



景德镇陶瓷大学
JINGDEZHEN CERAMIC INSTITUTE

工学学士学位论文（设计）

题目：基于微信小程序云开发的音乐播放系统的设计与实现

学 号： 116060500127
姓 名： 熊梦霞
院（系）： 信息工程学院
专 业： 16 级数字媒体技术
完成日期： 2020.5.24
指导老师： 余立毅

分类号_____

学校代码_____

UDC _____ 密级 _____

学 号 116060500127

景德镇陶瓷大学
信息工程学院毕业论文

基于微信小程序云开发的音乐播放系统的设计与实现

熊梦霞

指导老师：余立毅 讲师 _____ 景德镇陶瓷大学 信息工程学院

申请学位级别 工学学士 专业名称 16 级数字媒体技术

论文提交日期 2020 年 5 月 24 日

论文答辩日期 2020 年 5 月 24 日

学位授予单位和日期 _____ 景德镇陶瓷大学

答辩委员会主席： 周永正教授

论文评阅人： _____ 许祥

2020 年 5 月 24

摘要

音乐播放器是人们时常使用的一个软件，通过音乐软件人们可以得到放松和愉悦。目前市面上已有很多公司制作的音乐 APP 和小程序，但这些 APP 普遍存在一些弊端，比如很多不必要的功能占用了手机太多内存，降低了用户体验。在这个背景下，课题开发了一个基于小程序云开发的音乐播放器，具有轻量，功能完善，界面美观、操作便捷的特点，不仅能播放音乐，还能分享图片和文字并评论其他人的图片和文字满足用户的社交需求。

本系统的开发模式是微信小程序的云开发模式。小程序是一种新型的程序类型，具有高效开发、节约成本、稳定可靠的优点，小程序云开发是 Serverless（无服务开发）架构的一种。

本系统对于小程序端的开发借助了微信平台 and 腾讯云平台，开发了可以收听音乐的音乐模块、可以发布分享图片的发现模块、可以查看最近播放歌曲我的模块。开发过程中调用了网易云音乐的推荐歌单 API 接口、微信生成小程序码接口等第三方工具。服务端主要使用小程序云开发进行开发，云开发主要包括云函数、云数据库、云存储等基本功能，通过直接使用云开发的基础功能来实现小程序后端服务。CMC 端则借助了 vue-template-admin 框架和 Koa 框架开发了管理小程序后台数据的管理系统。

课题完成后可服务于使用智能手机和微信的用户，打开小程序后无需登录即可收听每日自动更新的歌单，进入我的页面可以查看最近播放的歌曲和自己发表过的文字与图片。用户授权后即可发表图片及文字，也可以评论其他用户发表的图片 and 文字，这些数据将会保存在云数据库中，可以随时查看。对于后台的数据可在用户管理系统进行管理。

关键词：云开发；微信小程序；音乐播放系统；后台管理系统

Abstract

Music player is a software that people often use, through music software people can get relaxation and pleasure. There are many music apps and small programs produced by companies on the market, but these apps are generally a bit of a disadvantage, such as many unnecessary features take up too much memory of the phone, reducing the user experience. In this context, the project developed a small program cloud-based music player, with lightweight, perfect function, beautiful interface, easy to operate characteristics, not only to play music, but also to share pictures and text and comment on other people's pictures and text to meet the social needs of users.

The development mode of this system is the cloud development mode of WeChat small program. Small program is a new type of program, with the advantages of efficient development, cost saving, stable and reliable, small program cloud development is a kind of Serverless (serviceless development) architecture.

The system for the development of miniprogram end with the help of WeChat platform and Tencent cloud platform, developed a music module that can listen to music, can publish a discovery module to share pictures, can view the recent playing of songs of my module. In the development process, third-party tools such as the recommended song sheet API interface of NetEase Cloud Music and weChat generation small program code interface were called. The service side mainly uses small program cloud development for development, cloud development mainly includes cloud function, cloud database, cloud storage and other basic functions, through the direct use of the basic functions of cloud development to achieve small program back-end services. The CMC side has developed a management system for managing background data for small programs with the vue-template-admin framework and the Koa framework.

When the project is complete, it can serve users who use smartphones and WeChat, open a small program and listen to daily automatically updated songs without logging in, and go to my page to view recently played songs and text and pictures of yourself. Users can publish images and text once authorized, or comment on images and text posted by other users, which will be stored in the cloud database and can be viewed at any time. Data for the background can be managed in the user management system.

Keywords : Cloud Development; WeChat Small Program; Music playback system; Background management system

目录

第一章 绪论.....	1
1.1 课题背景与研究意义.....	1
1.2 研究现状分析.....	1
1.3 论文结构概述.....	2
第二章 需求分析.....	3
2.1 可行性分析.....	3
2.2 系统用户角色分析.....	4
2.3 系统功能需求分析.....	6
2.4 系统数据流图设计.....	8
2.5 系统性能分析.....	11
第三章 相关理论和技术分析.....	12
3.1 微信小程序开发技术.....	12
3.2 小程序云开发技术.....	12
3.2.1 云函数.....	12
3.2.2 云数据库.....	13
3.2.3 云存储.....	13
第四章 系统设计.....	15
4.1 系统物理设计.....	15
4.2 系统功能设计.....	16
4.3 系统数据库设计.....	19
4.3.1 系统数据库结构设计.....	19
4.3.2 系统数据库物理设计.....	19
第五章 系统实现与测试.....	22
5.1 开发环境的搭建.....	22
5.2 播放功能.....	24
5.2.1 播放功能实现.....	24
5.2.2 播放功能测试.....	27
5.3 博客功能.....	27
5.3.1 博客功能实现.....	27
5.3.2 博客功能测试.....	29
5.4 歌单管理功能.....	29
5.4.1 歌单管理功能实现.....	29

5.4.2 歌单管理功能测试.....	30
5.5 轮播图管理功能.....	30
5.5.1 轮播图管理功能实现.....	30
5.5.2 轮播图管理功能测试.....	30
5.6 博客管理功能.....	31
5.6.1 博客管理功能实现.....	31
5.6.2 博客管理功能测试.....	31
第六章 总结与展望.....	32
致谢.....	33
参考文献.....	34

第一章 绪论

1.1 课题背景与研究意义

随着移动端市场的快速发展,人们的精神世界也与移动互联网的融合越来越深刻,手机已经占据了大部分人的日常,音乐向移动互联网的转移也顺应了工业文明的发展,解决了音乐播放的局限性,促进了音乐的传播,使音乐的价值得到了更多的体现。因此,设计一个简洁的音乐播放软件对于音乐的广泛传播有很大的促进作用,相对于臃肿的音乐播放 APP,微信小程序有传承了微信软件“少即是多”的特点,没有庞大的功能和炫目的外观,CPU 和内存占用率低^[1]。因此设计了一个功能完善、界面美观的微信小程序音乐播放系统。本系统不仅开发了基本的音乐播放功能,还开发了个人微型博客功能,满足了用户的社交需求,既为音乐提供了良好的展示平台,又丰富了用户体验。本系统使用微信小程序推出的云开发模式,致力于帮助开发者更加便捷实现小程序业务开发的云开发是微信团队和腾讯云共同提供的 Serverless 云服务。

1.2 研究现状分析

无服务计算(Serverless computing),作为云计算的模型之一,也叫做功能即服务。无服务运算将平台即服务(PaaS)为根基来供给一个微型的架构,终端客户不须要部署、配置或管理服务器服务,由云端平台来供应代码运行所需要的服务器服务^[2]。

小程序云开发是无服务架构的一种,是小程序的基础能力之一。小程序云开发不仅组合了腾讯云团队的云函数、数据库和文件存储等服务,而且为开发者供给了较完整的服务器架构。使用小程序云开发完成闭环后,普遍情况下,业务能省去不少的开发部署和维护成本。

剔除业务逻辑,日志、监控、告警、数据安全、系统性能都是我们日常开发中须要完善的根本能力。如若业务增长需要扩容,数据信息的统计以及从前端到后台长链路的定位,都是需要手动费心的事情。

小程序云开发提供的数据库和文件存储既可以在小程序中直接使用,还可以通过云函数去调用。除了在项目中调用,还可以通过小程序云开发的控制台去管理。在开发完成后的运维阶段,通过小程序云开发配备的日志、监控和统计分析等运维功能,能使数据存储的安全性、服务的稳定性得到保障。

目前,小程序云开发适用的场景有以下几种:

1、个人开发者

服务器后台的服务搭建、数据保存、系统安全、服务器维护等工作都不在开发者的考虑范围，开发人员只需要专注于小程序的业务开发。因此，小程序云开发的模式对于个人独立开发人员来说是最合适的，只需要使用云开发提供的所有服务就可以单独完成小程序的开发。

2、工具类小程序

对于一些效率工具、文件工具、图片编辑工具等一些工具类小程序的开发，可以进行一些小数据的存储，同时配合 `http/https` 请求的方式就能完成完整的功能。

3、复杂逻辑的函数计算

前端的能力范围正在逐渐变大，在业务中也会免不了遇到一些复杂的逻辑运算。这时候通常会把逻辑放后台运算，因为前端的性能和兼容性都因环境和平台的改变而发生改变，同时单线程的模式也限制了一部分大数据的运算分析能力。

而今，我们可以把这些庞大的逻辑计算放在云函数中进行，内存分配、计算性能都是云开发来保障。我们甚至和一部分开源库结合，或是结合云开发提供的 AI 能力，这样一些做不到的功能就有可能做到。

1.3 论文结构概述

本文主要介绍了基于微信小程序云开发的音乐播放系统的开发，全文主要分为七个章节。分别是：

第一章绪论：主要介绍该课题的背景情况和研究现状。

第二章需求分析：主要通过功能需求、性能需求、可靠性和可用性需求以及户错处理需求，系统和完整的对该课题进行了需求分析，其主要涉及两类类型角色用户，完成播放列表、播放器、博客等相关功能的实现，还包括与云资源对接的后台管理系统的功能。

第四章相关理论和技术分析：该系统的前端使用组件化开发完成了小程序的前端部分，后端部分使用小程序云开发完成，调用小程序的原生接口实现了对小程序云数据库、云函数、云存储的操作。使用前后端分离架构完成了小程序的后台管理系统，云开发提供的 HTTP API 使开发的后台管理系统对接云开发中的资源。

第五章详细设计：通过数据库的逻辑设计到物理设计，以及功能设计的介绍，完整的描述设计过程和思路。

第六章系统实现与测试：通过对功能描述、功能截图和关键代码的撰写，完整的去展示该课题系统的实现细节和过程以及测试的过程。

第七章总结与展望：本章通过对该课题的研究和实现，概括了系统开发的过程，与开发的难点，该课题系统在未来的发展趋势如何如何。

第二章 需求分析

2.1 可行性分析

在需求分析的第一步进行可行性分析是尤为重要的，除了避免开发中途夭折，还可以针对技术的缺陷和难点进行分析，及时弥补缺陷，并有机会将难点避开。现针对本系统进行经济、技术和可操作性进行全面的分析^[3]。

1、经济可行性

对于音乐播放系统而言，应用市场上的音乐 APP 种类繁多，但每种 APP 上的音乐资源不一致，在某个系统中有的音乐在另一个系统却要付费播放，另外，每个系统的音乐版权也不一致。当要播放的歌曲在手机上已有的 APP 上无法播放时，想要体验另一款播放系统还要下载。除此之外，每款软件也有很多相同的地方，多次下载造成了资源的浪费。基于上述场景下的痒点，用户此时打开小程序音乐系统不仅提升了用户体验，还能增加歌曲的收听率，为平台和音乐家带来收益。软件开发的过程中使用的软件都是开源的，无需为软件付费，仅有的少量成本是云开发的资源在申请了三个月免费的基础版资源后，如到期或需要扩充资源可以付少量的费用继续使用。整个小程序的开发成本比较低，小于系统获得的收益，因此本系统有经济可行性。

2、操作可行性

本系统的界面简洁，操作简单，登录微信后搜索使用即可，无需安装等待。后台管理系统的界面清晰，功能一目了然，对于歌单、轮播图、博客的管理操作过程流畅，且技术上都可实现。因此本系统具有操作可行性。

3、技术可行性

本系统采用了软件工程开发的流程，需求分析通过 UML 进行了用例图、数据流图、E-R 图的设计和描述。前端开发使用了微信开发者工具，后端开发也是在微信开发者工具上进行的一体化开发。通过微信小程序或者云函数调用云数据库中的资源到前端并通过数据绑定显示。云数据库里的资源是从第三方 API 里调用进去的。通过云开发的 HTTP API 可以将云资源中的数据与后台管理程序进行对接，从而实现对数据的管理。小程序云开发目前是比较新的技术，但微信为开发者提供了全面的开发者文档，而且云开发比传统开发更为便捷。后台管理系统的前后端使用的是两个较为成熟的框架搭建的。因此本系统无技术难点。

2.2 系统用户角色设计

系统的用户角色设计通过考虑不同使用者的身份和需求，通过用例设计进行区分，为后续的设计和实现做基本的铺垫。

本课题分为两个角色类型用户，普通用户主要操作智能手机对系统的前端进行操作来满足需求，主要注重的是小程序端的基本功能；而管理员的身份是对数据进行管理和运营，开发中主要侧重将管理数据的过程变得更方便^[4]。

对普通用户来说，由于是基于微信平台，所以不需要登录本系统。微信登陆成功后，通过微信扫码或搜索进入本小程序即可播放歌曲，浏览博客等。用户在授权本小程序后，将可以使用本系统的发布博客、评论和分享博客等。

普通用户的主要用例如下：

1、播放音乐：用户登录微信后进入小程序后，用户可以浏览歌单歌曲列表，点击歌曲列表里的歌曲这个用例就会启动，可以暂停、快进、查看歌词，关闭小程序后用例介绍。包含一个扩展查看播放历史。

2、评论博客：用户进入发现页面，可以浏览博客列表，点击评论按钮并授权用户信息这个用例就会启动，编辑评论点击发送，当页面回到发现页面时表示评论用例结束。

3、分享博客：当用户进入发现页面，浏览博客列表时点击分享并授权用户信息按钮，该用例启动，选择好友并成功分享，最后回到发现页面该用例结束。

4、发布博客：在发现页面点击发布按钮并授权用户信息后，用例启动，选择图片并编辑文字，点击发布按钮，回到发现页面用例结束。包括两个扩展，分别是搜索并查看博客和查看“我的”博客。

本系统的前台用户用例图，如图 2-1 所示

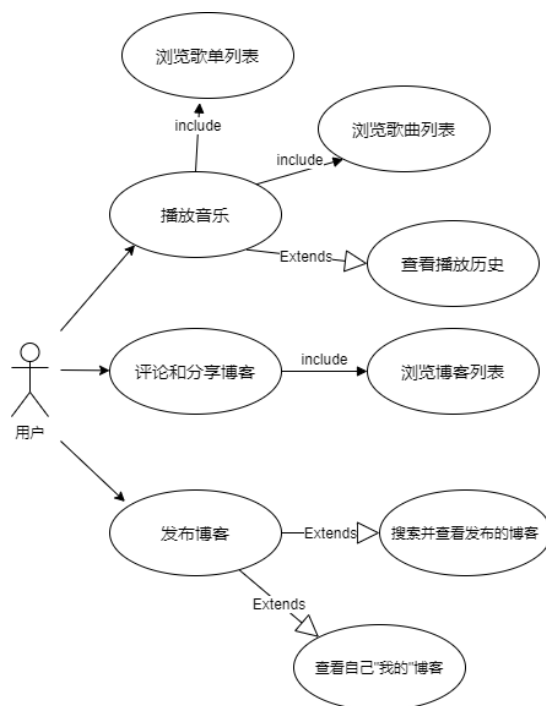


图 2-1 前台用户用例图

Figure 2-1 Foreground user use case diagram

对于后台的管理者，登录系统后，主要行使的功能为管理系统和云资源对接数据，包括对数据的删除、编辑和上传，小程序端的数据也会实时变化。后台使用者的主要用例包括浏览歌单数据、浏览轮播图数据、浏览博客数据。

1、浏览歌单数据：管理员登录系统后进入歌单管理页面即可浏览歌单数据，该用例包括两个扩展用例，分别是编辑歌单详情和删除歌单。

2、浏览轮播图数据：管理员登录系统后进入轮播图管理页面即可浏览轮播图数据，该用例包含两个扩展用例，分别是删除轮播图和上传轮播图。

3、浏览博客数据：管理员登录系统后进入博客管理页面即可浏览博客数据，该用例包含扩展用例，即删除博客。

后台管理者用例图如图 2-2 所示

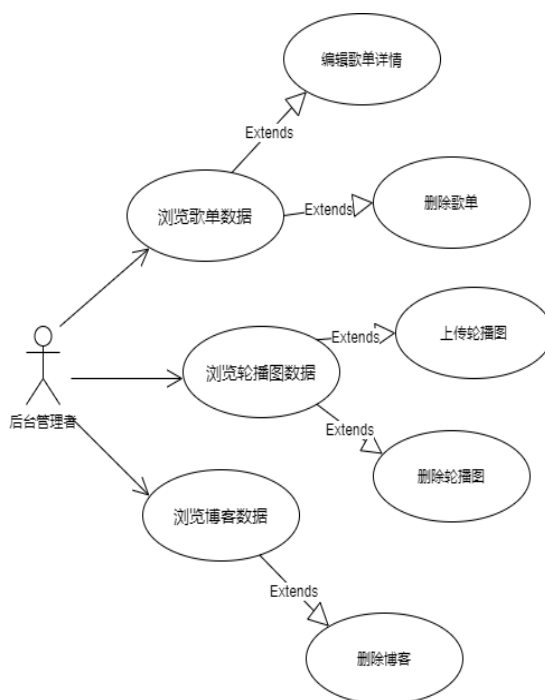


图 2-2 后台管理者用例图

Figure 2-2 Background Manager Use Case Diagram

2.3 系统功能需求分析

本系统的主要围绕以上两种角色的需求来进行业务的设计。前端主要满足普通用户的听音乐与发表博客这两个业务需求，后台管理者主要满足对于歌单的修改、博客的管理、轮播图的增减替换的需求。两种角色的所有需求构成了本系统的前台业务功能的需求。

一、前台用户的功能需求

为了满足前台用户的业务需求，本系统主要有以下功能：歌曲列表，播放器，发布博客，评论博客，分享博客，查看博客正文。具体功能需求如下：

1、歌曲列表

用户查看歌曲列表有两种模式，第一种是从主页的歌单列表选择歌单进入歌曲列表，歌曲列表应该显示歌单的详情和歌曲的基本信息。第二种是在“我的”页面的最近播放歌曲，进入可查看最近播放的歌曲，点击歌曲也可以播放。

2、播放器

播放器是用户在点击歌曲后进入的页面功能，小程序自动播放歌曲，并且有查看歌词、暂停播放、播放下一首或上一首、拖动歌曲进度条等功能。

3、发布博客

在“发现”页面，如没授权用户信息，在进入发布博客前询问用户是否授权用户信息。授权用户信息后，可以编辑文字、选择图片并发布。发布后的文字和图片会显示在“发现”页面，一同显示的还有发布时间、发布者微信昵称、微信头像。

4、评论博客

在博客组件的底部，点击评论，输入评论，会显示评论成功或者评论失败。发送的评论在用户点击博客组件触发博客的正文和评论列表的页面显示，新发送的评论会正常显示在博客评论下方，并且显示发送评论的内容、时间、发送者头像和昵称。

5、分享博客

点击博客组件的底部分享按钮，可以选择微信好友并将博客链接发送给好友，然后会显示分享成功或者发送失败。好友会受到链接，点进链接显示的是博客的正文页面。

二、管理员的功能需求

管理员的业务需求就是对小程序的数据进行管理，将一些违规的内容进行删除或编辑，或者上传需要上传的数据。通过后台的操作，可以间接改变数据库中的数据，显示在前端的数据也会随之改变。具体功能需求如下：

1、浏览并管理歌单

在歌单管理页面显示所有歌单的封面、名称、描述和操作。对于歌单的管理主要包括编辑和删除。管理员对一些非法的歌单可以进行编辑和删除。当歌单被删除时，该歌单里的所有歌曲也会被删除。点击编辑时主要可以编辑歌单的名称和歌单的描述。

2、浏览并管理轮播图

在轮播图管理页面显示所有轮播图的图片和操作。对于轮播图的管理主要是上传和删除。点击上传可以上传任意图片格式的文件，并最终会在小程序的主页上显示。当删除轮播图，小程序主页的轮播图上就不再显示该轮播图。

3、浏览并管理博客

在博客管理页面显示所有博客的内容、发布者昵称和操作。对于博客的管理主要是删除博客。当管理员删除博客时，博客里的评论数据也会一同删除。

功能模块图如图 2-3 所示。

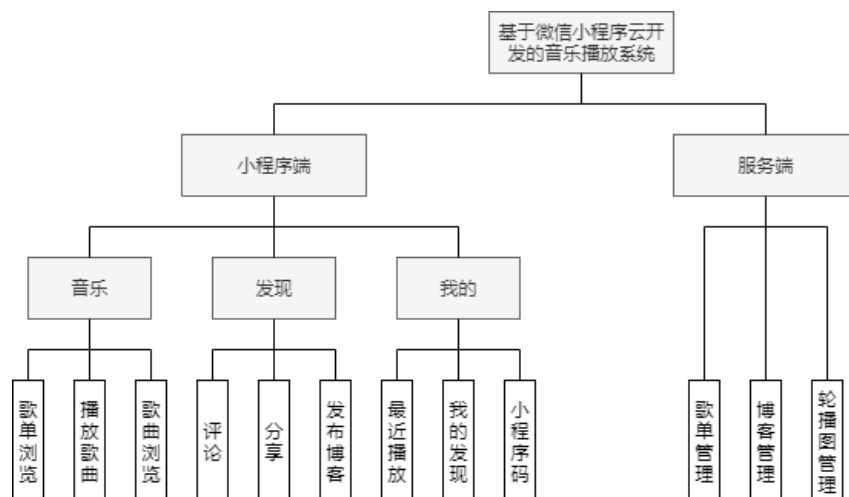


图 2-3 功能模块图

Figure 2-3 Function Module Diagram

2.4 系统数据流图设计

数据流图有三层，顶层是最基本的数据流图，后面每一层都是前一层的进一步细化。

数据流图的作用是展示数据流动的特点，本系统数据流图如下所示。

图 2-4 是系统顶层数据流图。这一层用户将请求提交给系统，系统返回请求的内容，内容包括用户想获得的资源或功能。后台管理员向系统提交管理维护请求，系统将数据或功能返回给管理员，管理员便可对系统的数据资源进行管理。

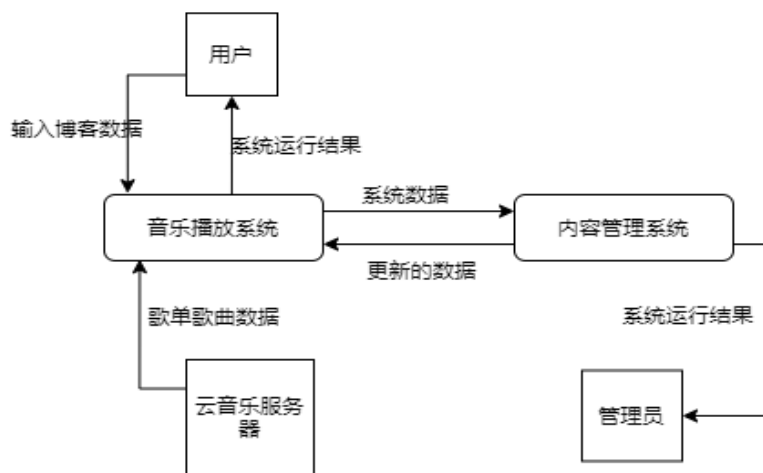


图 2-4 顶层数据流图

Figure 2-4 Top Stream Graph

图 2-5 是系统的第 0 层数据流图。这一层将用户与系统的交互这一处理模块分为若干功能，分别是博客功能、音乐功能。管理员与系统交互的模块分别是歌单管理、博客管理。

用户查询歌单信息，将请求发送到歌单查询功能模块，功能模块再调用歌单列表查询到相应的信息。歌单列表里包含若干条歌曲信息，包括歌曲的名称等。再由歌单查询传递到歌曲查询，最终查找到相应的歌曲列表。进入歌曲列表后接着进入歌曲收听模块，访问歌曲的 url 播放地址，就可以进行歌曲的收听，也可以收听播放记录里的歌曲。

用户查询博客模块进入博客列表，就可以查看到所有的博客信息。进入博客发布、评论模块可以更新博客的列表。

后台管理员管理歌单列表、博客列表，这些管理模块都是和相应的数据库里的数据表相链接的。

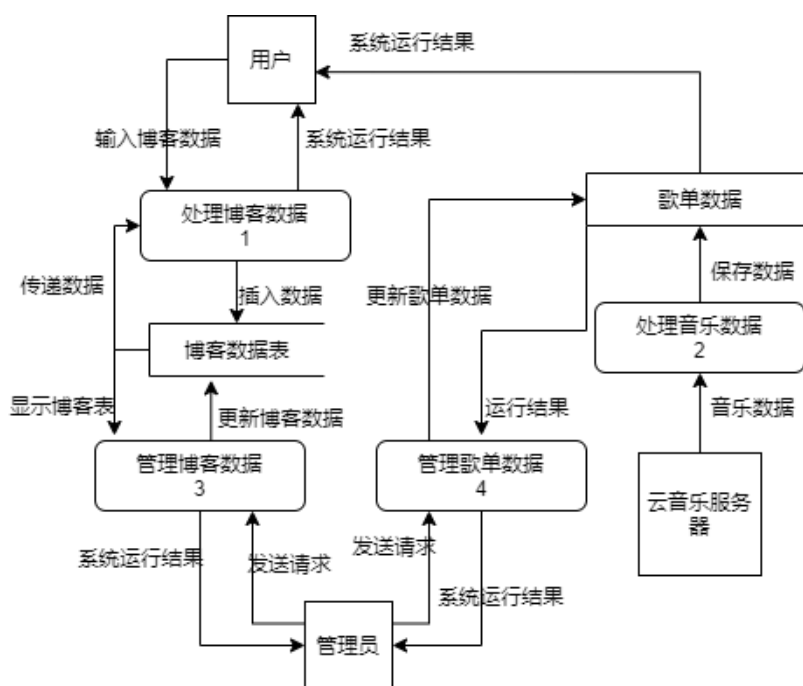


图 2-5 第 0 层数据流图

Figure 2-5 Layer 0 data flow diagram

将第 0 层数据流图进一步细化就形成了第一层数据流图。本系统的发布博客功能的第一层数据流图如图 2-6、图 2-7、图 2-8、图 2-9 所示。

发布博客的数据流图从用户确认授权开始，包括输入内容、选择图片，完成这一系列的操作后就可以发布博客，并在同时更新博客列表的数据信息。管理员主要对博客的列表数据进行管理，如发布内容不符合要求就可以删除该条博客。

第一层数据流图分为 4 部分，如下图所示。

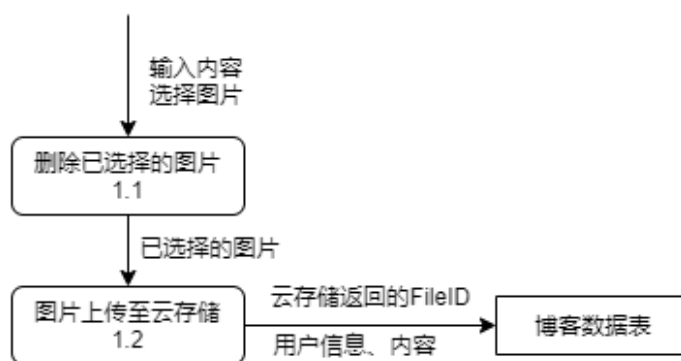


图 2-6 第一层数据流图（图一）

Figure 2-6 First Layer Data Flow Graph (Figure 1)

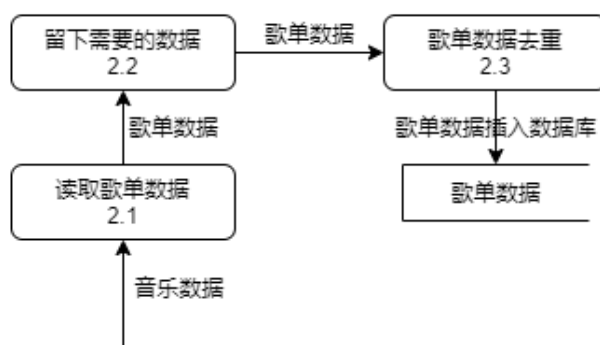


图 2-7 第一层数据流图（图二）

Figure 2-7 First Layer Data Flow Graph (Figure 2)

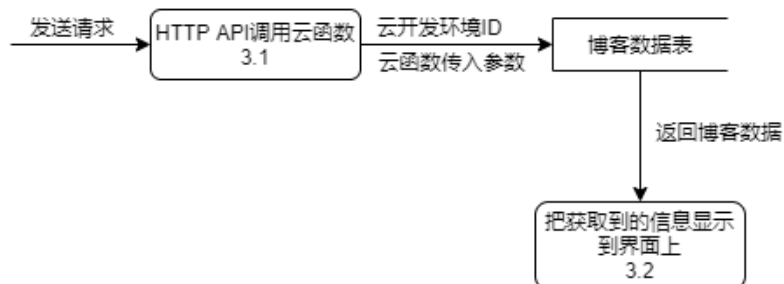


图 2-8 第一层数据流图（图三）

Figure 2-8 First Layer Data Flow Graph (Figure 3)

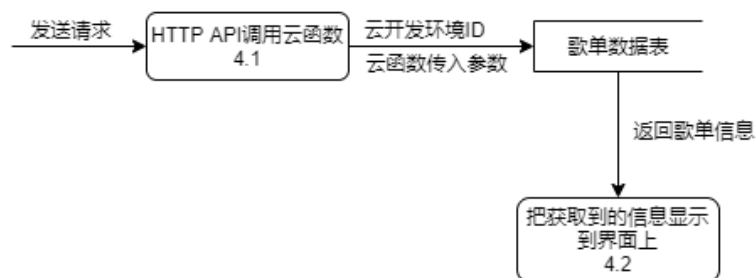


图 2-9 第一层数据流图（图四）

Figure 2-9 First Layer Data Flow Graph (Figure 4)

2.5 系统性能分析

在软件运行时，性能上的缺陷更可能导致系统的崩溃。因此一个系统不仅要有完备的需求功能，还要将系统的性能进行详尽的分析。好的性能能使系统的运行更加流畅，通常情况下系统的性能和功能的需求是相辅相成的。现通过系统的性能需求进行分析。

1、易用性

因为系统基于微信平台，查找小程序要通过搜索或者扫描二维码，过程比较繁琐。保证界面的良好设计以及功能主体的简化有助于提高系统的易用性，也可以减轻用户的厌倦感。本系统中，按钮摆放的位置明显，界面显示的功能明显，设计简洁，这些都有助于提升系统的易用性。

2、可靠性

可靠性是业务开展的保证和基础，是系统运行的第一要求，若无法保证功能稳定可靠地运行，使用者将对平台产生不信赖的感觉，这种感觉会严重影响用户对产品的评价，从而导致业务难以继续进行。在系统的产品设计、开发过程、运维测试过程都有必要遵循统一的原则。产品设计阶段要考虑到系统反应的速度慢等特殊情况，在开发过程尽量使用简洁的代码，减少冗余的代码，在运维测试阶段对所有的功能都要进行测试，保证可靠性和稳定性。

3、维护性

维护性是一个比较重要的性能需求，在系统的实现过程中使用分离层级，尽量减少代码的耦合，在前端使用组件化的开发思想就能较好的实现组件分离的目标。对于不同的组件、不同的功能使用不同的模块，这样如果要修改某一个组件，其他组件和功能便不会受到影响。

第三章 相关理论和技术分析

3.1 微信小程序开发技术

总所周知，小程序是无需下载安装即可使用的“微、小、轻”型的应用，它实现了应用的“唾手可及”，用户通过微信的搜一搜入口或者扫一扫即可打开应用。也体现了“用完即走”的理念，用户不用担心应用占用内存的问题，因此它的开发程度也较为简单。但是，微信小程序从技术和理念上都不是一个新事物，从技术上来说，它借鉴了 React Native 的一些概念，定义了一套微信自有的组件并根据运行环境的不同将这些组件转化为平台可运行的组件^[5]。

“对性能要求不高的”、“简单的”、“低频的”使用小程序开发会更加适合。“对性能要求不高”是因为小程序寄生在微信这个原生 APP 里，注定是不能开发高复杂度的游戏应用。“简单”是指应用的业务逻辑单纯，通常用户使用的服务的目的明确，不会有太复杂的业务。“低频”是小程序的另一个使用场景的特点，比如在微信聊天时突然手机没话费了，这时候不用去另一个应用，点开小程序就能更方便地充值^[6]。

3.2 小程序云开发技术

“云开发”的理念就是提供一套几乎完整的 Serverless 服务来减少开发者的成本，小程序·云开发是其中的一个出口。后端和运维概念在云开发里是被弱化的，云开发的原生云端和微信服务功能为开发者提供完整的支持，使用平台提供的 API 进行主要业务开发，项目即可快速上线和迭代。无需搭建服务器，即可使用云开发的云端能力开发微信小程序^[7]。云函数，云存储，云数据库是微信云开发的三大基础能力。

3.2.1 云函数

云函数就是一段运作在云端的相当于后端的代码，而且不需要管理服务器，只需要在开发工具当中编写代码，通过一键的上传部署就能运行这些代码。云函数运作在 Node.js 环境下。使用专门用于云函数调用的 API，开发者可以在云函数内使用 wx-server-sdk 提供的 getWXContext 方法获取到每次调用的上下文（appid、openid 等）。

3.2.2 云数据库

云数据库使用的是面向文档的 JSON 数据库，JSON 数据库中有多个集合，每个集合是一个 JSON 数组，数组中的每条记录可以看出一个 JSON 格式的对象。它最大的特点是它的查询语言能力非常强大，不需要掌握 SQL 语句就可以掌握数据库，进行增删改查的操作。

云数据库的内容直接在客户端调用，也可以在客户端调用云函数，然后在云函数中处理业务逻辑，之后在云函数中调用云数据库。这是因为数据库的 API 分为小程序端和服务端两种，数据库的 API 具有删除、查询、增加、修改的基本功能。

3.2.3 云存储

云存储是云开发免费给用户使用的一块存储空间，开发者可以在小程序端和云开发端通过 API 使用云存储功能，云存储中的文件可以通过云端直接上传，或者下载云端的文件，需要带权限管理。

第四章 系统设计

4.1 系统物理设计

物理设计分为五层，从上到下分别是系统的应用表现层、系统的网络层、系统功能层、系统技术层和系统数据库层。本系统的物理设计如图 4-1 所示。

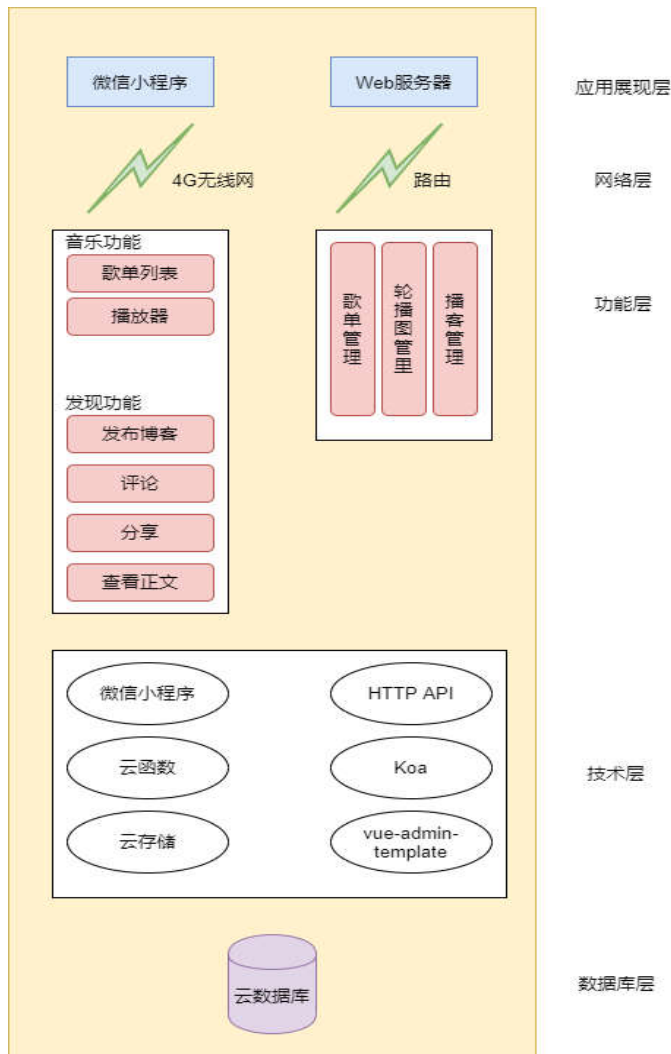


图 4-1 系统总体设计图

Figure 4-1 Overall System Design

应用表现层分别是 Web 服务端和小程序前端。服务端是通过 Web 浏览器登录并进行访问的，小程序前端使用手机等移动端设备进行展现，这两层通过 HTTP 网络协议与网络层链接。

系统的功能层进行细化和模块化处理后，有以下模块：

客户端有音乐模块和发现模块。音乐模块包括歌单列表功能和播放器功能，发现模块包括发布博客功能、评论功能、分享功能和查看正文功能。

后台管理端包括歌单管理功能、博客管理功能和轮播图管理功能。歌单管理包括对歌单的编辑、删除，博客管理包括对博客的删除，轮播图包括上传轮播图和删除轮播图。

下面一层技术层指的是系统实现过程需要的技术实现手段。前后端的实现都使用的是集成的微信小程序开发工具，后端使用了云函数、云数据库、云存储等小程序云开发功能。前后端之间的数据传递是通过云函数进行的。后台管理系统采用的是 VS Code 开发工具。使用了开源的 vue-admin-template 框架和 Koa 框架搭建了后台的前后端，后台与前台的数据连接是通过小程序云开发的 HTTP API 进行连接的。

数据库层就是云开发的云数据库，是一种 NoSQL 型数据库，这种数据库可以很方便进行数据的查询^[8]。

4.2 系统功能设计

前面已经对系统的功能需求进行了分析并给出了功能模块图，本小节将对系统的功能进行详细的流程设计。

对于前端使用者，功能模块主要包括音乐模块、发现模块和个人主页模块。在音乐模块中，可以进行歌单浏览、歌曲详情浏览、歌曲播放。在发现模块，主要可以进行评论博客、分享博客和发布博客。在个人主页模块主要有最近收听、我的发布、小程序码功能。

歌单浏览、歌曲详情浏览、音乐播放功能都不需要进行用户授权。用户进入小程序后点击歌单，再点击歌曲就可以直接进入音乐播放功能模块中。

评论博客是在进行用户授权后才可以在文本框内输入文字并发送，评论成功后，评论的结果将会在博客的详情页面显示。在用户授权后就可以将任意博客分享给微信好友，微信好友在收到博客后，点击可以进入该条小程序的博客详情页面。

个人主页模块不需要用户授权就可以进入，其中的最近播放功能和我的博客功能都可以直接使用。最近播放功能是一个用户最近播放歌曲的列表，我的博客功能是用户咋发现页面发布的博客列表。

对于后台的管理员，功能模块包括歌单管理、博客管理和轮播图管理。管理员登录后台管理系统后，可以分别对歌单、博客、轮播图进行管理。

歌单管理包含歌单的删除、编辑。博客管理包含博客的删除。轮播图管理包含轮播图的上传与删除。

关于发布博客的流程图如图 4-2 所示。该功能在授权成功后正式进入创建博客阶段，选择图片或者不选择图片都可以完成博客的发布，默认一定要输入文字才可进行发布，否则会提示请输入文字。

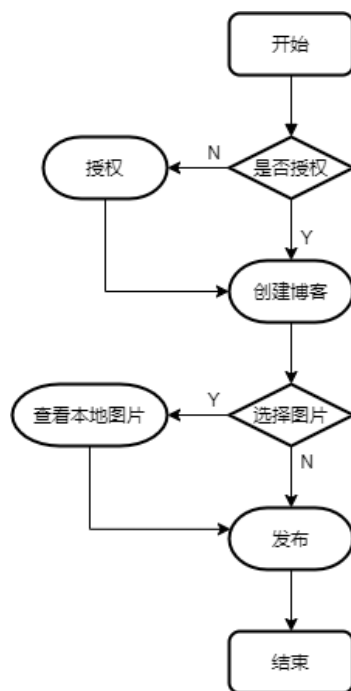


图 4-2 发布博客流程图

Figure 4-2 Publishing a blog flowchart

关于音乐播放的流程图如图 4-3 所示。该功能在进入小程序后就可以直接开始选择歌单和歌曲进行音乐的播放。

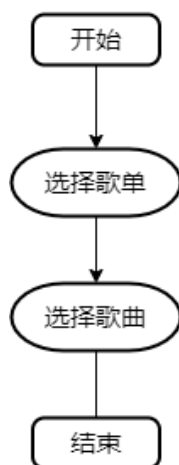


图 4-3 音乐播放流程图

Figure 4-3 Music Playback FlowChart

关于评论功能的流程图如图 4-4 所示。该功能在用户授权后才可以进行评论，最终评论成功的结果会显示在博客下方。

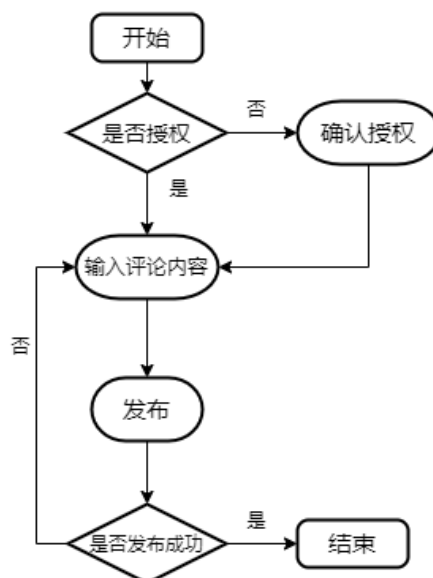


图 4-4 评论功能流程图

Figure 4-4 Comment feature flowchart

关于博客分享的流程图如图 4-5 所示。该功能在授权后可选择微信好友进行博客的分享。

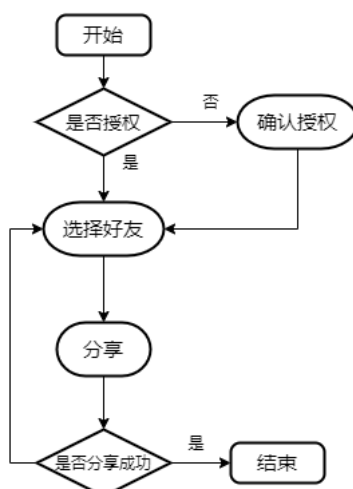


图 4-5 分享功能流程图

Figure 4-5 Share feature flowchart

图 4-6、图 4-7、4-8 是管理系统的歌单管理流程图、博客管理流程图和轮播图管理流程图，管理员可对小程序的数据和文件进行管理。

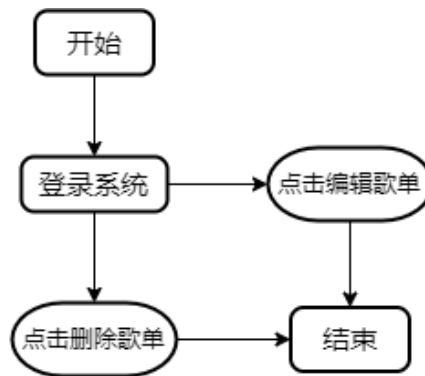


图 4-6 歌单管理流程图

Figure 4-6 Song sheet management flowchart

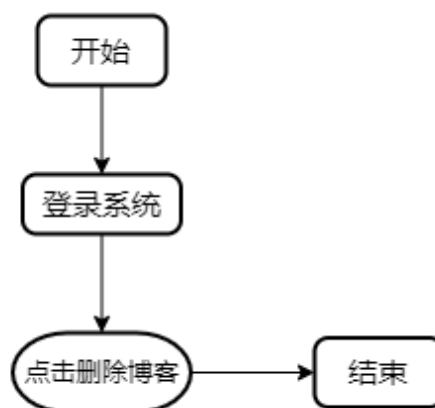


图 4-7 博客管理流程图

Figure 4-7 Blog Management Flowchart

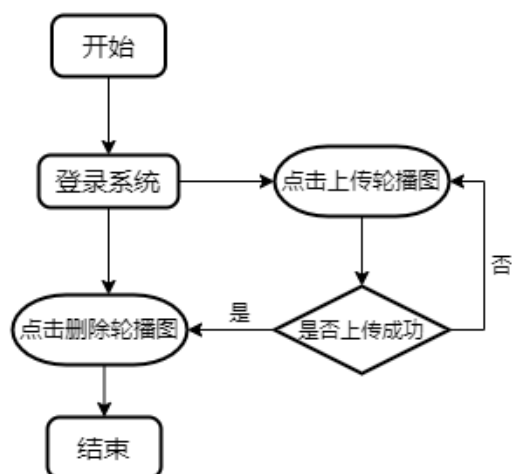


图 4-8 轮播图管理流程图

Figure 4-8 Carousel Management FlowChart

4.3 系统数据库设计

通过以上的需求分析和技术分析，我们下面将完成该系统的数据库设计。良好的数据库架构和精准的数据建模对于业务功能的实现和系统的工作效率有巨大的影响。本系统的数据库是基于云数据库的，云数据库是一种非关系型数据库，可以看成是 MongoDB 数据库，由于本系统有大量数据库的查询，使用云数据库可以本系统查询效率的问题^[9]。

4.3.1 系统数据库结构设计

本系统主要涉及到几个信息模块，包括音乐模块，博客模块。

音乐模块（歌单、歌曲列表）表的设计如下图 4-9 所示。

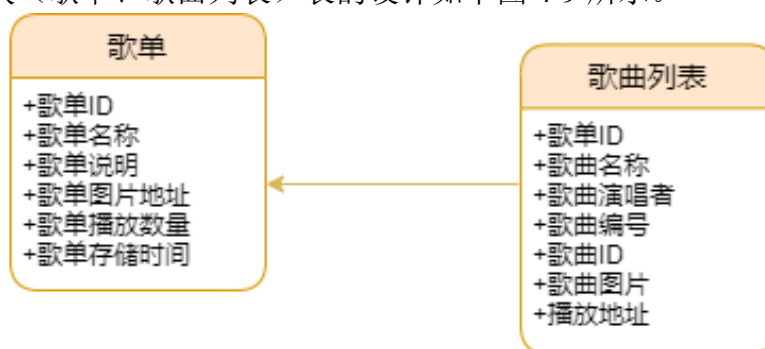


图 4-9 音乐模块结构设计

Figure 4-9 Structure design of music modules

博客模块（博客、评论）表的设计如下图 4-10 所示。

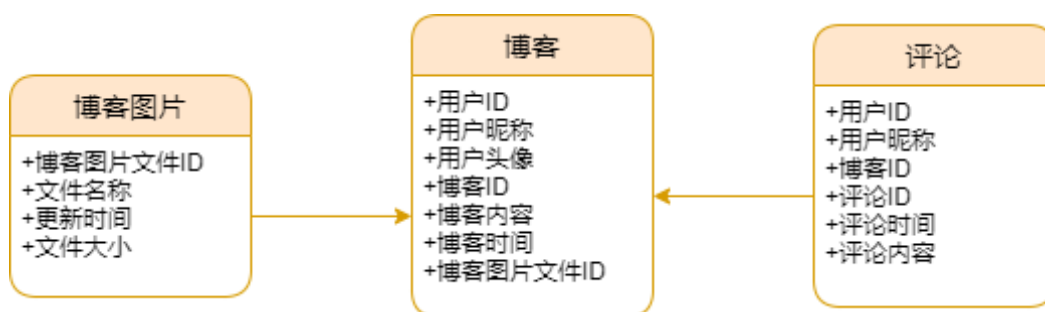


图 4-10 博客模块结构设计

Figure 4-10 Blog Module Structure Design

4.3.2 系统数据库物理设计

以上是本系统数据库表概要结构图，下面给出了部分逻辑表的具体字段及数据类型的详细描述。系统中每个数据库表的详细说明如下：

（1）获取用户信息

在不同的场景下获取用户信息有不同的方式。在界面上显示用户信息，只需在 wxml 文件下通过小程序的开放能力组件 `open-data` 就可以获取到用户的微信头像、昵称等，这种情况下不能在 js 文件下获取用户信息因此不能进行存储，且不需要用户授权。在 js 文件下，如果用户已经授权才可以通过 `wx.getUserInfo` 获取用户信息，但无法获取 `openid`。在云开发下获取用户 `openid` 的方法如下，在小程序中设置按钮和点击按钮处理函数，处理函数直接通过调用云开发自带的 `login` 云函数，`login` 云函数返回值里就有 `openid` 的信息。

（2）博客表

表 4-1 是系统的博客表，博客表包括博客 ID、博客内容、博客发布时间等字段。还包括与博客关联的用户信息的字段。

表 4-1 博客表

Table 4-1 Blog Table

字段名称	数据类型	说明
openid	String	博客用户的 openid
id	String	博客的唯一 id
nickName	String	博客用户的昵称
content	String	博客内容
createTime	Date	博客发布时间
img	Array	博客图片文件 ID

（3）评论表

表 4-2 是系统的博客评论表，评论表包括评论 ID、评论内容、评论发布时间。还包括与评论关联的博客的 id 和关联的用户的信息字段。

表 4-2 评论表

Table 4-2 Comment Table

字段名称	数据类型	说明
id	String	评论唯一 id
openid	String	评论用户 id
blogId	String	评论关联的博客 id
content	String	评论内容
createTime	Date	评论发布时间
nickName	String	评论用户昵称

（4）歌单表

表 4-3 是系统的歌单表，歌单表包括歌单 id、歌单名称、歌单说明、歌单存储时间、歌单图片地址等字段。

表 4-3 歌单表

Table 4-3 Song sheet table

字段名称	数据类型	说明
id	String	歌单唯一标识
copywriter	String	歌单说明
createTime	Date	歌单存储时间
name	String	歌单名称
picUrl	String	歌单图片地址
playCount	Number	歌单播放数量

第五章 系统实现与测试

5.1 开发环境的搭建

小程序云开发环境搭建有以下几个步骤：

- 1、登录微信开发者平台后，在官方网址下载并安装微信开发者工具，由于本系统是在 Windows 平台下开发的，所以下载的是稳定版的 Windows64 的安装包。下载完成后双击安装包安装完成。
- 2、安装完成后，双击微信开发者工具，用关联过的微信号扫码进入并新建项目就会出现如图 5-1 所示的项目新建界面。如图所示后端服务有两种，如果只开发的小程序的前端可以使用第二种方式，即不使用云服务。本系统使用的是小程序的云开发，所以选择第一种后端服务，即小程序·云开发。



图 5-1 项目新建界面

Figure 5-1 Project New Interface

- 3、创建完云开发小程序后开通点击进入开发面板里的云开发按钮进入小程序云开发界面开通云开发，如图 5-2 项目开发界面所示，开通完云开发后点击云开发按钮就进入了云开发的控制台如图 5-3 所示。点击开通按钮后，创建云开发的环境，每个环境会独立配备一套独立的云开发资源，包括云存储、云函数等，云开发资源的配额可以自己选择，也可以使用默认基础配额。

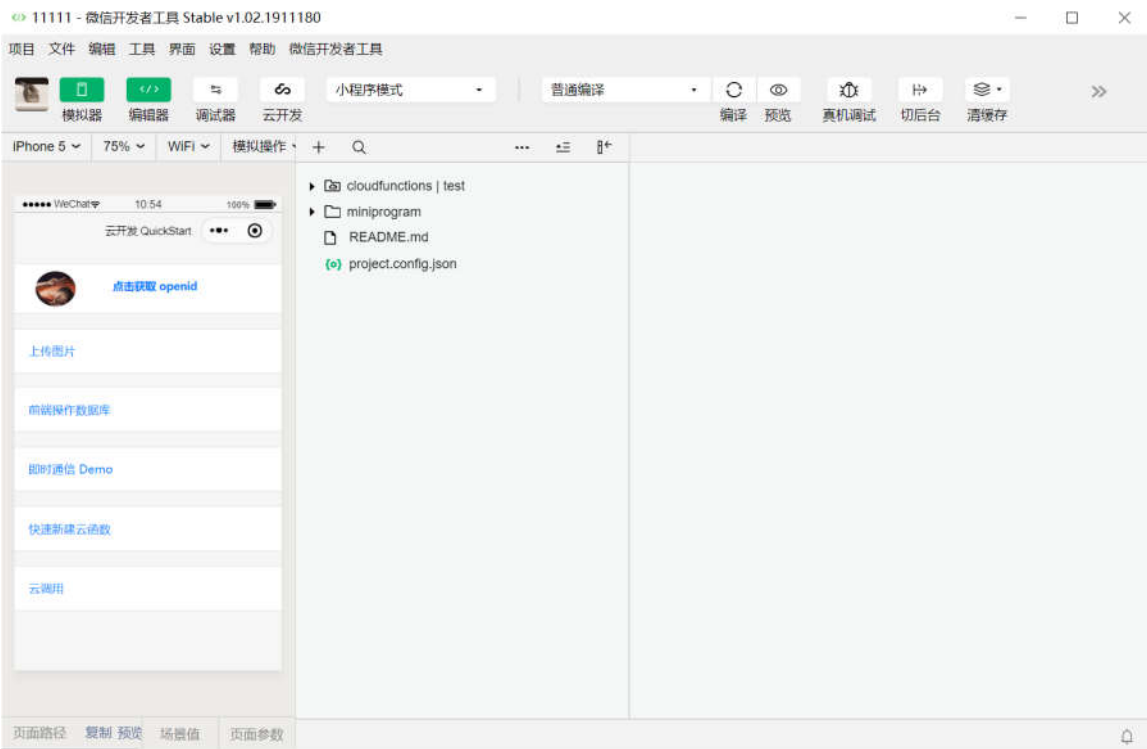


图 5-2 项目开发界面

Figure 5-2 Project Development Interface

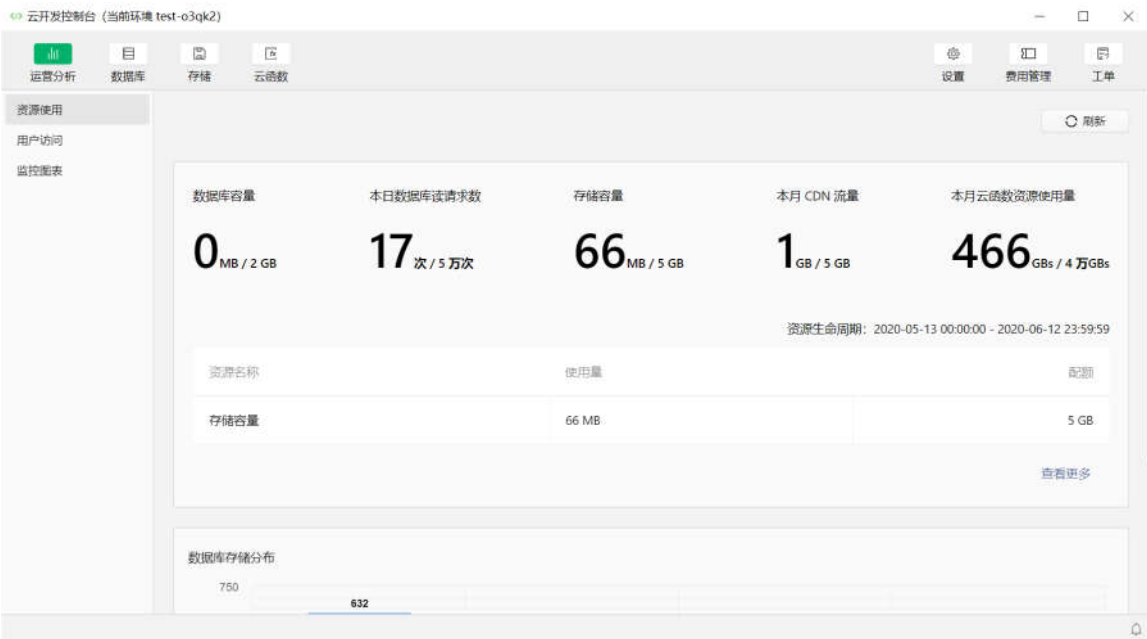


图 5-3 云开发控制台界面

Figure 5-3 Cloud Development Console Interface

4、点击云开发进入云开发控制台，首页运营分析页面可以看到资源使用情况、配额资源以及用户访问情况等。数据库页面可以看到项目中创建的所有集合以及每个集合里的记录。云存储页面可以查看存储在其中的文件，云函数页面可以查看在小程序端创建的云函数。

5.2 播放功能

5.2.1 播放功能实现

播放列表用来显示音乐歌单信息，后台会自动从网易云音乐 API 中更新歌单信息。前台用户通过搜索微信小程序，并成功授权后，就可以进入系统的主界面。系统的主界面如图 5-4 所示。



图 5-4 主界面和歌单列表

Figure 5-4 List of main interfaces and song lists

主界面的顶部是一个轮播图，轮播图滑动的间隔时间为两秒，持续时间为一秒。轮播器由微信小程序内置组件 swiper 实现轮播功能并设置轮播器属性。只能放置 swiper-item 组件的 swiper 组件是一个滑块视图容器。每个 swiper-item 组件通过 block 包装元素来控制其属性，为 swiper-item 绑定轮播图数组，使用 wx:for 控制，就可以重复渲染数组中的各项数据^[10]。

轮播图功能实现的代码如下：

```
<swiper indicator-dots="true" autoplay="true" interval="2000" duration="1000">
  <block wx:for="{{swiperImgUrls}}" wx:key="url">
    <swiper-item>
      <image src="{{item.url}}" mode="widthFix" class="img">
    </image>
  </swiper-item>
</block>
</swiper>
```

主界面的底部是一个 Tab 栏，其实现过程是在 app.json 文件中设置 tabBar 属性，并设置图标 png 图片的路径。有三个 Tab，分别是音乐、发现和我的。点击不同的图标将跳转到不同的页面中。

主界面的内容部分的歌单组件由自定义组件封装完成。在组件文件夹下创建一个自定义组件 m-playlist，每个组件包括四个文件，分别是 wxss、wxml、js、json，在组件中可以创建需要的任何结构、样式。创建好组件后，在主界面页面的 json 文件中引入该组件，通过组件的名称可以直接在 wxml 中引用该组件。

歌单详情内容来自网易云音乐 API，从其服务器端取出真实的歌单详情数据，并且将其数据写入到云数据库中。通过在云函数中设置定时触发器发送请求，每天定时取出所需的歌单图片信息、歌单描述信息、歌单收听信息的数据，这样能保证当前取到的都是最新的。并且将取出的数据与当前数据进行比较，保证取出的数据都是最新数据，最后把之前数据库中没有的数据写入数据库当中。其中使用到了 JavaScript 的异步操作，实现过程如下：

- 1、创建云函数发送请求，安装完 Node.js 后，在终端使用 npm 命令安装第三方库 request-promise。安装完成在云函数中引入 request-promise 以及需要的音乐接口地址，该接口可以获取当音乐的歌单信息。
- 2、在云函数中就可以使用 request-promise 中的函数 rp 向该地址发送请求，因为请求需要一定的时间，所以 rp 函数要使用 await，等待请求全部完成后才能将请求结果返回到云函数控制台。
- 3、返回值是一个 String 字符串型的数据，字符串类型的数据无法进行操作，所以使用了字符串转化为对象的 JSON.parse() 方法将请求结果 res 转化为对象。返回值

里只需要 `result` 数组，数组里每一个对象代表每一个歌单信息。所以要把请求的返回值 `res` 通过语句 `JSON.parse(res).result` 转化并获取我们需要的歌单信息数组。

4、将获取到的数组插入到云数据库当中，需要循环遍历数组中的每一条数据，依次插入到云数据库中。在云数据库中创建一条新的空集合，相当于传统数据库中的表，云函数中初始化数据库后，在循环中使用 `db.collection()` 获取到新创建的集合，通过 `add()` 的方式将获取到的数组插入到集合中。等待一条插入完才可以插入另一条，所以这一步也是一个异步操作，因此需要使用 `await`。

歌单信息每次被调用很可能有重复的，所以需要进行歌单的去重工作，数据库中已有的数据不应该再次插入数据库当中。操作过程如下：通过 `db.collection.get()` 方式获取到当前集合中的所有数据，使用 `for` 循环与从接口中获取到的数据进行比较，当前集合中没有的数据插入集合，已存在的数据不再插入集合。

通过 `db.collection.get()` 方式在云函数端获取云数据库中数据每次只能获取 100 条，但是接口获取到的数据每天都是不一样的，这样数据库中的歌单信息也会越来越多，之后很可能会突破 100 条，这样就不能更新程序中的歌单数据了。突破 100 条数据后，可以分次去取，每次取 100 条或 100 条以内的数据，将分次取出的数据拼接在一起，这样就能取出所有数据。

为了实现程序自动触发云函数获取最新的歌单数据，需要给云函数添加一个触发器，在云函数的文件夹下创建一个 `config.json` 文件里面配置每天的某个时间点定时触发云函数，就能实现自动更新数据库中的数据。

点进任意歌单组件，显示的是当前歌单对应的的歌曲详情信息，如图 5-5 显示。歌曲列表由自定义组件完成。



图 5-5 歌曲列表

Figure 5-5 Songs List

5.2.2 播放功能测试

本功能测试的前置条件是用户已经进入小程序，播放功能测试主要测试系统的功能性与界面显示。

在音乐界面选择任意歌单，然后进入相应歌曲列表，选择任意一首歌曲，测试界面是否会跳转到相应的播放界面，并播放相应歌曲，点击播放界面中间的图片，界面跳转至歌词界面。

歌曲的快进与倒退、暂停与播放功能是否正常，歌词与进度条是否有联动。

经过测试，系统满足最初的需求分析和功能要求，而且界面跳转正常。

5.3 博客功能

5.3.1 博客功能实现

用户点击发现 tab，页面会跳转到博客功能界面并返回所有数据库中的博客数据表，博客按照时间逆序排列，最晚发布的博客在最前面。本博客列表的界面如图 5-6 所示。

点击写博客发布按钮跳转到编辑页面，会弹出是否授权的弹窗询问是否授权，用户确认授权后输入文本添加图片发布博客，发布成功则在博客表中添加一条博客数据，并跳转至博客列表页面重新加载所有博客列表。



图 5-6 博客列表

Figure 5-6 List of Blogs

在未授权的状态下点击博客卡片下方的评论按钮，会弹出授权框，用户确认授权后会进入如图 5-7 所示评论界面，输入评论内容并发送成功后跳转至发现页面首页，评论信息存储在云数据库的评论表里。点击博客正文，用户可以看到所有评论。

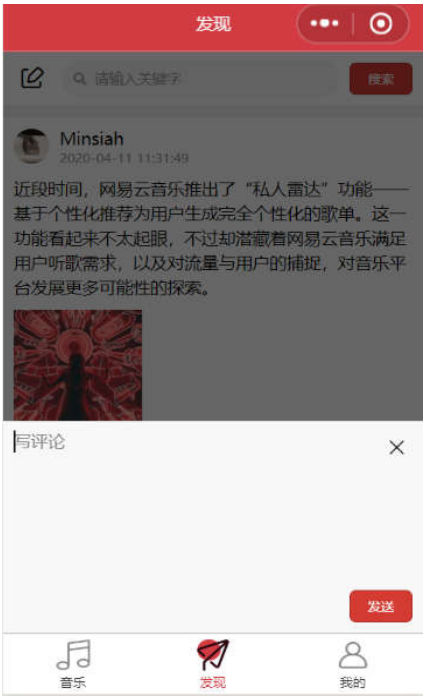


图 5-7 评论列表

Figure 5-7 Comments List

5.3.2 博客功能测试

用户在登录小程序的情况下点击正确的菜单栏，出现博客列表，博客列表会按时间先后依次排列，最新发布的博客在第一个，且之后的博客无错位显示。

对于发布博客功能的前置条件是用户登录小程序并成功授权。进入编辑博客页面后能正常输入文字并添加本地图片，只输入文字不添加图片可以正常发送，只添加图片将提醒输入文字后才可以发布博客，最多添加 9 张图片。

对于博客评论的功能的前置条件是用户进入小程序发现页面主页，或者进入博客正文页面。评论内容限制在 150 字内，发送并评论成功后跳转至原页面，点击博客正文评论内容按时间逆序排列显示，且评论对应用户头像、昵称和发送时间。

测试结束，系统能满足这些功能，数据无误显示，且页面正常跳转。

5.4 歌单管理功能

5.4.1 歌单管理功能实现

管理员登录后台后，可以对歌单进行管理，包括编辑歌单的名称和歌单的描述，也可以将歌单进行删除。编辑后的歌单会在小程序前台显示编辑后的内容，若删除歌单，歌单便不会显示在前台，并同时删除了歌单内的歌曲信息。歌单管理功能页面如图 5-8 所示。



图 5-8 歌单管理界面

Figure 5-8 Song sheet management interface

5.4.2 歌单管理功能测试

本功能测试的前置条件是管理员登录成功，并且选择了歌单管理模块。歌单管理页面会显示所有歌单信息。点击编辑后可以进入歌单编辑页面，更新成功后会正常显示在管理系统和小程序前台。点击删除，会询问是否确定删除，确认删除歌单后歌单会从管理页面和小程序前台消失，其他歌单正常排列，同时数据库中的歌单数据也被删除。

经测试，可以正常编辑歌单信息，歌单能正常被删除，且在系统中不再出现，并且不影响其他歌单的显示。

5.5 轮播图管理功能

5.5.1 轮播图管理功能实现

轮播图管理功能如图 5-9 所示。管理员可以对轮播图进行查看，也可以对轮播图进行删除与上传的操作。



图 5-9 轮播图管理界面

Figure 5-9 Carousel Management Interface

5.5.2 轮播图管理功能测试

对于轮播图管理功能的测试的前置条件是管理员登录系统并且进入了轮播图管理模块，本功能主要测试轮播图的删除和上传。经测试，点击删除后，轮播图将从管理页面中删除，并且不再显示在小程序前台，小程序前台中的其他轮播图能正常显示，轮播功能正常，其他图片显示正常。点击上传可以选择本地图片进行上传，上传成功后能正常显示在小程序前台轮播图组件中，符合需求。

5.6 博客管理功能

5.6.1 博客管理功能实现

博客管理功能页面如图 5-10 所示。可以对后台的博客进行管理，包括对博客的查看和删除。若删除某条博客，博客以及博客关联的评论内容都会在数据库中删除，并且不会显示在小程序前台。

Dashboard / 博客管理

	内容	发布人	操作
1	近段时间，网易云音乐推出了“私人雷达”功能——基于个性化推荐为用户生成完全个性化的歌单。这一功能看起来不太起眼，不过却潜藏着网易云音乐满足用户听歌需求，以及对流量与用户的捕捉，对音乐平台发展更多可能性的探索。	Minsiah	删除
2	后台系统测试	Minsiah	删除
3	枯萎🌿	Minsiah	删除
4	test	Minsiah	删除
5	测试刷新	wubba lubba dub dub	删除
6	猫	wubba lubba dub dub	删除
7	海洋	wubba lubba dub dub	删除
			删除

图 5-10 博客管理页面

Figure 5-10 Blog Management page

5.6.2 博客管理功能测试

本功能测试的前置条件是管理员已经登录系统并且进入了博客管理页面。博客管理页面会显示所有发布的博客，包括博客的内容和发布者。若进行删除操作将删除博客，包括删除博客的内容、图片以及和博客关联的评论。

经过测试，博客能正常删除，并且不影响其他博客的正常显示。

第六章 总结与展望

本文主要对音乐播放小程序系统进行了设计，针对了音乐模块、发现模块、以及我的模块具体分析。本系统的开发过程使用的是瀑布开发流程。使用了 UML 进行需求分析，制作了系统的流程图、数据流图、用例图等。前台使用的是小程序开发技术，后台使用的是小程序云开发。

能够实现基本需求，但仍然有些不足，例如：用户查看博客的评论只能点击进入博客的详情页面进行查看；用户只能通过歌单听歌，不能通过搜索功能或者选择歌手进行歌曲播放，选择歌曲的方式较为单一。以上不足之处需要对系统进行进一步的研究与扩展。

科技持续在变革，技术不停在变化，作为一个小程序要拥抱变化，也要拥抱未来。要使用小程序云开发提供的能力改建原有架构或许成本比较高，不过将其列入未来的可用方案也是可以的。

致谢

毕业论文完成之际，本科生涯也接近尾声，这几年有很多收获和成长。感谢每一位陪我度过大学时光的信息工程学院的所有尊敬的老师和亲爱的同学，因为有了你们的帮助和指导我才能顺利完成学业和毕业论文。

首先，诚挚感谢论文指导老师余立毅老师，本人的毕业论文在他的指导下才有了论文的选题方向，也是在他严格的监督和耐心的指导下完成了初稿以及最终稿等阶段，在此向余老师致以真诚的谢意。

感谢景德镇陶瓷大学 2016 级数字媒体技术的全体同学陪我度过的这一段难忘的历程和对我的帮助，以及朋友给我的精神力量，在我灰心丧气时给了我源源不断的鼓励，为我加油打气。

最后，感谢的是我的父母和我的家人，感谢他们在大学期间为我的所有养育和付出的心血，让我有机会接受大学教育并能够顺利完成毕业设计和论文。今天所有离不开他们的支持和亲情，他们是最坚实的前进的动力。

参考文献

- [1] 雷磊. 微信小程序开发入门与实践[M].清华大学出版社,2017.
- [2] 李莎莎. 基于 Serverless 的前端开发模式研究[J].电脑知识与技术,2019.
- [3] 金莹. 基于微信小程序的个人电台系统的设计与实现[D].吉林大学, 2015.
- [4] 邱郁惠. 系统分析师 UML 用例实战[M].机械工业出版社,2010.
- [5] 李哲 周灵. 微信小程序的架构与开发浅析[J].福建电脑,2019,第 12 期.
- [6] 邱月焯. 微信小程序的生命力[J].21 世纪商业评论,2018,第 12 期.
- [7] 张利香,甘发旺,簿建国. 基于微信小程序与云开发的成绩查询系统实现[J],内蒙古科技与经济,2019,第 13 期.
- [8] 陈绪. 基于 MongoDB 数据库的博客管理系统[D].河北农业大学, 2018.
- [9] 邹竞莹. Node.JS 博客系统的设计与实现[D].黑龙江大学,2016.
- [10] 白香芳,苏文芝. Swiper 组件在微信小程序开发中的应用研究[J].电脑编程技巧与维护,2019.