形状, 圆圈

描述已自动生成卡通画

中度可信度描述已自动生成

**本科毕业设计（论文）**

题目 基于数据分析的音乐推荐播放器的设计与实现

学 院 计算机科学与工程

年 级 2018级 专 业 软件工程（嵌入式培养）

班 级 Z094192 学 号 Z09418233

学生姓名 朱心熹

校内导师 周蓓 职 称 讲师

校外导师 职 称

论文提交日期 2023-04-30

常熟理工学院本科毕业设计(论文)诚信承诺书

本人郑重声明： 所呈交的本科毕业设计(论文)，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

|  |  |  |  |
| --- | --- | --- | --- |
| 本人签名： |  | 日期： | 2023年4月30日 |

常熟理工学院本科毕业设计(论文)使用授权说明

本人完全了解常熟理工学院有关收集、保留和使用毕业设计(论文)的规定，即：本科生在校期间进行毕业设计(论文)工作的知识产权单位属常熟理工学院。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许毕业设计(论文)被查阅和借阅；学校可以将毕业设计(论文)的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编毕业设计（论文），并且本人电子文档和纸质论文的内容相一致。

保密的毕业设计(论文)在解密后遵守此规定。

|  |  |  |  |
| --- | --- | --- | --- |
| 本人签名： |  | 日期： | 2023年4月30日 |
| 导师签名： |  | 日期： | 2023年4月30日 |

基于数据分析的音乐推荐播放器的设计与实现

摘要

聆听音乐作为当今快节奏生活的一种解压方式，往往能给辛苦劳作一天的人们带来心灵与灵魂上的慰藉。通常，人们的歌单中往往只有几首经年不变的歌曲，并循环播放着歌单。造成此问题的原因往往是人们找不到能够感触自己的新歌曲，而一些好歌可能就只能被少数人珍藏在自己的歌单中。如今不少音乐播放器都有音乐推荐的功能，但是为什么还是会出现上述情况呢？往往是因为这些音乐播放器没有抓住用户的喜好，推荐的歌曲也只是当今最火的歌曲。本系统为解决此类问题，实现了一个能够根据用户歌单、喜欢的音乐为用户个性化推荐音乐的播放器。

在本系统的技术选型上，采用了当今最为流行技术栈进行开发。前端基于 Vue3 技术栈、React18 技术栈 + Vite + ESLint + TS + Electron + FFmpeg 进行开发，实现了音乐推荐、歌曲歌单分类、专辑收藏、音乐下载、播放、用户权限管理、音乐管理、歌手管理、专辑管理、歌单管理等系统功能。后端系统基于 Spring Boot + Spring Security + JWT + MyBatis Plus + Mysql + Redis + ELK 进行开发，实现了对系统数据常规的增、删、改、查等操作，同时使用 ELK 对音乐数据进行分析，并通过 IK 分词搜索技术为用户检索本系统中的音乐信息提供了便利。

**关键词：**音乐播放器 音乐推荐 Electron 音乐下载

Design and Implementation of a Music Recommendation Player Based on Data Analysis

Abstract

Listening to music, as a way to relieve stress in today's fast-paced life, can often bring comfort to the mind and soul of people who have worked hard all day. Usually, people only have a few songs in their playlists that haven't changed over the years, and play the playlists on a loop. The reason for this problem is often that people can't find new songs that can touch them, and some good songs may only be treasured by a few people in their own playlists. Nowadays, many music players have music recommendation functions, but why do the above situations still occur? Often, it is because these music players have not captured the user's preferences, and the recommended songs are only the most popular songs of the moment. This system aims to solve such problems and implements a player that can recommend music personalized to users based on their playlists and favorite music.

This system was developed using the most popular technology stack, with the front-end based on Vue3 technology stack and React18 technology stack + Vite + ESLint + TS + Electron + FFmpeg, implementing system functions such as music recommendation, song playlist classification, album collection, music download, playback, user permission management, music management, singer management, album management, playlist management and more. The back-end system was developed based on Spring Boot + Spring Security + JWT + MyBatis Plus + Mysql + Redis + ELK, implementing routine operations such as adding, deleting, modifying, and querying system data, as well as analyzing music data using ELK and providing convenience for users to retrieve music information in this system through IK word segmentation search technology.

**Key Words:** Music Player; Music recommendation; Electron; Music Download

目 录

[摘要 I](#_Toc131794550)

[Abstract II](#_Toc131794551)

[1.引言 1](#_Toc131794552)

[1.1 研究背景 1](#_Toc131794553)

[1.2 国内外研究概览 2](#_Toc131794554)

[1.3 研究意义 3](#_Toc131794555)

[1.4 课题研究工作 4](#_Toc131794556)

[1.5 工程伦理 5](#_Toc131794557)

[2.开发技术介绍 6](#_Toc131794558)

[2.1 Vue 6](#_Toc131794559)

[2.2 React 7](#_Toc131794560)

[2.3 Vite 7](#_Toc131794561)

[2.4 Element Plus 7](#_Toc131794562)

[2.5 Antd 8](#_Toc131794563)

[2.6 ECharts 9](#_Toc131794564)

[2.7 Spring Boot 9](#_Toc131794565)

[2.8 Redis 10](#_Toc131794566)

[2.9 ELK 10](#_Toc131794567)

[2.10 FFmpeg 11](#_Toc131794568)

[参考文献 12](#_Toc131794569)

1.引言

1.1 **研究背景**

伴随着当今时代信息化程度的不断加深以及互联网技术的不断前进，音乐播放器已成功走进每个人的个人电脑。许多调查研究表明，恰当的音乐可以改善工作环境的氛围，减轻压力和疲劳，因此对工作效率产生积极的影响[1]。同时，在课余时间以及工作放松之余，聆听音乐已成大多数网民每天的必修课程。

当下的音乐类型五花八门，有流行，有古典，有民谣，更有古风等多种音乐类型，而这多如牛毛的音乐类型下更有不计其数的音乐，这让广大的音乐聆听者们不知道如何选择。然而传统的音乐播放器只能按部就班的给用户推送当今最流行、最新的歌曲，却没有考虑到用户是否喜欢这类的音乐。这使得广大的用户都对播放器里面的推荐模块嗤之以鼻，甚至从用户下载播放器开始，就没有用过这模块的功能。导致不少用户一直循环播放着自己珍藏多年歌单里的音乐，甚至歌单里几年没有出现新歌曲。因而，传统音乐播放器需要持续创新和发展，以满足用户的需求[2]。

为打破此困境，本论文设计并实现了一款能够基于数据分析的音乐推荐播放器，通过分析用户歌单、收藏中的歌曲，利用按比推荐算法，在为用户推荐可能喜欢的歌曲的同时，向用户推荐未涉足领域类型的音乐，使得用户既可以获取到自己所喜好的音乐，还能尝试探索新风格的音乐。

在当今有着许多优秀且生态完整的框架以及技术来支持一个完整的软件系统开发生命周期。对于本次系统的设计中，歌曲检索模块用到了 Elasticsearch、Logstash、Kibana技术。ELK 本质上是一套开源的日志分析、日志管理软件，它们可以对日志进行收集、分析、检索等相关操作。在本系统中，它们将数据库中的音乐资源数据全量同步至 Elasticsearch中，并利用其出色的倒排索引及强大的性能，为用户提供了音乐资源的全文分词搜索。

本系统身为一个音乐播放器，必然需要拥有可以对音乐进行解析、读取的功能。FFmpeg 不失为一个绝佳的选择。FFmpeg 是一种开源的多媒体框架，它具有跨平台、高效、功能丰富等优点，因此在音视频编解码领域被广泛使用；由于该框架可以在不同操作系统上运行，因此可以方便地应用于各种设备和平台，如计算机、移动设备和网络流媒体服务器等；此外，该框架还提供了丰富的编解码器和工具，使得开发人员可以方便地完成音视频处理和转换任务[3]。在本系统中，使用 FFmpeg 作为音乐元数据读取与写入的工具，实现了下载一首歌曲的同时附带了其所有的相关信息，如歌手、专辑图片、歌词、描述的信息。使得音乐与元数据绑定，这样用户将此歌曲放入其他播放器后也无需单独下载歌曲的元数据，只要此播放器能够读取音乐的元数据。

对于本系统的基础模块开发，还应用到了 Vue3、React18、Spring Boot、MySql、Mybatis Plus 等技术框架。

1.2 国内外研究概览

1.2.1 QQ音乐

QQ音乐是一款支持音乐推荐的播放器，作为国内内容最丰富的音乐平台之一，QQ音乐拥有大量的音乐资源，并且不断更新，以满足用户的音乐需求；用户可以通过QQ音乐播放器轻松地搜索、收听和管理自己喜欢的音乐，同时还可以享受高品质的音乐音效和用户友好的界面设计[4]。QQ音乐作为腾讯旗下的音乐播放器，其音乐资源远超国内其他音乐平台，虽然有着这种先天优势，但是其音乐资源下载后只能在QQ音乐播放器中播放，如果用户想要将歌曲下载到自己的手机或者 MP3 中，则无法播放。

1.2.2 网易云音乐

作为网易音乐团队主导研发的音乐产品，其主要的目标是为用户提供优质的原创歌曲，不少网络音乐人都有在此平台上活跃；得益于其年轻、活跃的产品气氛，深受当今年轻人的喜爱；用户可以在网易云音乐中轻松地搜索和收听自己喜爱的音乐，同时还可以享受到丰富的社交和互动功能，如评论、分享、打赏等[5]。它下载的音乐资源与QQ音乐一样，只能在自己的播放器中播放，致使音乐的分享收到限制。

1.2.3 Spotify

Spotify是一个国外的歌曲服务平台,于二零零八年开始在瑞典城市斯德哥尔摩推出；在此平台上，免费用户在使用过程中会接收到部分广告，付费用户则可以享受无广告和更高品质的音乐；此外，付费用户还可以获得其他额外的功能和服务，例如离线播放、无限跳过、高品质音乐流和个性化推荐等；无论是免费用户还是付费用户，Spotify都提供了丰富的音乐资源和用户友好的界面设计，方便用户发现和收听自己喜欢的音乐[6]。身为一款国外的音乐播放器，最大的问题就是限制国内的访问，同时其音乐资源基本上都是以英文歌曲为主，只受国内小众用户青睐。

1.3 研究意义

1.3.1 用户价值

音乐推荐播放器的研究意义之一在于提高用户体验。传统的音乐播放器只能通过搜索、分类等方式让用户寻找自己喜欢的音乐，缺乏个性化推荐服务。而基于数据分析的音乐推荐算法可以根据用户的历史偏好、播放记录和行为模式等数据，为用户推荐更加符合其口味的歌曲和歌单，提高用户的满意度和体验感受。

1.3.2 发展价值

音乐推荐播放器的研究意义之二是推动音乐产业的发展。音乐产业是一个数字化程度较高的行业，而基于数据分析的音乐推荐算法可以帮助音乐公司更好地了解用户需求，优化音乐资源配置，从而推动音乐产业的健康发展。此外，音乐推荐播放器还可以帮助新歌曲和新歌手更快地被观众发现，从而推动音乐市场的多元化和创新发展。

1.3.3 文化价值

音乐推荐播放器的研究意义之三在于促进音乐文化的传播和交流。基于数据分析的音乐推荐算法可以根据用户的喜好和偏好，为用户推荐不同国家、不同地区和不同风格的音乐，从而促进音乐文化的传播和交流。此外，音乐推荐播放器还可以为用户提供多语言歌曲和翻译服务，促进不同国家和地区的音乐文化交流。

1.4 课题研究工作

本系统使用了当下最为火热且最新的技术进行开发，程序 UI界面符合当下年轻人的审美，系统功能齐全，操作简单。本系统使用的技术及工作内容主要包括以下几点：

1.4.1 桌面端播放器

桌面端播放器基于 Vue3 + Vue Router + Pinia + Vite + ESLint + TS + ElementPlus + Electron + FFmpeg 进行开发；通过 Vue3 技术栈，配合使用 Pinia 作为状态管理工具，并使用 ESLint 作为代码规范工具，对前端基础模块进行组件化、规范化开发，实现登录、注册、主页推荐、歌曲歌单分类、专辑收藏、喜欢的歌单、音乐下载、播放列表、播放页面、滚动歌词、播放控制等基础页面功能；结合 Electron 中主进程与渲染进程通讯，配合监听 Electron 的 will-download 事件对音乐下载进度进行管理，实现批量下载、暂停下载、继续下载、取消下载以及当异常退出程序后的下载恢复等基础功能；利用 FFmpeg 对下载完成的音乐进行歌曲标题、专辑信息、歌手信息、歌词、歌曲封面等音乐元数据的写入，使 PC 端、移动端等默认播放器可读取该音乐的元数据，解决了在一些大众音乐平台所下载的音乐中只有音乐而无音乐基本信息的问题；同时可对本地音乐的元数据信息进行读取、解析。

1.4.2 Web端音乐后台

Web 端音乐后台基于 React18 + React Router + Redux Toolkit + Vite + ESLint + TS + Ant Design + ECharts 进行开发；通过 React18 技术栈，配合使用 Redux Toolkit 作为状态管理工具，并使用 ESLint 作为代码规范工具，对项目页面进行函数式组件化、规范化开发，实现了管理员注册、管理员登录、个人中心、用户管理、角色管理、权限管理、音乐管理、歌手管理、分类管理、专辑管理、歌单管理、收藏管理等系统功能；使用 ECharts 绘制系统主页大屏，提供可视化的系统信息展示，清晰的展示了系统中用户、音乐等数据信息；结合 React Router 与后端基于 rbac 的权限控制系统，实现前端动态路由以及按键级的权限控制，如动态导航栏、动态按钮等。

1.4.3 音乐收集

音乐收集基于 Node.js + Koa2 + TS + Mysql + 网易云音乐 API 进行开发；利用 Node.js、Koa2 对整体项目框架进行前后端分离模式的音乐收集接口搭建；通过以网易云音乐 API 作为数据来源，对音乐数据进行结构化收集；结合 Mysql 对音乐数据进行数据入库，并结合 Node.js 的 fs 模块进行音乐数据下载，实现音乐数据本地化存储。

1.4.4 后端系统

后端系统基于 Spring Boot + Spring Security + JWT + MyBatis Plus + Mysql + Redis + Elasticsearch + Logstash + Kibana 进行开发；通过 Spring Boot 结合 Spring Security、JWT 对系统进行动态鉴权并基于 rbac 模式进行权限设计，同时利用 Redis 对 token 进行存储与管理；结合 Redis 重写 MyBatis Plus 二级缓存，大幅减少了数据库的查询压力，并很大程度上提高了接口的访问速度；利用 Elasticsearch 结合 Logstash 实现 Mysql 数据全量同步至 Elasticsearch，并通过 Elasticsearch 的 IK 分词索引查询实现实现音乐数据的检索；使用 Mysql 对用户音乐喜好进行分析，为不同用户检索出自己喜爱的音乐。

1.5 工程伦理

对于本系统中的工程伦理，将有以下几个方面的的讨论：

1.5.1 用户的隐私保护

本音乐推荐播放器将收集用户的个人对音乐喜好的数据，如收藏的歌单、歌曲等，以便进行音乐的个性化推荐。在收集和处理用户数据时，需要严格遵循用户的隐私保护原则，确保用户数据的安全和私密性。

1.5.2 用户知情权和选择权

用户应该具有知情权和选择权，可以自主选择是否使用本音乐推荐播放器。因此，在设计和实现音乐推荐播放器的同时，需要明确的告知用户其数据的收集、使用方式。

1.5.3 商业利益及公共利益间的关系

本音乐推荐播放器旨在给用户提供方便、快捷、个性化的音乐播放服务，其中的音乐来源可能涉及商业利益和公共利益的权衡。在处理此关系时，需审慎权衡各种利益关系，并严格确保公共利益不受商业利益的影响。

2.开发技术介绍

2.1 Vue

Vue 是一款可以扩展和灵活定制的前端框架，它的设计理念是“渐进式框架”，可逐步集成到现有项目中或用于构建全新的应用程序。

Vue 的核心类库提供了数据绑定、组件化和事件处理等基本功能，开发者可以根据需求灵活选择使用这些功能，并结合其他库和工具进行集成。此外，Vue还提供了一系列插件和工具，如 Vuex、Vue Router 和 Vue CLI，以帮助开发者构建大型和复杂的应用程序。这些插件和工具都设计为可插拔的方式，开发者可以根据需求选择使用。

总的来说，Vue 的设计非常注重灵活性以及可扩展性，因此 Vue 非常适用于各种形式和规模的项目。开发者可以根据自己的需求选择性地使用 Vue 的功能，并将其与其他库和工具进行集成，从而构建出高效和灵活的单页面应用程序。

2.2 React

React是一个功能强大、使用灵活且运行高效的 JavaScript 库。它采用陈述式的编程风格，使得开发者可以通过非常简洁的代码就可以实现非常复杂的 UI。React 因为其组件化开发的思想，所以可以将 UI 拆分为非常多个小且独立的组件，很大程度上提高了代码的可维护性和可重用性。

React 的弹性和可扩展性也是它的优势之一，它可以与其他库和框架轻松得进行无缝集成。正是由于它使用了虚拟DOM技术，才使得其可以快速地进行 DOM更新和操作，大大地提升了应用程序的性能与响应速度。不仅如此，React 还完美的支持了服务端渲染，可以提高应用程序的SEO友好性。

2.3 Vite

Vite 是一款有 Vue 原班人马打造的全新的前端构建工具，它相对于 Webpack 在热模块更新速度上有着巨大的提升。Vite 基于原生 ES 模块，提供了丰富的内置功能，其中包括速度极快的模块热更新（HMR）功能。Vite 使用 Rollup 打包代码，并预配置了用于生产环境的高度优化的静态资源输出。

总的来说，Vite 是一种全新的前端构建工具，它可以极大地提高前端开发的效率和体验。它由开发服务器和构建指令两部分组成，其中开发服务器基于原生ES模块，提供了丰富的内置功能，包括速度极快的模块热更新功能。Vite 预设了在生产环境使用的配置，使得开发者无需过度配置便能投入使用。得益于开发友好、高效的特性，使其成为前端开发中不可或缺的工具。

2.4 Element Plus

Element Plus 是一款基于 Vue.js 的开发的 UI 组件库，是 Element UI 的升级版。它不仅向开发者提供了一系列美观、易用、高质量的 UI 组件及配套的功能，还帮助开发者轻松构建高质量的 Web 应用程序。

该组件库不仅包括了常用的导航、表格、按钮、弹窗、表单、布局等组件，同时还提供了样式定制选项和丰富的主题，开发者可以根据需求来进行灵活的扩展和定制。此外，Element Plus 还提供了对支持 TypeScript 的 API 的支持和组件，使得开发者在开发过程更加高效及可靠。

Element Plus的文档和示例十分丰富，可以帮助开发者快速上手并使用该组件库。总的来说，Element Plus是一款定制灵活、功能强大、易用美观的 UI 组件库，是Vue.js开发中不可或缺的工具之一。

2.5 Antd

Antd 是一款使用方便快捷、界面友好的 UI 组件库，由蚂蚁大前端团队负责开发和维护。Antd提供了一系列高质量、美观且易于使用的UI组件和配套功能，帮助开发者快速构建Web应用程序。

Antd的组件库包含了常用的按钮、表单、表格、弹窗、导航、布局等组件。此外，Antd还提供了丰富的主题和样式定制选项，可以根据需求进行灵活的定制和扩展。同时，Antd支持TypeScript的API和组件，提高了开发效率和代码的可靠性。

Antd 的文档和示例非常丰富，可以帮助开发者快速上手使用该组件库。Antd 还提供了其他配套的工具和插件，如脚手架工具和代码风格检查插件等，以提高开发效率和代码质量。

总的来说，Antd 是一款强大、美观、易用且高度可定制的 UI 组件库，对于React开发者来说是不可或缺的工具之一。如果你想要快速构建高质量的 Web 应用程序，Antd 是一个很好的选择。

2.6 ECharts

ECharts 是一种 JavaScript 图表库，由百度公司开发，用于呈现各种复杂数据。它能够支持多种图表类型，包括线性图、柱状图、散点图、饼图和地图等。ECharts 采用基于 Canvas 的渲染技术，可以快速渲染大量数据，并在各种浏览器和设备上运行。

除了常规的静态图表展示， ECharts 还支持动态交互，以进行数据探索和分析。用户可以使用 ECharts 提供的工具进行缩放、拖拽、数据标注、数据筛选等操作，以更好地理解和分析数据。此外， ECharts 还提供多语言支持和丰富的定制化选项，以满足用户多样化的需求。

ECharts 是一个开源项目，源代码和文档可在 GitHub 上免费获取。除了使用 ECharts 自带的图表模板外，用户还可以使用 ECharts 提供的 API 进行自定义开发，以满足其需求。

2.7 Spring Boot

Spring Boot 是在 Spring 基础上增加了常用配置的后端框架，开发者能够使用它来快速构建自己的 Spring 应用。它通过预先配置了大量在 Spring 中需要手动配置的设置，大大的简化了开发者开发后端系统的时间，使得开发人员能够更快速地实现业务逻辑。

Spring Boot 提供了许多特性，如自动配置、嵌入式 Web 服务器、健康检查、指标度量和安全性等。此外，它还支持多种数据源和持久化技术，如 JDBC、Redis、ES 等，以轻松地集成第三方库和组件。

Spring Boot 还提供了一个丰富的社区和生态系统，其中包括 Spring Security、Spring Batch、Spring Cloud 等项目。这些项目提供了各种功能和扩展，可以进一步增强 Spring Boot 应用程序的功能和性能，同时也有助于提高开发效率和代码质量。

2.8 Redis

Redis 是一种基于内存的开源数据结构存储系统，也是一种 NoSQL 数据库，被广泛应用于互联网开发领域。Redis 支持多种常见得数据结构，同时还提供了各种高级特性，如事务、持久化、复制和发布/订阅等，可以满足各种复杂的数据处理需求。

由于 Redis 是通过内存来存储数据得，因此它的读写速度非常快。此外，Redis还支持集群部署，可以实现数据的分布式存储和高可用性。

Redis 非常易于安装和使用，提供了丰富的客户端库和命令行工具，支持多种编程语言，如 Java、Python、PHP 等。此外，Redis 还有一个庞大的社区和生态系统，提供了各种扩展和工具，可以进一步增强 Redis 的功能和性能。

2.9 ELK

ELK 是一个由三个开源软件组件组成的日志管理平台，包括 Elasticsearch、Logstash 和 Kibana。这些组件提供了日志收集、存储和分析的功能，可用于管理和分析日志数据，帮助开发人员、系统管理员和数据分析师更好地监控系统运行状况、发现问题和优化性能。

Elasticsearch 是一个基于 Lucene 的分布式搜索引擎，可用于全文检索、日志存储和数据分析等领域。Logstash 是一个数据收集和转换工具，可从各种来源收集日志数据，并将其转换为统一的格式。Kibana 是一个数据可视化工具，可创建各种图表、仪表盘和报表，以展示和分析日志数据。

ELK 支持实时数据处理和可扩展性，可处理海量的日志数据，并快速地进行搜索、过滤和聚合等操作。它的安装和配置相对简单，同时还有丰富的社区和生态系统支持，提供了各种插件和工具，可以满足各种特定的日志管理需求。因此，ELK 在互联网开发中得到了广泛的应用，成为了一种必不可少的工具。

2.10 FFmpeg

FFmpeg 是一个跨平台的开源音视频处理工具，可以用于录制、转换、编辑和流媒体处理等多种应用场景。它支持多种音视频格式，包括 MP4、AVI、WMV、FLV、MP3、AAC 等，可以在不同的平台和设备上播放和编辑这些格式的音视频文件。

FFmpeg 提供了丰富的命令行工具和 API，可以实现各种音视频处理操作，如剪辑、合并、分离、转码、裁剪、调整音量、添加字幕等。它还支持多种视频编解码器和音频编解码器，可以将不同格式的音视频文件进行相互转换。

FFmpeg 的应用范围非常广泛，包括音视频编辑、视频转码、直播流处理、视频压缩等多个领域。它在互联网、娱乐、广告、教育等行业中得到了广泛的应用。

由于 FFmpeg 是一个开源的工具，因此它具有可扩展性和灵活性，用户可以根据自己的需求进行二次开发和扩展。此外，FFmpeg 的社区和生态系统非常活跃，提供了丰富的文档、教程、插件和工具，可以帮助用户更好地使用和应用这个工具。

3.系统分析与设计

3.1 可行性分析

本系统在技术方面采用了当下最新、使用最广的技术，它们具有良好的稳定性，不仅如此，它们的生态环境也是非常的完善。

在音乐数据获取模块中，采用了在 GitHub 上具有近 30k star 的网易云音乐 API 开源项目，通过它可以轻松的将网易云音乐中的音乐资源存储至自己的移动硬盘及数据库中。

在桌面端音乐播放器模块的设计与实现中，采用了 Vue3 + Vite + TS + Element Plus + Electron + FFmpeg 进行开发。Vue3 是由尤雨溪团队打造的能够快速搭建企业级单页面应用的框架，备受国内各大厂商的好评。结合 VIte、TS 以及 Element Plus，能在短时间内搭建起既符合模块化开发思想，又满足系统所需所有功能的完整系统。对于音乐资源的处理，如音乐元数据的读取与写入，采用了开源的 FFmpeg 工具，此工具能够通过 JS 来写入与读取音乐元数据，这使得本系统中音乐资源与音乐元数据绑定的功能得以实现。当然，纯粹的浏览器应用是无法运行此工具的，因此本系统使用了 Electron 作为应用的运行容器，提供了渲染层所需的在 NodeJS 环境运行的 API。

对于 Web 端音乐后台管理系统的设计与实现中，采用了 React 技术栈进行开发。React 在使用上比 Vue3 更加灵活，可以个性化的实现各种所需的效果与功能。结合 Antd，使得后台管理系统更易开发。在本模块中，采用了 ECharts 作为系统信息可视化展示的工具，可将系统信息以图表的方式清晰的呈现在用户眼中。

在本系统后端的开发上，采用了 Sprong Boot 作为开发框架，结合 Spring Security、MyBatis Plus、MySql、Redis、ELK 作为系统功能实现的工具，快速的构建出了一个具有 rbac 权限控制、数据库读取缓存、音乐数据分词检索、歌行推荐用户音乐喜好的完整系统。

综上所述，本系统在开发上具备可行性，并且符合本课题要求。

3.2 系统用例分析

在本系统的设计中，采用 rbac 的权限控制，将本系统的角色划分成五种：普通角色、后台普通角色、用户管理员角色、系统管理员角色、音乐管理员角色。

3.2.1 普通角色用例分析

普通角色只能在桌面端音乐播放器上进行登录，并具备在本模块上所有的功能。普通角色的用例如图 3-1 所示：

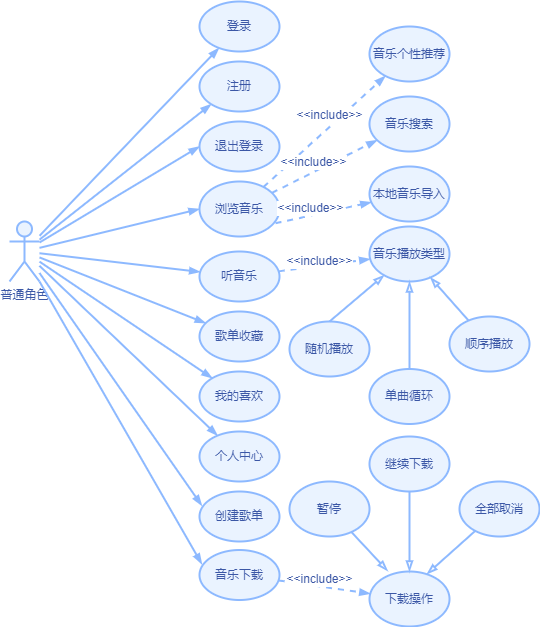


图3-1普通角色用例图

3.2.2 后台普通管理员角色用例分析

后台普通管理员角色可在后台管理系统中登录，限制其只拥有查看主页、个人中心、修改个人信息的功能。后台普通管理员角色的用例如图 3-2 所示：

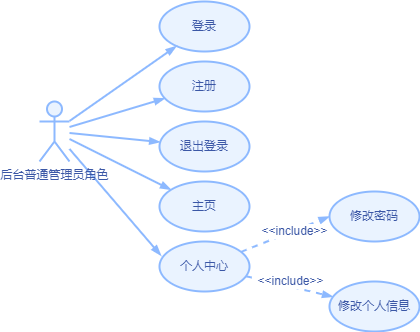


图3-2后台普通管理员角色用例图

3.2.3 用户管理员角色用例分析

用户管理员角色可在后台管理系统中登录，并具有主页、个人中心、修改个人信息、用户管理的功能。用户管理员角色的用例图如图 3-3 所示：

图示

描述已自动生成

图3-3用户管理员角色用例图

3.2.4 系统管理员角色用例分析

系统管理员角色拥有首页、个人中心、个人信息修改、系统用户管理、权限管理、用户角色管理、系统用户权限管理等基本功能。系统管理员角色用例图如图 3-4 所示：

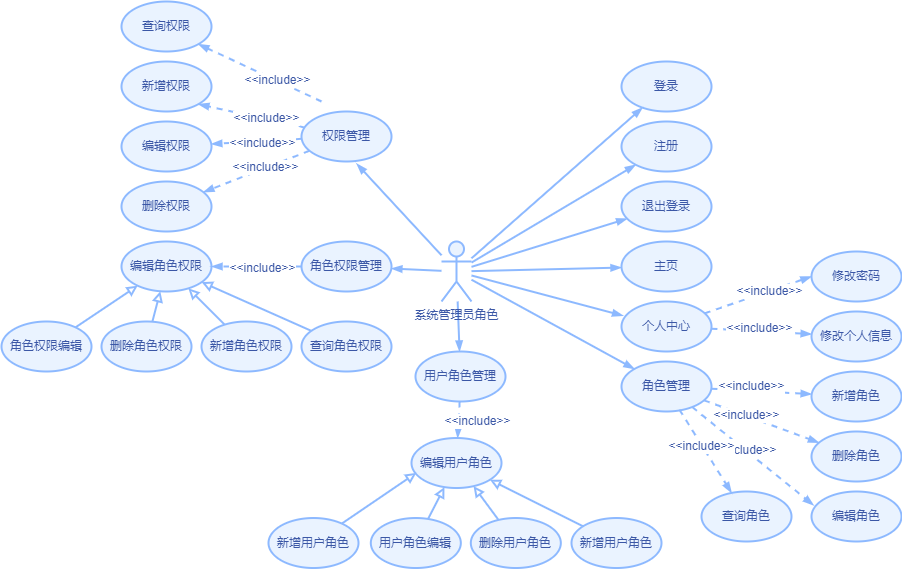


图3-4系统管理员角色用例图

3.2.5 音乐管理员角色用例分析

音乐管理员角色具备对本系统中音乐相关数据的操作权限，如音乐管理、专辑管理、歌手管理、歌单管理、音乐分类管理等权限。音乐管理员角色用例图如图 3-5 所示：

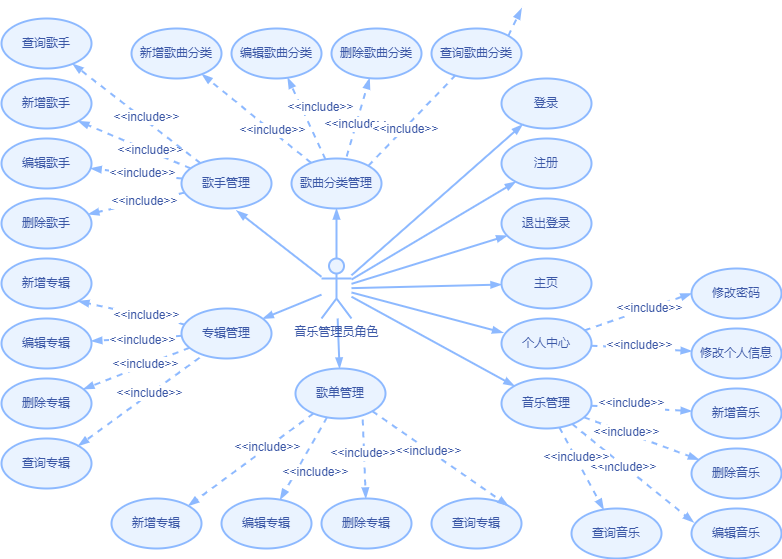


图3-5音乐管理员角色用例图

3.3 系统架构设计

前端和后端分离是现代化 Web 应用程序的普遍实践。前端负责展示数据和交互逻辑，后端则负责处理请求和响应数据。因此本系统采用了前后端分离的开发模式，将系统划分为五层：视图层、控制层、业务层、数据层和工具层：

3.3.1 视图层

视图层是前端的核心模块，负责展示数据和用户交互。在音乐推荐播放器中，视图层需要使用 Vue3 及 React 框架来实现以下功能：

（1）音乐播放页展示：展示用户当前播放的音乐，包含滚动歌词、音乐专辑封面等。

（2）搜索功能：采用 Elasticsearch的 IK 分词功能，实现用户检索歌曲、歌手、分类、歌词或专辑。

（3）播放器控制：允许用户控制音乐的播放、暂停、停止、单曲循环、顺序播放和随机播放等。

（4）推荐列表：在主页个性推荐模块中展示推荐给用户的音乐、歌单列表。

（5）歌曲下载：使用 Electron 中的 willDownload 事件和 FFmpeg对音乐进行下载，实现批量下载、暂停下载、断点续传、音乐元数据写入的功能。

3.3.2 控制层

控制层是响应前端、对接后端数据的中间层，负责处理视图层和业务层之间的交互。在音乐推荐播放器中，前端采用 Axios 向后端发送请求，控制层将响应数据返回给视图层。

3.3.3 业务层

业务层是后端的核心模块，负责授权与用户权限认证、处理来自控制层的用户请求、推荐音乐、数据存储等。在音乐推荐播放器中，业务层需要使用 Spring Boot 框架来实现以下功能：

（1）用户认证：采用 rbac 的权限控制方式，对用户权限进行设置与验证，保护用户数据与系统的安全性。

（2）音乐推荐：根据用户的收藏的歌曲、歌单等信息，向用户个性化推荐音乐。

（3）数据检索、编辑：响应控制层接收来自视图层的用户请求，向用户提供请求数据、数据编辑功能。

3.3.4 数据层

数据层同样是后端的核心模块，负责提供数据访问和管理功能。在音乐推荐播放器中，数据层使用 MySql、Redis、Elasticsearch 来存储歌曲、歌单、专辑、歌曲分类及缓存等数据；使用移动硬盘作为音乐资源的存储容器。

3.3.5 工具层

工具层是整个系统的辅助模块，负责提供音乐数据的获取、音乐元数据的读取与写入等功能。在音乐推荐播放器中，工具层需要使用 NodeJS 和 FFmpeg 等工具来实现以下功能：

（1）音乐数据获取：利用网易云音乐 API 的开源项目，使用 NodeJS 将音乐数据存储至 MySql 及移动硬盘中。

（2）音乐元数据的读取与写入：使用 FFmpeg 读取音乐的元数据，可在不联网的情况下获取到音乐的基本数据，如专辑封面、歌词、歌词名称、歌手等数据；在歌曲下载的过程中，FFmpeg 可将音乐信息写入音乐中，满足音乐与音乐信息的一体性。

综上所述，基于数据分析的音乐推荐播放器的系统架构设计包括前后端分离、视图层、控制层、业务层、数据层、工具层等多个模块。这些模块共同协作，实现了音乐推荐播放器的各项功能，并为用户提供了良好的使用体验。具体系统架构图如图 3-6 所示：

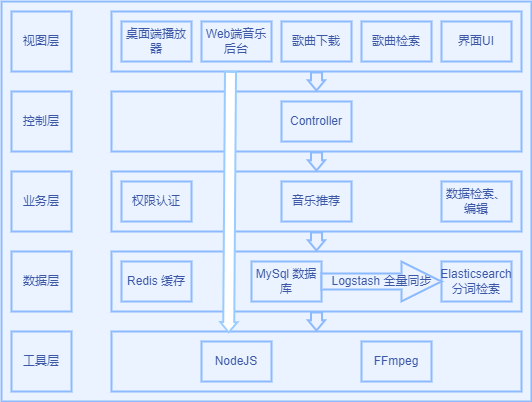


图3-6系统架构图

# 参考文献

1. 杨春红.听喜欢的音乐对工作效率的影响研究[J].中国现代商业,2021,17(9):126-127.
2. 王梦婷,王承雨.传统音乐播放器在数字时代的变革与创新[J].数字时代,2020,(2):91-92.
3. 张亚飞,陈志强.基于FFmpeg的音视频编解码技术研究[J].电脑知识与技术,2019,15(27):89-91.
4. [EB/OL].https://baike.baidu.com/item/QQ%E9%9F%B3%E4%B9%90/1157130
5. [EB/OL].https://baike.baidu.com/item/%E7%BD%91%E6%98%93%E4%BA%91%E9%9F%B3%E4%B9%90/4453795
6. [EB/OL].https://baike.baidu.com/item/Spotify/3266385