐切换。报告编写。 | | | | | |
| 姓名 | 唐伟铭 | 学号 | 202114060412 | 得分 |  |
| 主要工作：用户登录，后端接口编写，请求封装，接口调用。 | | | | | |
| 姓名 | 李睿奇 | 学号 | 202114060416 | 得分 |  |
| 主要工作：查看歌单，主题切换，动态评论查看。PPT制作。 | | | | | |

目录

[第一章 绪论 6](#_Toc17366)

[1. 课题背景及意义 6](#_Toc24853)

[2. 国内外研究现状 6](#_Toc29267)

[3. 相关技术和工具 7](#_Toc15517)

[4. 本文的工作 7](#_Toc30947)

[第二章 需求识别 8](#_Toc29457)

[1. 需求陈述 8](#_Toc13880)

[2. 产品分析 8](#_Toc23459)

[3. 用户分析： 9](#_Toc21235)

[3.1目标用户群体 9](#_Toc22971)

[3.2 用户需求分析 10](#_Toc21735)

[3.3 用户行为分析 13](#_Toc29642)

[3.4 用户痛点分析 15](#_Toc26908)

[4.结论 16](#_Toc18855)

[4. 产品方向 17](#_Toc823)

[4.1高质量音质和音乐资源 17](#_Toc12002)

[4.2 个性化推荐和智能播放 17](#_Toc9472)

[4.3 多设备协同和跨平台同步 18](#_Toc4279)

[4.4智能设备支持： 18](#_Toc30479)

[5. 业务流程 18](#_Toc19586)

[5.1 用户登录 18](#_Toc25577)

[5.2 首页推荐和音乐发现 19](#_Toc737)

[5.3 音乐播放和控制 19](#_Toc12573)

[第三章 分析设计 20](#_Toc5358)

[1. 系统框架 20](#_Toc11801)

[2. 产品结构 21](#_Toc9266)

[2.1 功能结构 21](#_Toc3787)

[2.2 信息结构 22](#_Toc29502)

[2.3 页面结构 23](#_Toc30811)

[3. 分析建模 25](#_Toc13331)

[3.1对象模型 25](#_Toc14447)

[3.2 功能模型 31](#_Toc31421)

[4. 原型设计 32](#_Toc9182)

[4.1登录界面 32](#_Toc32204)

[4.2歌单界面 33](#_Toc16322)

[4.3播放界面 33](#_Toc23440)

[4.4 当播放列表界面 34](#_Toc12355)

[4.5 歌词界面 34](#_Toc4678)

[4.6 动态界面 35](#_Toc660)

[4.7 评论界面 35](#_Toc2712)

[4.8 我的界面 36](#_Toc21493)

[4.9设置界面 36](#_Toc20245)

[第四章 系统实现 38](#_Toc9662)

[1. 后端实现 38](#_Toc8345)

[2. 移动端实现 38](#_Toc28345)

[2.1移动端架构 38](#_Toc22197)

[2.2 工程目录 38](#_Toc15447)

[3. 模块实现 41](#_Toc13467)

[3.1.登录模块 41](#_Toc2694)

[3.2. 用户信息模块 42](#_Toc24084)

[3.3.音乐播放模块 43](#_Toc28459)

[3.4设置模块 44](#_Toc13712)

[3.5动态模块 45](#_Toc24214)

[第五章 系统测试 46](#_Toc29521)

[1. 测试环境 46](#_Toc14440)

[2. 功能测试 46](#_Toc1848)

[2.1系统功能测试 46](#_Toc2237)

[3. 非功能测试 47](#_Toc8644)

[3.1界面测试 47](#_Toc1947)

[4. 测试结论 48](#_Toc26178)

[总结 49](#_Toc28922)

[参考文献 52](#_Toc5950)

# 绪论

## 课题背景及意义

随着互联网和移动设备的普及，传统的音乐购买和下载方式被流媒体音乐服务所取代。流媒体音乐服务通过互联网提供音乐播放，用户可以通过流媒体音乐服务下载音乐，从而实现音乐的无缝衔接。互联网云音乐运营平台是中国领先的音乐运营平台，依托互联网+音乐运营服务，提升用户满意度和市场竞争力。流媒体音乐平台的开发和普及，推动音乐产业从传统模式向数字化转型，促进音乐的广泛传播和产业发展。

## 国内外研究现状

随着互联网技术的飞速发展，音乐流媒体平台的质量问题迅速普及，成为用户获取和分享音乐的主要方式。国内以网络云音乐为代表的流媒体平台，通过精准的个性化定制和丰富的用户社交功能，构建了独特的音乐社区氛围，取得了极佳的反响，受到了用户的广泛好评。国内的流媒体平台提供商Spotify、Apple Music等均通过先进的算法和广泛的音乐库，为用户提供了优质的音乐体验。特别在流媒体方面进行了深入的研究，利用数据分析和大数据，为用户提供了优质的音乐体验。国内的流媒体平台提供商Spotify对音乐流媒体平台进行了深入的研究，主要目的是优化用户行为、分析音乐情感、提升音乐用户体验。

## 相关技术和工具

在使用DevEco Studio工具开发仿网易云音乐应用时，开发者可以依托华为提供的强大集成开发环境，该环境集成了开发、调试和测试等多种功能，大幅简化了开发流程。DevEco Studio支持HarmonyOS的开发框架和界面设计语言，开发者可以直观地设计和实现跨设备平台的用户界面，确保应用在不同设备上的一致性和良好的用户体验。开发者可以使用Java或Kotlin等语言，利用DevEco Studio提供的丰富的开发工具和模板，快速搭建应用框架和实现业务逻辑。此外，DevEco Studio还集成了调试工具和模拟器，帮助开发者在开发过程中及时发现和解决问题，提高开发效率。通过DevEco Studio，开发者能够充分利用HarmonyOS的分布式能力和安全框架，实现音乐数据的管理、用户交互的优化以及应用性能的提升。综上所述，利用DevEco Studio工具开发仿网易云音乐应用，不仅能够简化开发流程，还能够确保应用的稳定性和用户体验的一致性，为开发者带来便利和效益。

## 本文的工作

本论文从系统设计、数据库架构、算法实现、前端交互设计、实时通信以及性能与安全性优化等多个方面，详细描述了一个仿网易云音乐应用系统的完整实现过程。我们希望通过这些工作，能够为读者提供一个全面的系统开发案例参考。

# 需求识别

## 需求陈述

系统具体需要实现的功能如下：

登录：系统用户只能为网易云用户，通过已有的账户通过扫描二维码进行登录，成功登录后可进行其他功能使用。

音乐播放：通过登录后能够在收藏的歌单列表中选择歌曲进行播放，通过点击需要播放的歌曲来进行播放。点击暂停会停止播放，能够切换其他歌曲，当点击播放下一首时会播放列表中此歌曲的下一首歌曲。播放上一首时会播放歌单中此歌曲的上一首歌曲。

歌词查看：能够点击查看歌词来查看当前所播放的歌曲的歌词，查看其他歌曲的歌词时需要先切换到需要查看歌词的歌曲。

播放方式选择：在用户需要长时间播放时可以先选择播放方式，循环播放使能够循环播放正在播放的歌曲。随机播放，在歌单列表中随机选择一首歌曲在当前歌曲播放结束后进行播放。顺序播放，当前歌曲播放结束后，选择歌单列表中的下一首进行播放。

主题更换：通过主题选择使界面进行切换。

动态查看：能够查看用户所关注的博主发放的动态

## 产品分析

产品名称：仿网易云播放器

产品简介：仿网易云音乐播放器是一款为音乐爱好者设计的多功能音乐播放应用，旨在提供丰富的音乐资源、个性化的音乐推荐、高清的音质体验，以及跨设备的无缝播放体验。基于 HarmonyOS 开发，该播放器集成了网易云音乐的核心功能和用户界面设计，确保用户享受到高质量的音乐服务。

## 用户分析：

### 3.1目标用户群体

3.1.1年轻音乐爱好者

年龄范围：16-35 岁

特征：热爱音乐，喜欢探索新歌和小众音乐，有较强的社交需求，愿意分享和互动。

需求：高质量音质、多样化的音乐资源、个性化推荐、便捷的社交功能。

3.1.2专业音乐从业者

年龄范围：25-45 岁

特征：音乐制作人、DJ、音乐评论家等，对音质和音频格式有高要求，关注音乐行业动态。

需求：无损音质、高清音频下载、专业播放和管理功能、音乐资讯和评论。

3.1.3普通音乐用户

年龄范围：18-50 岁

特征：日常听音乐的习惯，使用音乐播放器来放松、健身或背景音乐。

需求：简洁易用的界面、稳定的播放性能、多种播放模式（随机、循环等）、歌词显示。

### 3.2 用户需求分析

#### 3.2.1 高质量音质

用户期望在使用音乐播放器时，能够享受到高保真、无损音质的音乐体验。这不仅包括普通音乐爱好者，更包括对音质有更高要求的专业用户，如音乐制作人、音响发烧友等。他们需要：

高分辨率音频格式：支持FLAC、ALAC等无损音频格式，保证音乐细节不丢失。

高动态范围：提供广泛的动态范围，使得音乐的强弱变化更加明显和自然。

低延迟和高稳定性：确保播放时音质的稳定输出，不会因为网络或设备问题而影响听觉体验。

#### 3.2.2 多样化的音乐资源

用户希望音乐播放器能够提供一个丰富、多样化的音乐库，涵盖广泛的音乐类型和风格，以满足不同用户的喜好和需求，包括但不限于：

流行音乐：最新的流行单曲和专辑，涵盖不同国家和地区的流行音乐。

古典音乐：从巴洛克到现代的经典曲目，提供详细的曲目分类和作曲家介绍。

摇滚音乐：涵盖经典摇滚、硬摇滚、重金属等不同风格。

电子音乐：包括EDM、Techno、House等，适合电子音乐爱好者。

世界音乐：提供来自世界各地的音乐，如非洲鼓乐、印度古典等。

#### 3.2.3 个性化推荐

用户希望通过音乐播放器发现更多自己喜欢的音乐，个性化推荐可以根据用户的听歌历史和偏好，智能地推荐合适的歌曲和歌单。具体需求包括：

智能推荐算法：基于机器学习和用户行为分析，精准捕捉用户喜好，推荐适合的音乐。

用户定制歌单：自动生成用户专属的歌单，包含可能喜欢的新歌和老歌。

场景推荐：根据用户的日常活动（如运动、学习、休闲），推荐相应风格的音乐。

#### 3.2.4 便捷的播放列表管理

用户需要一个方便、灵活的工具来创建、编辑和管理他们的播放列表。具体需求包括：

播放列表创建：快速创建新播放列表，支持多种排序方式（如按添加时间、歌曲名、歌手等）。

播放列表编辑：支持添加、删除、重排序、重命名歌曲，便于用户个性化管理。

导入导出功能：支持从外部导入播放列表（如M3U文件），或导出到其他设备或应用。

#### 3.2.5 实时歌词和专辑封面

用户希望在听歌时能够看到同步显示的歌词和高清专辑封面，这能大大增强他们的视觉和听觉体验。具体需求包括：

同步歌词显示：自动匹配并显示与当前播放歌曲同步的歌词，支持卡拉OK模式。

高清专辑封面：展示高质量的专辑封面图片，提供专辑和艺术家详细信息。

自定义背景：允许用户自定义播放界面的背景图，如使用专辑封面或自己的照片。

#### 3.2.6 社交互动

用户希望能够在音乐播放器中与其他音乐爱好者互动，具体需求包括：

歌曲评论：能够对歌曲进行评论，查看其他用户的评论和评分。

音乐分享：支持将自己喜欢的歌曲、歌单分享到社交媒体或直接发送给好友。

关注好友动态：查看和跟踪好友的听歌动态、推荐内容和创建的歌单。

#### 3.2.7 跨设备同步

用户希望能够在不同的设备之间无缝切换音乐播放，包括手机、平板、智能音箱、车载系统等。具体需求包括：

多设备同步：在一个设备上暂停音乐，另一个设备可以无缝继续播放。

同步播放列表：确保所有设备上的播放列表和音乐库一致，自动同步更新。

多设备控制：允许在一个设备上控制其他设备的播放，如在手机上控制智能音箱。

### 3.3 用户行为分析

#### 3.3.1 听歌习惯

大多数用户都有固定的听歌习惯，尤其是在特定场合或时间段。具体行为包括：

日常听歌：用户通常每天都会在通勤、工作、学习和休闲时段听音乐。

探索新歌：年轻用户尤其喜欢探索新音乐和不同类型的音乐，他们更容易接受新的音乐风格和艺术家。

长时间听歌：很多用户会长时间听音乐，特别是在背景音乐的情况下，比如学习和工作时。

#### 3.3.2 使用场景

用户在各种场景下使用音乐播放器，具体包括：

独自聆听：用户在独自时会使用播放器聆听音乐，如在家里、办公室或旅途中。

聚会娱乐：在聚会或社交活动中，用户会使用音乐播放器播放背景音乐或舞曲。

运动健身：用户在健身房或户外运动时，喜欢听音乐来保持动力。

学习专注：用户在学习时使用音乐播放器播放轻音乐或背景音乐，以帮助提高专注力。

#### 3.3.3 社交行为

用户喜欢在社交媒体上分享音乐，具体表现为：

分享歌曲和歌单：用户会将自己喜欢的歌曲和歌单分享给朋友或发布在社交媒体上。

评论互动：通过评论和点赞，与其他音乐爱好者交流，分享对音乐的看法和感受。

参与音乐社区：加入音乐相关的在线社区，参与讨论和分享，获取音乐推荐和信息。

#### 3.3.4 设备使用

用户在多种设备上使用音乐播放器，具体表现为：

智能手机：是最主要的音乐播放设备，用户喜欢在手机上随时随地听音乐。

智能音箱：在家中使用智能音箱播放音乐，享受更好的音质和便捷的语音控制。

车载系统：在驾驶时，用户希望使用车载系统播放音乐，提供娱乐和导航信息。

### 3.4 用户痛点分析

#### 3.4.1 音质不佳

部分音乐播放器的音质无法满足用户需求，具体问题包括：

音质压缩：很多播放器对音质进行了过度压缩，导致音质不佳。

无损音质缺乏：对于需要高保真音质的用户，许多播放器不支持无损音质的音乐格式。

音频延迟：网络或设备问题导致音频播放有延迟，影响听觉体验。

#### 3.4.2 推荐不精准

个性化推荐算法不够精准，导致用户不能有效地发现喜欢的音乐，具体问题包括：

推荐歌曲无关：推荐的歌曲与用户喜好无关，用户需要手动筛选大量不相关的音乐。

重复推荐：经常推荐用户已经听过的歌曲，缺乏新鲜感。

算法不更新：推荐算法更新不及时，无法捕捉用户最新的音乐偏好。

#### 3.4.3 操作复杂

一些音乐播放器的界面和功能设计过于复杂，具体表现为：

界面混乱：界面设计不清晰，用户很难找到需要的功能和设置。

操作步骤繁琐：执行简单的操作需要经过多个步骤，用户体验不佳。

缺乏指导：没有提供足够的用户指导和提示，新用户难以上手。

#### 3.4.4 广告干扰

免费版本中广告过多，严重影响听歌体验，具体问题包括：

频繁插入广告：播放过程中频繁插入广告，打断用户的听歌体验。

广告时长过长：广告时间较长，用户不得不等待较长时间才能继续听音乐。

无法跳过广告：强制观看广告，用户无法跳过或关闭。

#### 3.4.5 跨设备同步不便

部分播放器的跨设备同步功能不够完善，具体问题包括：

同步延迟：设备之间的同步存在延迟，导致音乐播放无法无缝切换。

同步失败：有时同步功能会失败，导致播放列表和音乐库无法更新。

配置复杂：跨设备同步需要复杂的配置和设置，用户体验差。

### 4.结论

通过详细的用户分析，我们可以更好地理解目标用户的需求和行为，针对性地优化仿网易云音乐播放器的功能和设计，提高用户满意度和使用体验。具体措施包括提升音质、优化个性化推荐算法、简化操作界面、减少广告干扰、完善跨设备同步功能等。

## 产品方向

### 4.1高质量音质和音乐资源

目标：提升音质，扩充音乐资源库，满足用户对音乐品质和多样性的需求。

措施：

支持无损音质：提供无损音质选项，满足对音质有高要求的用户需求。

扩充音乐库：增加版权合作，扩充音乐资源，涵盖流行、古典、摇滚、电子等多种音乐类型。

高清音频下载：允许用户下载高清音频文件，实现离线无损播放。

### 4.2 个性化推荐和智能播放

目标：通过大数据和人工智能技术，提供精准的个性化推荐，提高用户的音乐发现体验。

措施：

个性化推荐算法：基于用户的听歌历史、偏好和行为，优化推荐算法，提供个性化歌曲和歌单推荐。

智能播放模式：开发智能播放模式，根据用户的听歌习惯和时间地点自动切换播放内容。

动态歌单：创建动态歌单，根据用户心情、天气等因素动态调整播放内容。

### 4.3 多设备协同和跨平台同步

目标：实现多设备间的无缝播放和同步，提升用户在不同场景下的音乐体验。

措施：

多设备同步：支持用户在手机、平板、智能音箱、车载系统等多设备之间无缝切换播放，保持播放进度一致。

云端音乐库：用户的音乐收藏、播放列表和听歌历史都可云端同步，保证跨平台一致性。

### 4.4智能设备支持：

优化对智能音箱、可穿戴设备等智能设备的支持，实现更便捷的控制和操作。

## 业务流程

### 5.1 用户登录

用户在登录界面扫描二维码

通过在网易云账户授权

登录成功后，用户信息和状态保存到本地和服务器。

**核心功能：**

用户身份验证

数据库存储和管理

用户信息加密和保护

### 5.2 首页推荐和音乐发现

**首页推荐：**

用户登录后进入首页，显示个性化推荐内容。

系统根据用户的历史播放记录、偏好、时间和地点等因素推荐音乐。

### 5.3 音乐播放和控制

**音乐播放：**

用户点击播放按钮，应用发送请求到服务器获取音乐文件。

服务器返回音乐文件，播放器解码并播放音乐。

播放器界面显示音乐进度、歌词和专辑封面。

**播放控制：**

用户可以暂停、继续、跳转、切换歌曲。

用户可以调整音量、选择播放模式（单曲循环、随机播放等）。

**核心功能：**

音乐文件请求和播放

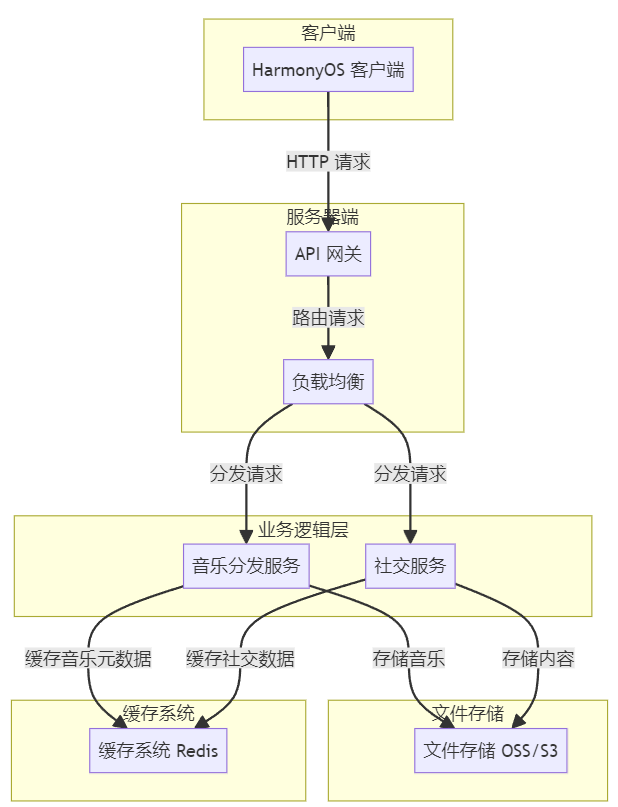
播放器控制接口

音乐进度和歌词同步

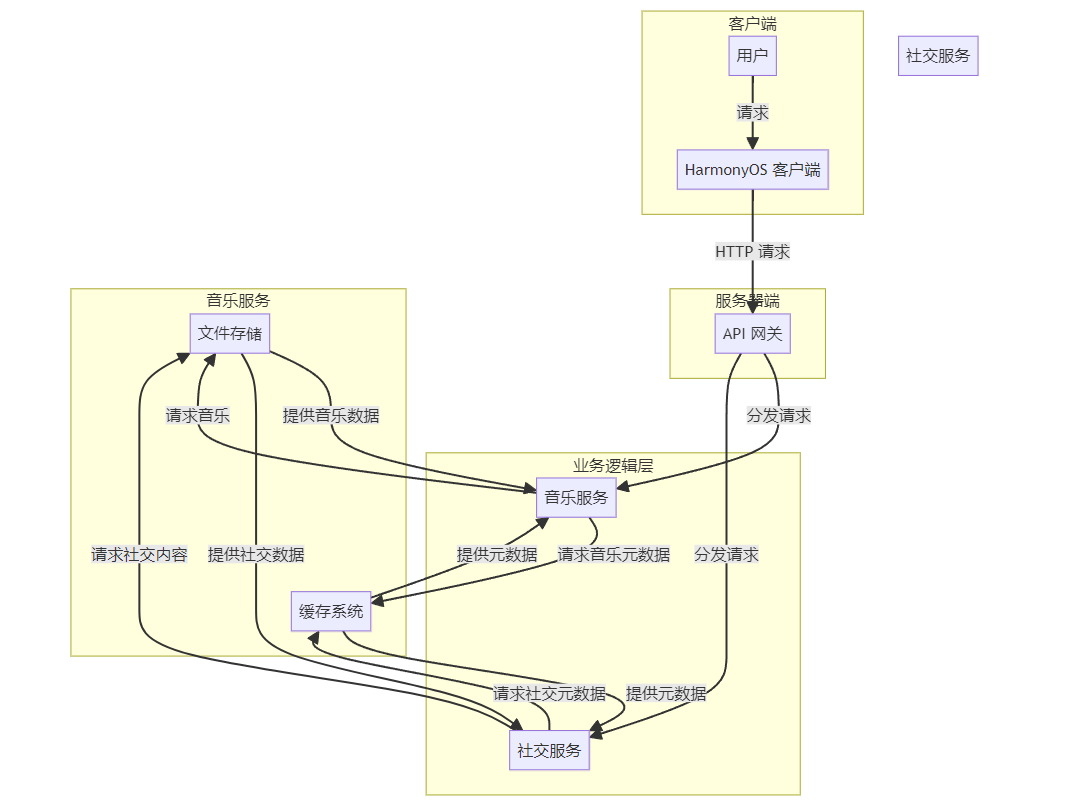
# 分析设计

## 系统框架

仿网云音乐系统的架构设计包括从前端到管理系统的多个模块和组件，涵盖了用户管理、音乐推荐、播放管理、社交互动等核心功能。通过合理的技术选型和架构设计，系统能够保证高可用性和高性能，提供丰富的功能和良好的用户体验。在实际开发过程中，建议根据业务需求不断优化和调整架构，确保系统能够满足市场需求。



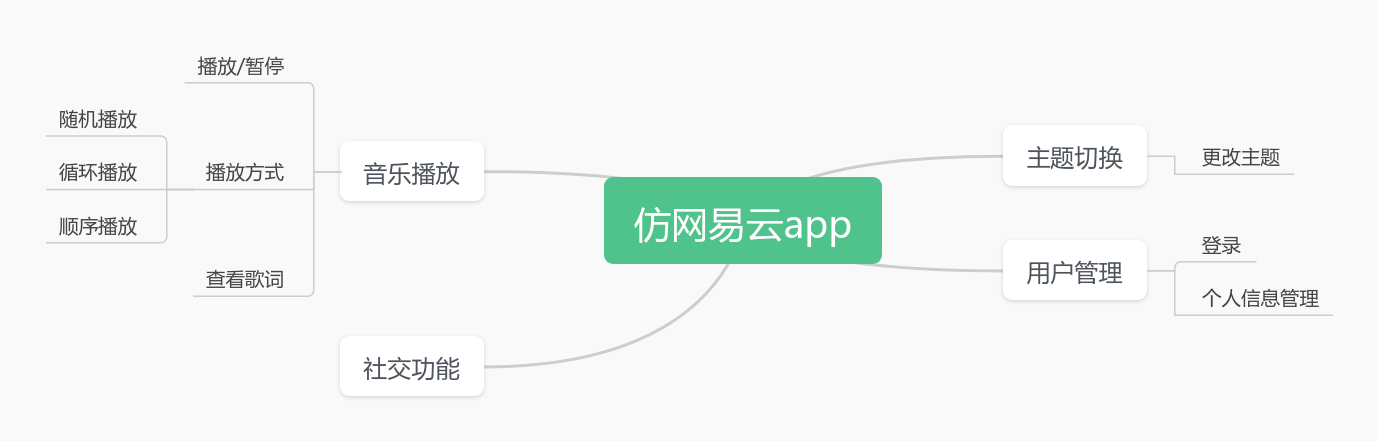
仿网易云app系统架构图



仿网易云app数据流图

## 产品结构

### 功能结构



仿网易云app结构图

仿网易云音乐应用：是整个应用的顶层节点，下面连接各主要功能模块。

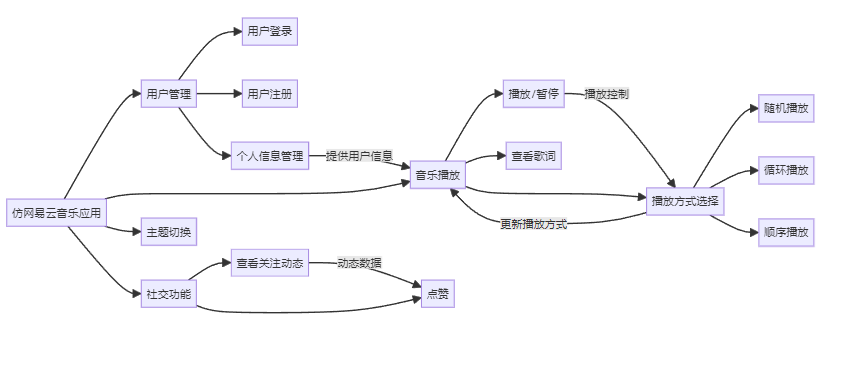
主题切换：独立的功能模块，用户可以更改应用的主题，提升用户体验。

用户管理：涉及用户身份验证和个人信息管理。所有用户相关操作集中在这个模块。

音乐播放：是应用的核心功能模块。用户可以播放音乐、查看歌词，并选择播放方式。三种播放方式（随机、循环、顺序）可以提升用户的听歌体验。

社交功能：用户可以查看关注的博主动态并进行互动，如点赞。这些功能提高了应用的用户粘性和互动性。

### 信息结构



仿网易云app信息结构图

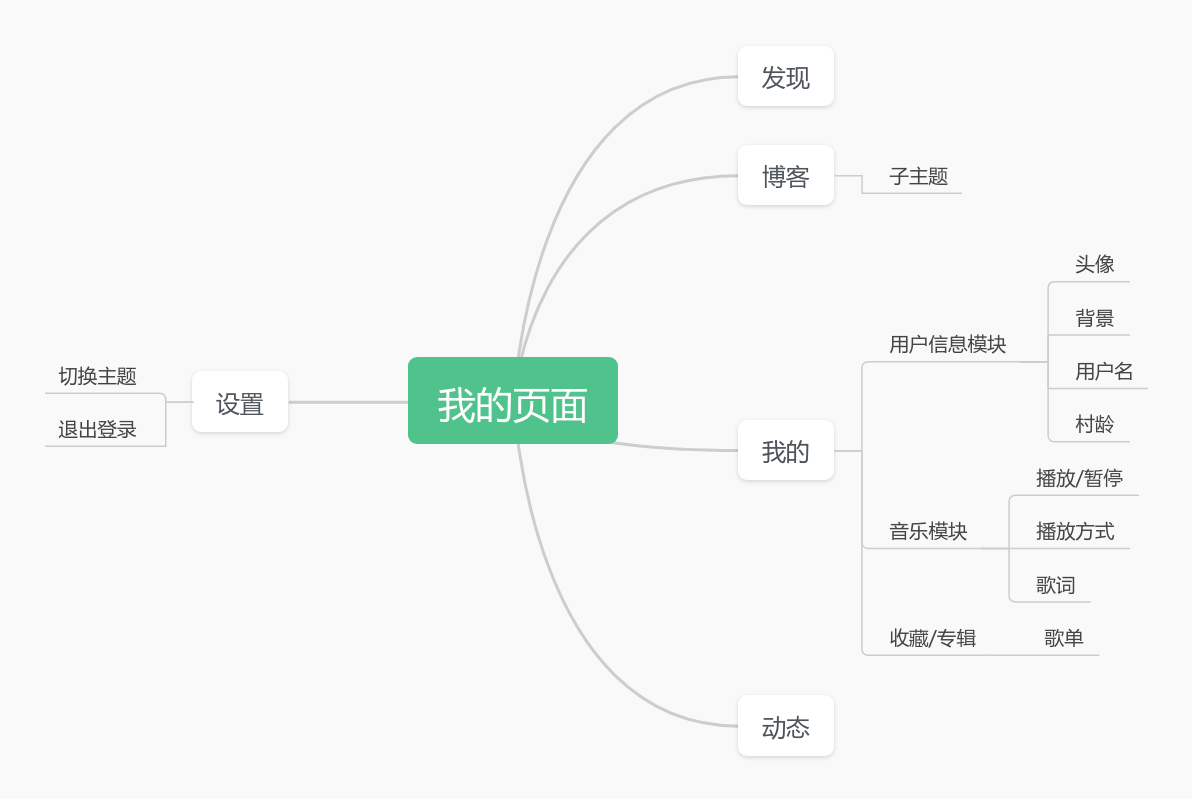
用户管理：用户通过登录和注册提供信息，这些信息存储在个人信息管理模块中。

音乐播放：个人信息管理模块提供用户偏好和设置，影响音乐播放模块的播放行为。

主题切换：用户可以通过主题切换功能改变应用的外观，这会影响其他模块的显示效果。

社交功能：用户可以查看动态并点赞，数据流动在查看关注动态和点赞之间，用户的互动行为会反馈到社交模块中。

### 2.3 页面结构



页面结构图

我的页面：

中心节点，代表整个“我的页面”的入口。

从该页面可以导航到不同的子页面和功能模块。

发现：

功能模块，与“我的页面”直接相连。

该模块的功能主要用于探索新内容，比如推荐的歌曲、新发布的专辑、热门歌单等。

博客：

功能模块，与“我的页面”直接相连。

子主题：表示该模块下的具体内容，比如用户发布的博客文章、动态分享、评论互动等。

我的：

功能模块，与“我的页面”直接相连。

包含了多个子模块，详细描述如下：

头像：显示用户的头像，可以点击进行修改或查看大图。

背景：用户的个人主页背景图片，也可以进行更改。

用户名：显示当前登录用户的用户名。

村龄：表示用户在该平台的注册时长或会员等级。

音乐模块：包含了与音乐播放相关的功能，详细描述如下：

播放/暂停：控制音乐的播放和暂停。

播放方式：提供三种播放模式，分别是随机播放、循环播放和顺序播放。

歌词：显示当前播放歌曲的歌词。

收藏/专辑：用户可以查看已收藏的歌曲和专辑。

歌单：用户创建和管理自己的歌单，方便音乐的分类和播放。

动态：

功能模块，与“我的页面”直接相连。

用户可以在此查看朋友或关注的博主的动态，如新发布的歌曲、分享的心情、活动参与情况等。

设置：

功能模块，与“我的页面”直接相连。

包含两大功能：

切换主题：用户可以在应用中更改不同的主题风格，改变应用界面的外观。

退出登录：用户可以在此退出当前账户，返回到登录页面。

直观层次：页面结构以“我的页面”为中心，功能模块按照用户的需求和操作习惯分布，清晰地展示了应用的主要功能。

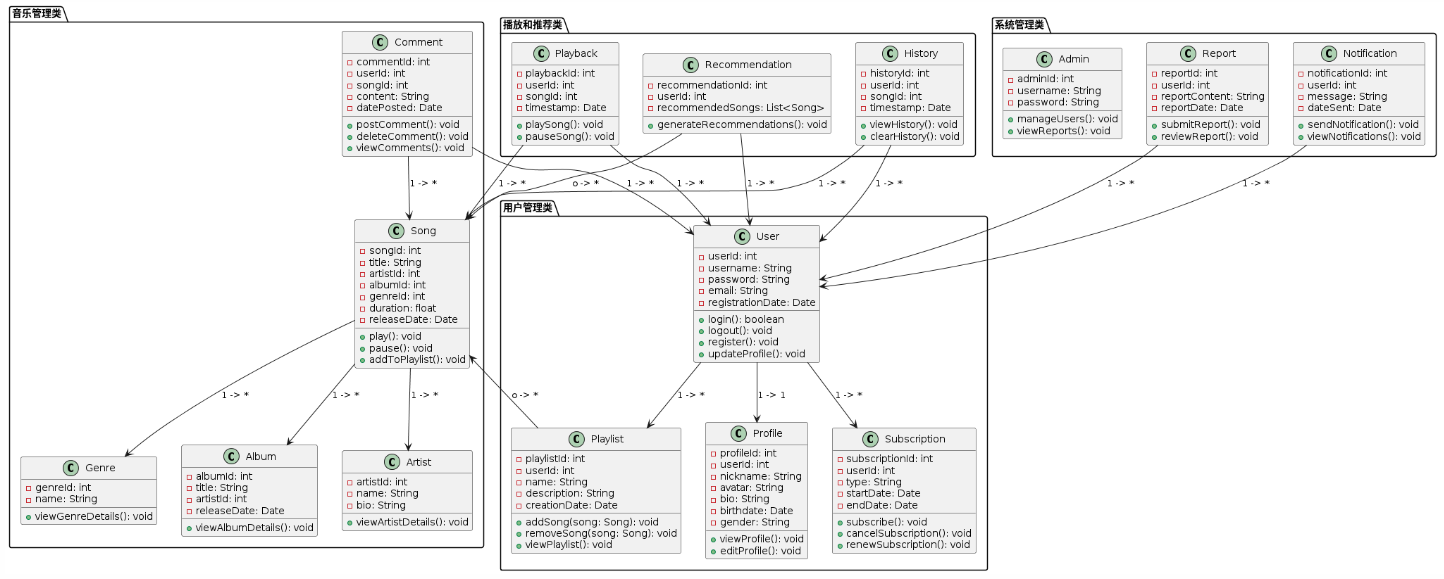
用户体验：通过这种结构，用户可以方便地访问不同的功能模块，了解和管理自己的音乐播放、动态和个性化设置。

功能整合：

各功能模块紧密关联，使用户可以在一个页面中完成音乐播放、动态查看和个性化设置的多个操作。

## 分析建模

### 3.1对象模型



在仿网易云音乐应用系统中，类图的每个类都承担了特定的职责，确保系统的各个部分能高效协同工作。以下是对每个类的详细说明及其作用：

用户管理类

user

作用：管理用户的基本信息和操作。包括用户的注册、登录、登出等基本账户操作。

主要功能：

处理用户的认证（登录/登出）。

管理用户的账户信息（如用户名、密码、邮箱）。

维护与用户相关的其他信息（如用户订阅、歌单等）。

存储和管理用户的详细个人资料信息。负责用户的公共信息展示和编辑。

显示和编辑用户的个人信息（如昵称、头像、简介）。

提供用户个人资料的查看功能。

Subscription

作用：管理用户的订阅状态和相关信息。处理订阅的创建、续订和取消。

主要功能：

管理用户订阅类型（如免费或付费）。

处理订阅周期的开始和结束。

提供订阅的创建、取消和续订功能。

Playlist

作用：管理用户创建的歌单。提供添加、删除和查看歌单中的歌曲功能。

主要功能：

用户创建和管理自己的歌单。

允许用户在歌单中添加和移除歌曲。

提供歌单的查看和管理接口。

音乐管理类

Song

作用：代表系统中的一首歌曲，包含歌曲的基本信息。负责与歌曲播放相关的操作。

主要功能：

存储歌曲的基本信息（如标题、艺术家、专辑、类型）。

处理歌曲的播放和暂停。

提供将歌曲添加到歌单的功能。

Album

作用：管理和显示专辑信息。提供专辑的详细信息查看功能。

主要功能：

存储专辑信息（如专辑标题、艺术家、发布日期）。

提供查看专辑详情的功能。

Artist

作用：管理和展示艺术家的信息。提供艺术家详细信息的查看功能。

主要功能：

存储艺术家的基本信息（如姓名、简介）。

提供查看艺术家详情的功能。

Genre

作用：管理音乐类型信息。用于分类和展示不同类型的音乐。

主要功能：

存储音乐类型信息（如类型名称）。

提供查看音乐类型详情的功能。

Comment

作用：处理用户对歌曲的评论。管理评论的创建、删除和查看。

主要功能：

存储评论内容和关联的用户、歌曲信息。

提供评论的发布、删除和查看功能。

播放和推荐类

Playback

作用：管理用户的播放记录。记录用户播放的每首歌曲的详细信息。

主要功能：

存储播放记录的详细信息（如播放时间、用户、歌曲）。

提供播放和暂停歌曲的功能。

Recommendation

作用：根据用户的听歌历史和偏好生成推荐歌曲。提供个性化的音乐推荐。

主要功能：

存储推荐记录和推荐的歌曲列表。

生成和展示推荐列表。

History

作用：记录用户的播放历史。管理用户的播放记录的查看和清除。

主要功能：

存储用户播放历史记录。

提供查看和清除播放历史的功能。

系统管理类

Admin

作用：管理系统的管理员用户。负责系统的管理和维护工作。

主要功能：

管理普通用户和系统的相关设置。

查看和处理用户报告。

Report

作用：处理用户的报告和反馈。管理报告的提交和审核。

主要功能：

存储用户的报告内容和提交日期。

提供报告的提交和审核功能。

Notification

作用：管理和发送系统通知。负责通知用户系统中的重要事件和信息。

主要功能：

存储通知内容和发送时间。

提供发送和查看通知的功能。

技术实现类

APIController

作用：处理系统的API请求。作为系统与外部通信的接口，处理各类数据请求。

主要功能：

存储API端点和请求类型。

处理API请求并返回响应。

DatabaseConnection

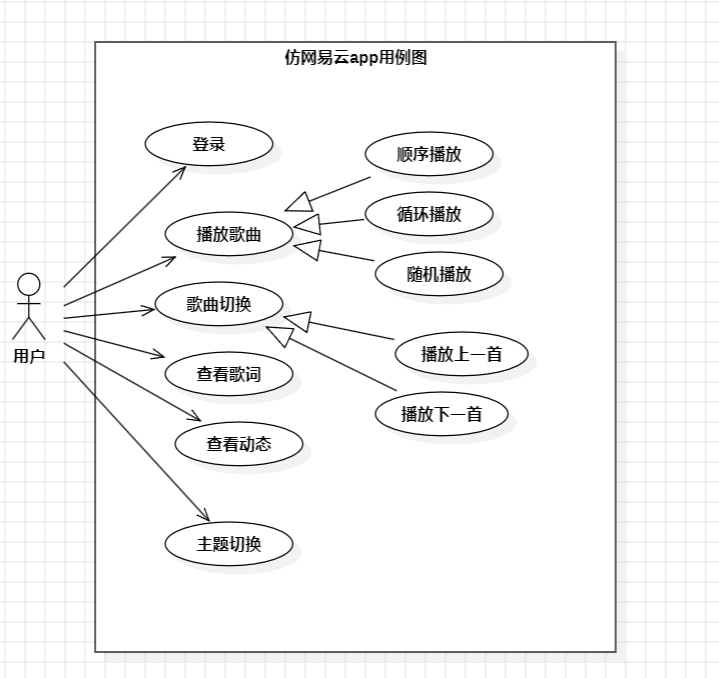
作用：管理数据库连接。处理数据库的连接和断开操作。

主要功能：

存储数据库连接的基本信息。

提供连接和断开数据库的方法。

### 3.2 功能模型



仿网易云app用例图

用户管理;

用户注册与登录：现有的网易云账户

用户个人信息管理：包括昵称、头像、性别、生日等信息的查看和修改。

账户安全：密码修改、绑定手机/邮箱、二次验证、找回密码等。

音乐内容管理

音乐库：包括歌曲、专辑、歌手、MV等的管理。

音乐推荐：基于用户历史行为、喜好、流行趋势等进行个性化推荐。

歌单管理：用户可以创建、收藏、分享歌单。

排行榜：提供新歌榜、热歌榜、飙升榜等多种类型排行榜。

搜索功能

全局搜索：支持歌曲、歌手、专辑、歌单、用户等多维度搜索。

智能推荐：根据搜索关键词推荐相关内容。

历史搜索记录：保存并显示用户的搜索历史。

播放管理

音频播放：支持播放、暂停、快进、快退、音量调节等操作。

播放列表：当前播放队列的管理，支持添加、删除、排序。

播放模式：包括顺序播放、随机播放、单曲循环等。

社交功能

动态发布与查看：用户可以发布动态，查看好友和关注对象的动态。

评论与点赞：歌曲、歌单、动态等内容支持评论、点赞和回复。

私信功能

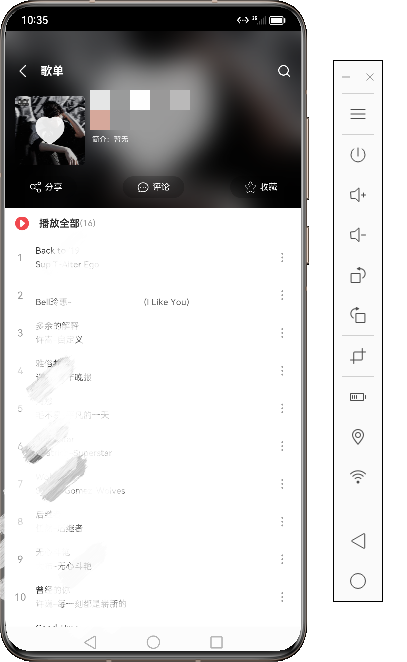
用户之间可以互发私信，进行私密交流。

## 原型设计

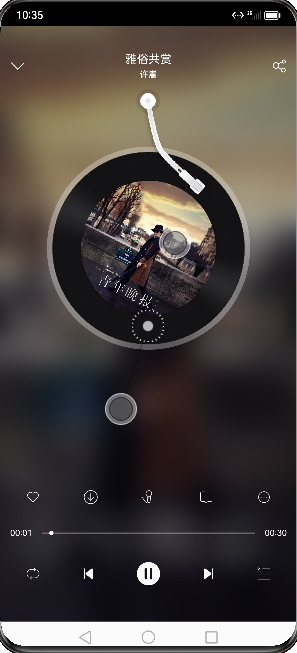
### 4.1登录界面



### 4.2歌单界面



### 4.3播放界面



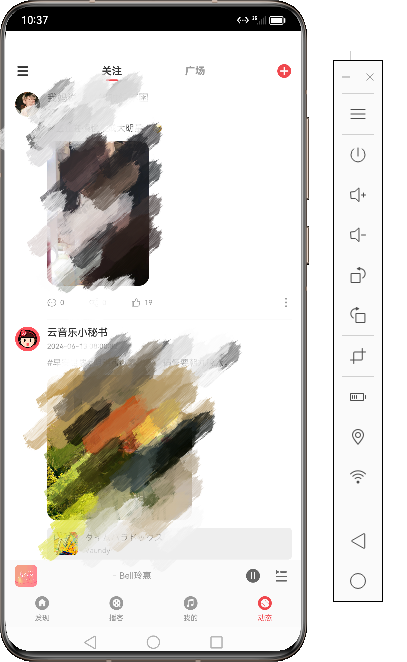
### 4.4 当播放列表界面



### 4.5 歌词界面



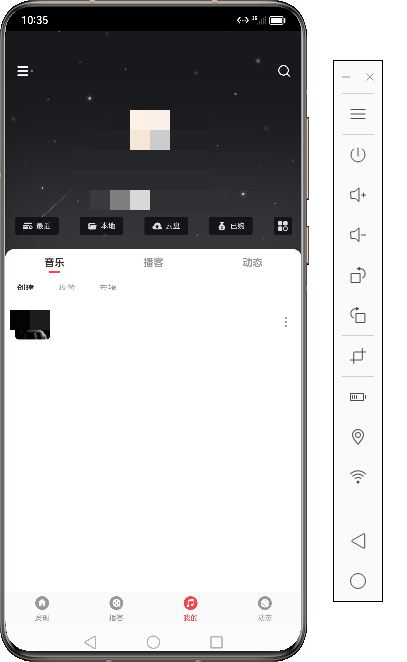
### 4.6 动态界面



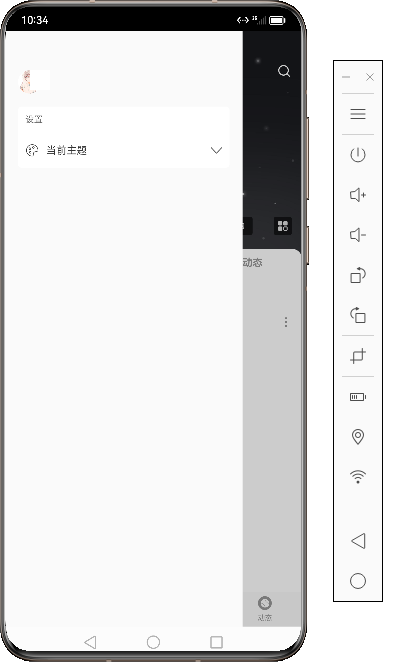
### 4.7 评论界面



### 4.8 我的界面



### 4.9设置界面



# 系统实现

## 后端实现

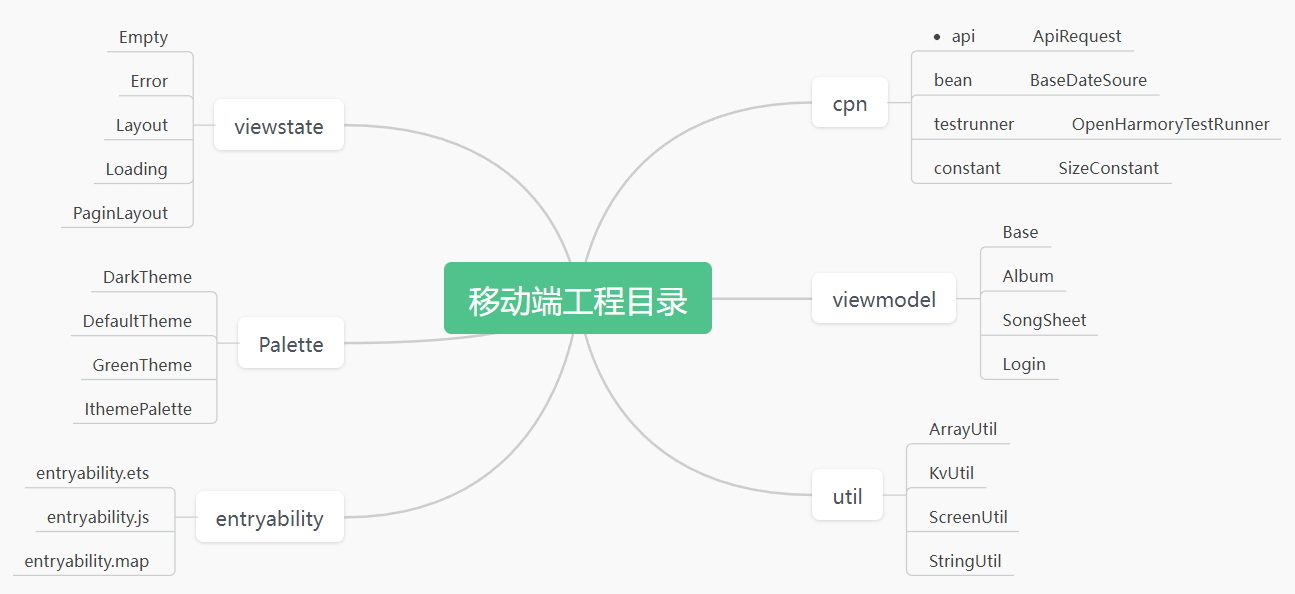
跨站请求伪造（CSRF），伪造请求头，调用网易云官方API。

## 移动端实现

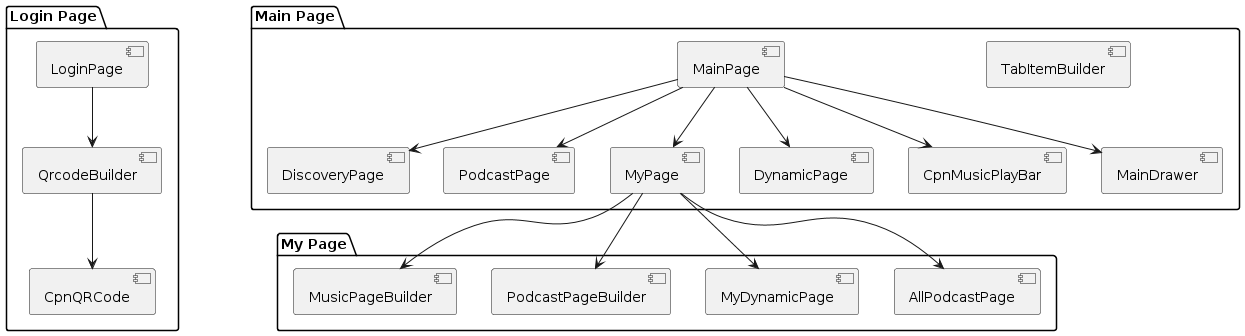
### 2.1移动端架构

ArkUI+Arkts+HarmonyOS http+网易云官方API

### 2.2 工程目录



移动端工程目录图

2.3页面结构  


页面结构图

登录

Image: 应用程序标志图像。

Text: 用于显示"扫码登陆体验"的文本。

QrcodeBuilder: 用于生成二维码的组件。

Text: 用于显示提示信息的文本

首页

Tabs: 底部导航栏。

TabContent (发现): 发现页面。

TabContent (播客): 播客页面。

TabContent (我的): 我的页面。

TabContent (动态): 动态页面。

CpnMusicPlayBar: 播放状态栏（如果有）。

MainDrawer: 侧边栏。

我的

HeaderBackground: 头部背景。

CollapsibleLayout: 可折叠布局。

AppBar: 应用栏。

StatusBar: 状态栏。

RelativeContainer: 相对布局容器。

Image (Menu Icon): 菜单图标。

Row (User Info): 用户信息行。

CommonNetworkImage (Avatar): 用户头像。

Text (Nickname): 用户昵称。

Image (Search Icon): 搜索图标。

ScrollableHeader: 可滚动头部。

CpnMyHeader: 自定义头部组件。

StickyHeader: 固定头部。

TabLayout: Tab布局。

Conditional TabLayout: 条件Tab布局。

Content: 内容部分。

TabPager (myTabMediator): 主Tab分页器。

MusicPage: 音乐页面。

TabPager (musicTabMediator): 音乐Tab分页器。

CreateMusicPage: 创建音乐页面。

CollectMusicPage: 收藏音乐页面。

AlbumMusicPage: 专辑页面。

PodcastPage: 播客页面。

TabPager (podcastTabMediator): 播客Tab分页器。

AllPodcastPage: 全部播客页面。

MyPodcastPage: 我的播客页面。

VoiceBookPage: 有声书页面。

MyDynamicPage: 我的动态页面。

动态

ViewStatePagingLayout: 状态分页布局。

ItemBuilder: 项目生成器。

CpnDynamicItem: 动态项组件。

## 模块实现

### 3.1.登录模块

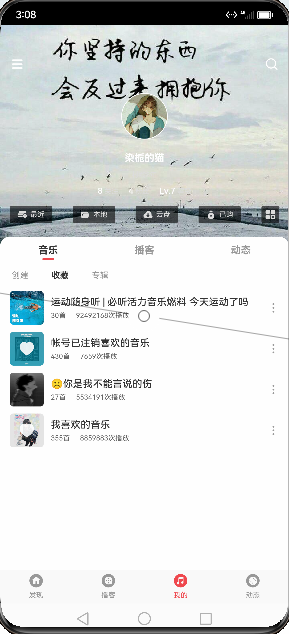
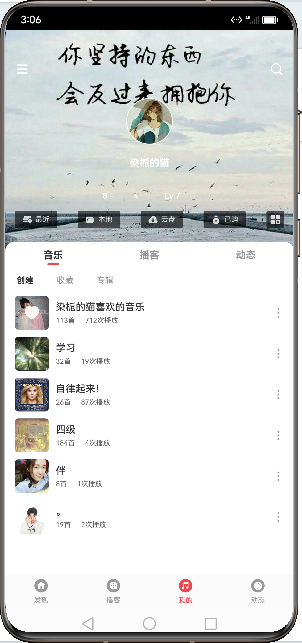
用户启动 App 软件时，首先进入登录界面页，等待加载二维码。在未登录情况下可不可以进行任意操作，等待用户扫描二维码进行登录，每四秒扫描一次查看用户是否确认登录。



### 3.2. 用户信息模块

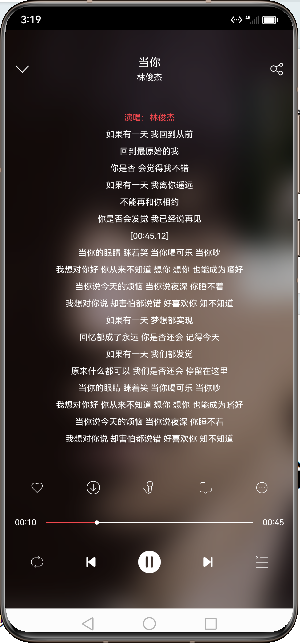
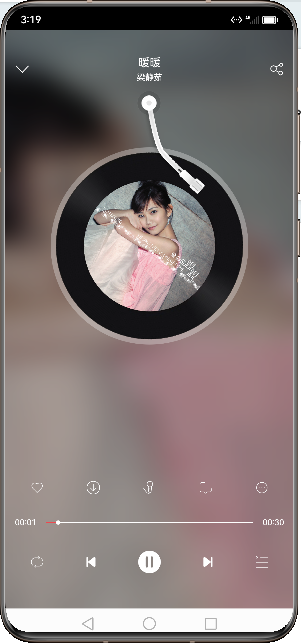
用户登录成功后进入到音乐播放模块，该模块有用户头像及用户名等用户个人信息：详情见下图，还能看到用户收藏的音乐歌单。

有用户自己创建的歌单以及专辑。



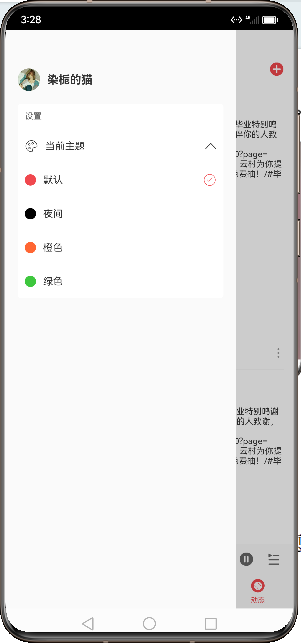
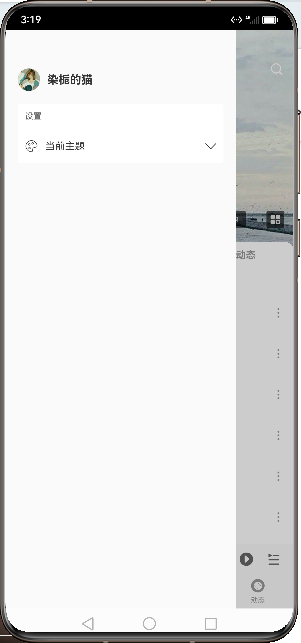
### 3.3.音乐播放模块

进入歌单之后就能够选择歌曲进行播放，在此模块能够进行暂停和播放，也能够进行歌词查看。通过点击播放方式选择按钮能够选择播放方式，比如随机，循环，顺序等，还能够调节播放进度。在下方有播放列表，可查看历史播放和当前播放列表。点击切换按钮能够切换上一首或者下一首歌曲



### 3.4设置模块

此模块能够调整app主题，通过点击当前主题查看当前主题和可更换的主题，初始用户都为默认主题，还有其他三种主题方式可进行更换。根据自己爱好点击即可。



### 3.5动态模块

查看自己所关注的博主发放的动态，，查看其他用户的评论，也能够点赞。



# 系统测试

## 测试环境

测试工具：DevEco studio

软件环境：Windows11系统

硬件环境：Win11电脑端

## 功能测试

### 2.1系统功能测试

|  |  |  |
| --- | --- | --- |
| 测试编号 | 操作 | 预期结果 |
| Case1 | 扫描二维码登录 | 扫描二维码后出现点击授权界面 |
| Case2 | 授权登录 | 用户成功登录 |
| Case3 | 主题切换 | 成功切换不同主题 |
| Case4 | 播放歌曲 | 开始播放当前歌曲 |
| Case5 | 查看歌词 | 显示出当前歌曲的歌词 |
| Case6 | 暂停播放 | 停止播放音乐 |
| Case7 | 切换上一首 | 播放上一首歌曲 |
| Case8 | 切换下一首 | 播放下一首歌曲 |
| Case9 | 循环播放 | 当前歌曲重复播放 |
| Case10 | 随机播放 | 下一首歌曲随机播放 |
| Case11 | 顺序播放 | 播放歌单列表中的下一首 |
| Case12 | 指定进度播放 | 播放指定进度所指的歌词 |
| Case13 | 点击动态查看 | 查看关注博主的动态 |
| Case14 | 点赞评论 | 点赞成功 |

## 3. 非功能测试

### 3.1界面测试

|  |  |  |
| --- | --- | --- |
| 编号 | 检查项 | 结果 |
| 1 | 界面主题是否一致 | 是 |
| 2 | 下拉是否刷新 | 是 |
| 3 | 是否有错别字 | 否 |
| 4 | 点击按钮是否有效果 | 否 |
| 5 | 页面是否正常跳转 | 是 |
| 6 | 歌单数据是否正常 | 否 |
| 7 | 能否显示歌词 | 是 |
| 8 | 点赞是否成功 | 是 |

## 4. 测试结论

受限于测试水平、时间、经验等条件限制，测试中仅采用部分测试方法对软件进行各项要求的功能和非功能测试。通过测试，发现了 App 中没有达到预期的问题，部分功能不够完善，如攻略和动态数据暂时未能实现动态加载，当歌单数据体量较大时加载响应速度不高，界面下拉刷新暂未绑定接口实现等，总体而言，通过测试，系统的整体功能基本完成、区域稳定。如果能充分利用各种测试方法，对系统做更进一层次的测试，一定可以不断完善此软件的各项功能。

# 总结

论文介绍了仿网易云 App的设计与实现过程,结合前后端分离开发和相关技术与应用，提出了产品需求分析，进而完成系统的总体设计与详细设计，并通过测试验证了产品设计的正确性。期间主要完成了以下工作:

1、研究音乐播放器的基本特征和背景意义，相关行业和音乐平台的发展历程、现状和未来趋势，研究国内外相关的研究和应用现状，介绍了相关技术、对开发工具，并对其进行实际应用。

2、研究国内对于音乐播放器需求，平台用户主体分析、产品分析定位。通过调研玩家对音乐的需求，以及国内市场此类软件的产品模式、互通点和各自特色，分析定位音乐播放器的核心用户，产品痛处和共同的业务流在，开展产品需求分析，梳理业务需求，并确认细分功能点的需求。

3、整体上对系统进行设计。确定系统的框架和技术选型和相关架构，并确定音乐播放器的产品各个结构，进行需求建模、原型经制从而保证系统设计符合实际需要:

4、详细设计和实现各个功能模块,开展部分 API服务接口的详细设计工作，选择合适的技术栈。

实际过程中遇到了很多困难，技术选择和用户体验，数据安全和隐私保护，性能可伸缩和需求变更。为了改善问题进行以下的改进：

1. 进行技术调研和评估，选择广泛使用且社区活跃的技术，如React Native或Flutter作为移动应用开发框架，Spring Boot或Node.js作为后端框架。参考类似系统的架构设计模式，例如微服务架构，以提高系统的灵活性和可扩展性。利用云服务提供商（如AWS、Azure、阿里云等）的服务，如服务器托管、数据库管理等，减少基础设施管理的复杂度。
2. 进行用户调研和原型设计，了解用户的需求和行为习惯。使用现代化的UI/UX设计工具（如Sketch、Adobe XD等），创建高保真的界面原型，并进行用户测试和反馈收集。参考已有的成功案例和UI设计规范，确保设计符合用户的预期和习惯。
3. 使用加密技术保护用户的敏感数据，如密码、个人信息等。实施严格的访问控制和权限管理，确保只有授权人员可以访问敏感数据。遵循数据保护法规和标准（如GDPR、HIPAA等），确保系统设计符合法律法规的要求。定期进行安全审计和漏洞扫描，及时修复和更新系统的安全漏洞。
4. 使用性能测试工具对系统进行负载测试，模拟高流量场景，评估系统的承载能力和瓶颈。使用缓存和CDN技术优化静态资源的访问速度，减少服务器压力。实施水平扩展和负载均衡策略，通过增加服务器实例来处理增加的流量。使用监控和报警系统实时监测系统的运行状态和性能指标，及时调整和优化系统配置。
5. 采用敏捷开发方法论，如Scrum或Kanban，将开发过程分解为短周期迭代，快速响应和适应需求变更.使用版本控制工具（如Git）管理代码和变更，确保代码的版本控制和追溯能力。与客户和产品所有者保持密切沟通，及时更新需求和优先级，确保团队理解和执行正确的开发方向。

通过有效的团队协作、技术选型和良好的项目管理实践，可以解决设计和开发仿网易云音乐App过程中的各种困难，确保项目顺利推进并达到预期的目标和质量标准。

# 参考文献

[1]萨师煊，王珊.数据库系统概论[M].第3版.北京:高等教育出版社，2000

[2]Todd Cook.JSP 从入门到精通[M].北京:电子工业出版社，2003.1-385.

[3]周绍广.JSP编程基础与实例[M].北京:机械工业出版社，2002

[4]李清国.Windows 2000+ JSP SQL Server 案例教程[M].北京:中科多媒体电子出版

社，2001

[5]程文.基于 JSP 开发动态网站的设计与实现[J].现代计算机，2002，总第一四九期:

47-50

[6]施伯乐，丁宝康，汪卫.数据库系统教程[M].第3版.北京:高等教育出版社，2008

[7]张海藩.软件工程导论[M].第5版.北京:清华大学出版社，2008

[8]汪礼成.船舶机务管理信息系统的设计与开发[D].上海:上海海运学院，2001

[9]任毅，孙建.基于 WEB 架构的 GPSS/JAVA 的研究与开发[].系统仿真技术及其研究

2006，第 10 卷

[10]Eric Armstrong ,Jennifer Ball ,Stephanie BodoffThe Java(TM)Web Services

Tutorial[EB].http://ajava.org/book/java/1082.html

[11]郭伟伟，刘端阳.JavaScript 全程指南[M].北京:电子工业出版社，2008

[12]李晓黎.JSP+SQLSERVER 网络应用系统开发指南[M].北京:人民邮电出版社，2003

[13]张小字.基于 web 的学术期刊管理系统[D].西安:西安电子科技大学，2010