

**本科毕业设计说明书**

**基于J2EE的数字音乐销售网站**

DIGITAL MUSIC SALES WEBSITE BASED ON J2EE

学院（部）： 经济与管理学院

专业班级： 电子商务 16-1

学生姓名： 段会会

指导教师： 杨超宇（副教授）

2020年 5 月 24日

基于J2EE的数字音乐销售网站

摘要

电子商务作为互联网发展的产物丰富着我们的物质生活，是推动我国经济发展强有力的因素。在CNNIC发布第的44次《中国互联网络发展状况统计报告》中指出，至2019年6月，我国网络购物用户规模达接近7亿，占网民整体的75%。中国网民蕴含着巨大的消费潜力，在物质生活不断提升的今天，网民对丰富精神生活的渴望不断提升。

本论文便是基于巨大的市场需求来建设在线数字音乐销售网站。研究课题的主要内容是基于J2EE来建设数字音乐在线销售网站。数字音乐是以一种非结构化数据形式存储在网络某一节点中的文件。无论被传播多少次，数字音乐的品质仍不会改变。本网站由时下较为流行的SpringBoot搭建而成的项目。前端也采用框架建设模式，即使用Layui前端模板。Layui作为WEB项目的前端快速开发框架他具诸多优势；例如多种已经定义好的组件可以直接会用，也可以仅通过标签的class属性引入相对应的CSS样式就可以实现布局优化，数据模版引擎可以实现后台数据动态展示等等。通过Layui和Thymeleaf的结合可以更好的实现数据控制和展示。

而用于后台业务逻辑处理的J2EE则采用了SpringBoot框架，该框架因自带Tomcat服务器而部署和开发都节省了大量的人力物力，SpringBoot非常适合小型项目的快速成型建设与中型项目的后台架构建设。SpringBoot带有大量的注解，可以减少代码书写，降低程序员工作强度。同时该框架可以完美兼容SpringMVC ，Spring以及MyBatis这样能更好的快速利用MVC模式搭建小型项目。SpringBoot契合Thymeleaf模板，通过Thymeleaf模板引擎实现前后端分离。

后台数据持久化层采用MySQL数据库来实现相关功能，MySQL是轻量关系型数据库，它通过MyBatis框架的逆向工程可以快速建立项目初始化。MyBatis数据库的大多数存储引擎都不是简单地行级锁,基于为了提升系统并发性能的考虑,他们一般都同时实现了多版本系统并发性能的控制(mvcc)。

**关键词：电商系统，J2EE，SSM，SpringBoot**

目录

[1 绪论 1](#_Toc41253265)

[1.1 国内外数字音乐平台研究现状 1](#_Toc41253266)

[1.1.1 国外数字音乐平台研究现状 1](#_Toc41253267)

[1.1.1 国内数字音乐平台研究现状 2](#_Toc41253268)

[1.2 研究技术架构 4](#_Toc41253269)

[1.2.1 研究理论架构 4](#_Toc41253270)

[1.2.2 研究技术架构 4](#_Toc41253271)

[1.3 本文结构 5](#_Toc41253272)

[2 相关技术概述 6](#_Toc41253273)

[2.1 SpringBoot框架 7](#_Toc41253274)

[2.1.1 SpringBoot框架的发展 7](#_Toc41253275)

[2.1.2 SpringBoot框架的功能特点 7](#_Toc41253276)

[2.1.3 SpringBoot的两个重要策略 8](#_Toc41253277)

[2.2 Thymeleaf模板引擎 8](#_Toc41253278)

[2.2.1 Thymeleaf定义 9](#_Toc41253279)

[2.2.2 Thymeleaf处理模板的种类 10](#_Toc41253280)

[2.3 SSM框架 11](#_Toc41253281)

[2.3.1 Spring 11](#_Toc41253282)

[2.3.2 Spring MVC 12](#_Toc41253283)

[2.3.3 Mybatis 14](#_Toc41253284)

[2.4 Layui框架简介 15](#_Toc41253285)

[2.5 数据库 15](#_Toc41253286)

[2.5.1 MySQL简介 16](#_Toc41253287)

[2.5.2 MySQL运行原理 16](#_Toc41253288)

[2.6 Redis技术 17](#_Toc41253289)

[2.7 本章总结 17](#_Toc41253290)

[3 需求分析设计 18](#_Toc41253291)

[3.1 需求分析 18](#_Toc41253292)

[3.1.1 功能性需求分析 18](#_Toc41253293)

[3.1.2 非功能性分析 22](#_Toc41253294)

[3.2 系统功能设计 22](#_Toc41253295)

[3.2.1 网站架构设计 22](#_Toc41253296)

[3.2.2 总体功能设计 23](#_Toc41253297)

[3.2.3 网站资源管理设计 25](#_Toc41253298)

[3.3 系统数据库设计 29](#_Toc41253299)

[3.3.1 数据库部分实体关系设计 29](#_Toc41253300)

[3.3.2 逻辑设计 30](#_Toc41253301)

[3.3.3 数据库物理设计 31](#_Toc41253302)

[3.4 本章总结 33](#_Toc41253303)

[4 网站功能实现与测试 34](#_Toc41253304)

[4.1 音乐展示模块 34](#_Toc41253305)

[4.1.1 发现音乐具体实现 34](#_Toc41253306)

[4.2 用户信息模块 35](#_Toc41253307)

[4.2.1 历史播放数据 35](#_Toc41253308)

[4.3 榜单推荐模块 36](#_Toc41253309)

[4.4 商城模块 36](#_Toc41253310)

[4.5 关键逻辑实现 36](#_Toc41253311)

[4.5.1 注册登录实现 36](#_Toc41253312)

[4.5.2 页面数据展示 38](#_Toc41253313)

[参考文献 40](#_Toc41253314)

[致谢 1](#_Toc41253315)

1. 绪论
   1. 国内外数字音乐平台研究现状
      1. 国外数字音乐平台研究现状

国外互联网起步早于国内，且科学技术发达，数字音乐平台也是种类较多。较为优秀的国外数字音乐平台有MySpace Music，Last.fm，TuneIn，Spotify等等。

Spotify数字音乐平台在众多的数字音乐平台里我们以Spotify为例介绍他的运营模式，它运营的服务模式主要分为免费服务和收费服务，免费用户在使用时会弹出广告而收费用户则没有广告，收费用户还会得到更好的音乐，在其他设备上登录Spotify还能够享受平台的全部功能。Spotify主要的收入来源是广告商的高额广告费用和收费会员的会员订阅费用。

Spotify对平台通过广泛收集用户非敏感浏览信息经过整合再通过深度学习预测用户的喜好，采用协同推荐算法实现用户的个性推荐功能。在用户更多的使用Spotify，Spotify平台就会收集更多的用户浏览信息，这样会形成一个良性的持久的循环。计算出用户特征后对用户推荐就会更加精准从而增加用户粘性。在于用户保持一定好感度之后，Spotify平台会尝试推荐一些收费的内容给普通用户。这些premium会给付费用户带来极好的体验，优质的体验难以割舍，让用户不断的续费以实现会员扩增。

Spotify对平台的数字音乐的版权有着强有力的保护措施，Spotify平台采用多重加密算法传输网络数字音乐资源，原因其一为了保证网络传输过程避免数据丢失，保持信息的完整性，避免给用户带来不良的音乐体验，提高音乐的品质。其二就是为了防止网络爬虫抓取付费音乐，避免原创音乐人承受版权侵害的损失。

Spotify在线数字音乐平台在成长初期就具备了电子商务的特点：

（1）交易虚拟化，通过WEB应用进行网上付费，享受付费后带来的音乐品质的提升，有权限使用Spotify数字音乐平台的全部功能。

（2）交易成本低，通过网络的数字音乐购买行为，会员充值行为用很低的成本在运行着。

（3）交易效率高，Spotify在线数字音乐平台可以同时与众多用户进行交易且可以实现“钱到货到”。几乎是零延迟完成交易行为。

综合国外研究现状可以总结出在线音乐平台具备以下四个特点：

（1）用户之间的差异化

普通用户与付费用户应有不同的特权，且付费用户可以享受平台的全部功能。

（2）用户喜好个性化

利用深度学习计算用户的偏好，推荐用户喜欢的内容与歌曲类型

（3）维护原创人的根本利益

通过强有力的版权保护措施来防止原创音乐人利益受损，激励原创音乐人创作激情。

（4）数字音乐商品化

通过对数字音乐的商品化获得收益。使网站运营进入到一个良性循环中。

* + 1. 国内数字音乐平台研究现状

目前的在线数字音乐平台国内外均有较为成熟的数字音乐平台例如国内的平台有网易云音乐，搜狗音乐等。国外的数字音乐平台有MySpace Music，Last.fm，TuneIn，Spotify等，这些成熟的数字音乐网站用自己独特的方式吸引着特定的用户群体。下面具体以我国在线数字音乐现状做比较分析。

在谈现状之前要了解前面我国在线数字音乐发展史。与国外的在线数字音乐发展不同，我国在线数字音乐在早期发展就较为困难，整体的大环境对在线数字音乐的版权保护十分不友好，各大数字音乐平台处于亏损较为严重的情境下仍坚持对数字音乐发展的态势持乐观态度。但整个市场版权意识淡薄，在线数字音乐平台提供线上播放试听，免费下载的服务，才绑定了一批用户。但是并没有给这些数字音乐平台带来可观的收入。

随着知识产权在我国日益受到重视，版权监管更加严格，各大数字音乐平台相互之间的斗争开始由版权之争趋向平缓。个性化的推荐成为锁定用户的主要手段，以用户为中心，逐步建设更加和谐温馨的数字音乐社区，隔离甚至排斥其他数字音乐平台的用户，这就是用户需求至上为中心产生的结果，这也给平台带来了收益。国外与国内的在线数字音乐发展很相似，都具备用户差异化，喜好推荐，版权保护以及数字音乐商品化。

概括来说就是通过细分市场，仅对数字音乐这类商品做针对性营销，通过电子商务的模式来运营整个数字音乐销售生态，使这个系统不断的发展壮大。国内借鉴了国外先进的运营思想，首先建立起能吸引用户的特色WEB应用，然后通过推荐系统增加客户好感和用户粘性，最后结合用户差异化策略激励用户付费。近些年国家不断加强对知识产权的保护，整个社会消费态度也在慢慢转向支持正版。

接下来根据国内较为热门的网易云音乐平台的运营状况为例分析国内整体的数字音乐平台发展状况

根据易观（2018）报告显示，酷狗音乐和QQ音乐占有国内数字音乐发布平台绝大部分市场，尽管网易云音乐平台年度活跃用户数量排名在第四位，与市场巨头相差甚远。但网易云音乐用户活跃度却领先一些，成为用户活跃度的榜一[[[1]](#endnote-1)]。用户活跃度越高，音乐平台的用户就越不容易流失。创造如此稳定的用户群体这就要归功于网易云音乐的独特的用户原创内容。

首先，音乐准确推荐是网易的主要优势之一。其算法可实现基于用户行为的智能推荐系统，并每天向用户推出个性歌单。如果用户想要发掘更多个性化的音乐，则可以打开“私人定制FM”，可根据用户喜好不间断推荐歌曲。网易云音乐的UGC（User Generated Content）评论也是其主要优势之一[[[2]](#endnote-2)]。在每首歌曲下，用户都可以添加评论，阅读有趣的评论逐渐成为大量用户的主要乐趣之一[[[3]](#endnote-3)]。

网易云音乐通过UGC评论吸引众多用户的同时也吸引了众多独立音乐人。大量的艺人带来了大量的作品，网易云音乐的曲库种类丰富，数目庞大。而随着知识版权在中国越来越被重视，平台用户更愿意通过消费来获得优质音乐或者独家定制内容。

根据国内的音乐平台的分析可以概括国内音乐平台的特点；

（1）个性推荐

众多音乐平台常见的系统，主要是增加客户留存。

（2）UGC互动

通过UGC互动来增加平台的乐趣，增强音乐烘托的情感氛围，获得用户好感。

（3）维护版权

网易云平台通过给原创艺人添加认证，原创歌曲增加标签的方法来维护原创艺人的利益。

通过分析国内外数字音乐平台的建设也可以看出一些不足的地方，例如推荐系统过于僵硬，仅推荐用户喜欢的风格而统一风格久而久之势必会生厌；普通用户和付费用户差异化较为明显，甚至限制平台的功能，这就会阻碍新用户的体验；曲库数量有限不能很好的满足用户需求。

国内外数字音乐平台已经走上较为成熟的发展时期，大数据的进一步发展将会成为数字音乐平台的又一次质变的强力推进剂。音乐平台的特点是用户喜好个性化，数字音乐商品化，版权维护正规化以及网站建设现代化。所以音乐平台必须遵从市场规律，紧跟市场发展方向。

概括国内外目前针对数字音乐平台的研究情况可以从中得到建设数字音乐销售网站的经验。作为电商平台的一个分支，其具备电子商务开展活动的先天条件:商城，消费者，产品以及“物流”。因为在线数字音乐销售系统本就是一个商城，其中的数字音乐会作为产品，而物流部分会被简化在网络上快速进行。建设电子商务平台时还要让平台具备一定的个性推荐能力，版权维护能力，内容创新能力。

* 1. 研究技术架构
     1. 研究理论架构

在前面分析国内外数字音乐平台发展情况时采用了案例研究法。通过收集大量文献资料，将以Spotify，网易云音乐研究分析数字音乐平台的特征，优势以及不足。通过对个例对比研究分析得出此次数字音乐平台建设应具备哪些基本属性和功能。

* + 1. 研究技术架构

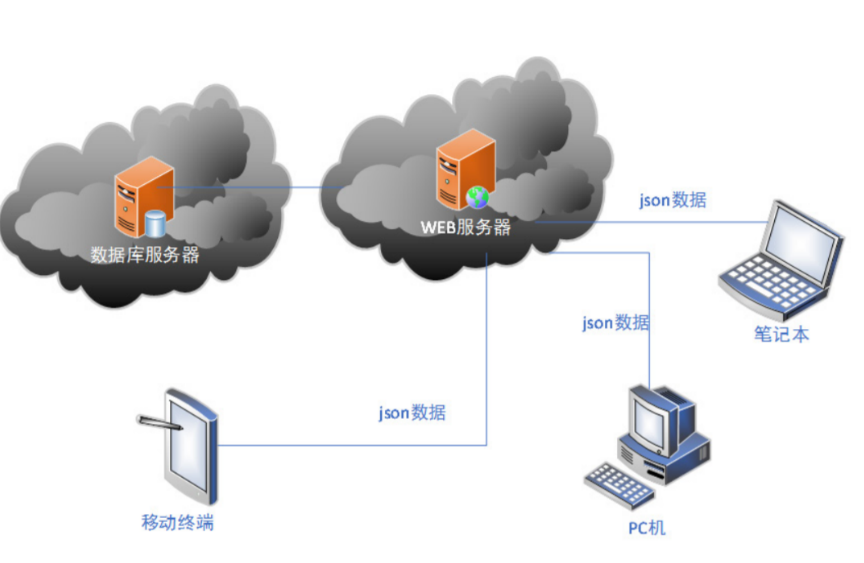
B/S 结构是一种常见的网络应用架构模式。这种架构模式统一了终端访问功能，将系统功能的重要逻辑处理部分集中到应用服务器上，加快了系统的开发进程、降低了维护和使用的费用[[[4]](#endnote-4)]。

图1-1：B/S 结构的拓扑图

在线数字音乐平台的建设是采用前后端分离技术，主要是前端通过 ajax请求去调用后端的接口，使用 JSON 格式的数据在前后端之间通信。这种开发框架下，controller和一部分service也被认为是前端的一部分， 后端主要用来处理前台的业务请求，查询、插入和更新数据库以及调用其他service[[[5]](#endnote-5)]。前后端分离模式会将项目一分为二：前端开发项目和后端开发项目。前端项目是WEB展示的页面开发，后端项目是指数据库访问和部分业务逻辑处理，通过JSON的形式来提供API接口供前端调用。管理API接口的工具众多，本次使用Swagger Tools分类归整APIs。这种模式的部署结构如下图所示。

5G的到来会降低用户对APP的依赖，更多的云应用会应运而生，遍地开花。前后端分离技术借助SpringBoot来实现，SpringBoot作为目前流行的项目构建支持技术，深受开发人员的喜爱，其高拓展性给SpringBoot项目带来更多可能。同时SpringBoot能很好的融合SSM开发框架，这样更容易搭建项目后台基础与架构。前端借助Layui成熟的前端框架体系，减少前端代码的编写加快开发速度。Layui拥有响应式框架，这样能更好的兼容不同的设备，这更合B/S的高可移植性向契合。对于后台数据库，本项目将采用关系型数据库MySQL数据库。同时借助Redis技术加快用户点击功能键的反应速度，提升用户体验。

* 1. 本文结构

这篇文章用四个章节来叙述了基于J2EE的数字音乐销售网站设计。

第一章节主要介绍了当下数字音乐网站的发展历史与现状，并举例分析了几种不同类型的数字音乐网站优缺点。同时介绍了本论文的组织结构。

第二章节重点讲述搭建网站使用到的相关技术，其中有B/S模式的技术原理， SpringBoot的发展与技术原理，最后讨论本次研究要用到的工具并介绍他们的功能特点和基本原理。

第三章节主要对系统做需求分析，结构设计以及后台POJO实体类设计，并绘制模块的用例图、功能的ER图以及整体的结构功能设计图。

第四章会对系统各个功能板块以及实现方法做整体细致的阐述。

1. 相关技术概述

在线数字音乐平台作为互联网时代的产品必须符合互联网的特点；敏捷开发，快速响应，安全稳定。所以本次研究打算采用Spring系列的产品，Spring系列框架作为目前主流的应用开发框架，被广泛的应用于各个领域。其中的SpringBoot将会被应用于此次开发，作为后端的成熟的框架，SpringBoot内嵌Apache服务器，同时利用maven功能对后端需要用的jar包做统一版本管理，减少缺少依赖jar，jar版本冲突等相关问题。Layui前端优化响应式框架能合理的兼容各种设备，同时数据库利用MSQL数据库快速存储。

整体架构如图2-1所示。

Layui

Thymeleaf

HTML

CSS

JavaScript

SpringBoot

Mybatis

MySQL

Redis

图2-1：网站技术架构

在经典的MVC架构下，实现对整个应用服务软件的适配工作。以下将介绍本次研究用到的技术架构并进行详细分析与探讨。图2-1展示本网站的主要架构

* 1. SpringBoot框架
     1. SpringBoot框架的发展

随着框架技术的快速发展，JavaWeb应用从最早的Servlet开始，经历了许多的演变，最终形成了一套具有完整生态系统的开源组件。而SpringBoot作为开源组件中核心成员功能丰富、开发便捷，逐渐成为目前开发人员使用最广泛的开源框架之一[[[6]](#endnote-6)]。

SpringBoot框架中集合了众多Spring的子模块。Spring主要由许多统一的业务模块组成，这些模块包括Spring的上下文容器(IOC)、切面编程(AOP)、系统稳定性框架、持久化框架、以及常见的诸如SpringMVC框架和用于单元测试的Junit等。其中控制反转功能可以使系统程序设计解耦合，提高了系统的可维护性，也使测试工作变的更加容易。Spring框架一直以来就比较成熟和稳定，也是目前大部分公司做开发时候的选择。但是随着项目业务的深入，系统功能的庞杂，Spring带来了大量繁杂的配置工作。大量的XML文件配置工作和注解会导致在项目开发中系统的维护难度增加，开发效率变低。在这种情形下，框架开发人员寻求一种更轻量级和快捷开发工作，可以使开发人员的主要工作注意力放在业务的实现上而不是配置上，从而导致了SpringBoot框架的诞生。2012年10月，Mike Youngstrom提出了在Spring框架中能够支持无容器Web应用程序体系结构。他首先提出了Spring的Web应用体系结构可以通过Spring组件和配置模型的工具大大简化，然后通过main()方法引导的Spring容器内嵌入和统一这些常用的Web容器服务的配置。这一要求也导致了SpringBoot的诞生，2014年，Spring Boot1.0第一个版本诞生了。随后Spring Boot的各种版本都诞生了，功能也愈加齐全。2018年，Spring Boot2.0上线了2.0版本的SpringBoot框架可以不但支持Java9.0兼容Quart，而且还支持嵌入式Netty等。

* + 1. SpringBoot框架的功能特点

SpringBoot是对Spring高度整合的开源框架。该框架的用途是简化SSM框架的搭建， SpringBoot作为一款轻量级开源框架，其主要有Out of Box-“开箱即用”和Convention over Configuration-“约定大于配置”两大重要策略。基于这两大策略，开发者摆脱了繁琐的配置工作以及复杂的依赖管理任务，只需要专注于业务的实现逻辑即可，maven会通过pom文件自动管理相关的jar。不会生成额外代码量，不需要进行XML文件的配置。SpringBoot提供了更快的基于Spring的开发体验。在大型项目开发中也经常用到SpringBoot为我们提供的丰富特性。

同时SpringBoot的内置嵌入式Tomcat服务器能让开发的系统直接打成JAR或者WAR，只要有Java环境，就可以通过命令启动SpringBoot项目，从而实现项目的灵活移植。嵌入式的服务器让前端页面可以直接在项目里面运行，借助Thymeleaf模板引擎以实现前后端分离，降低前后端的耦合。让thymeleaf模板开发一次实现到处运行。

SpringBoot所具备的特征有：

（1）可以创建独立的Spring应用程序，并且基于其Maven或Gradle插件，可以创建可执行的JARs和WARs

（2）内嵌Tomcat或Jetty等Servlet容器

（3）含有项目对象模型，简化maven配置

（4）尽可能自动配置Spring容器

（5）有着外部化配置器

（6）无代码生成，剔除xml类型文件[[[7]](#endnote-7)]。

* + 1. SpringBoot的两个重要策略

1、Out of Box “开箱即用”

是指在开发过程中，通过在maven项目的pom文件中添加相关依赖包，然后使用对应注解来代替繁琐的XML配置文件以管理对象的生命周期[[[8]](#endnote-8)]。这个特点使得开发人员摆脱了复杂的配置工作以及依赖的管理工作，更加专注于业务逻辑。

2、Convention over Configuration “约定大于配置”

是一种由SpringBoot开发后自带的配置目标结构功能，由开发者在结构目标文件中添加相应信息的软件设计规则。这一特点降低了部分活用性，增加了BUG定位的复杂性，但减少了开发人员需要做出决定的数量，同时减少了大量的XML配置，并且可以将代码编译、测试和打包等工作自动化。

* 1. Thymeleaf模板引擎

Spring Boot 开发的项目压缩方式是JAR，且采用的是内置的Tomcat服务器，并不支持JSP格式的前端页面。 SpringBoot 官方推荐使用 Thymeleaf模板引擎，Thymeleaf支持SpEL。通过下面的语句就可以在SpringBoot项目中引入Thymeleaf模板引擎

Thymeleaf可以用于渲染HTML5内容的模板数据引擎。它也可以轻松的与SpringMVC等框架进行集成作为Web应用的模板引擎。Thymeleaf 支持HTML、JAVASCRIPT、CSS 。这就意味着他完全和HTML代码完美相容。与其它市面上的模板数据引擎相比， Thymeleaf最大的特点是能够直接在web容器中中打开并正确显示数据渲染后的页面，而不需要启动整个Web应用。使用如下模板代码即可设置。

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

* + 1. Thymeleaf定义

Thymeleaf 是面向 Web 和独立环境的现代服务器Java 模板引擎，其主要用于为web容器提供一个显示正确，反应良好的模板使用方案，Thymeleaf不仅用于开发具有静态效果的页面，更适合开发可以动态展示数据的前端页面。Thymeleaf在嵌入原HTML代码时不会影响原HTML代码的表现，原HTML代码也不会干扰Thymeleaf在程序运行时的表现。Thymeleaf作为一种模板语言有以下特征 [[[9]](#endnote-9)]。

（一）、数据- Date

数据会存储一定的信息，这里会将后台数据库传送到的结构化数据以JSON格式的字符串传递到前台页面，前台通过模板引擎的渲染将数据放在页面合适的位置。

（二）、模板- Template

模板就是一种数据渲染方案，即一个与类型基本没有关联的JAVA类。该类在使用时会自动创建一个带有参数类型传入的类，并利用参数的传入进行数据初始化。

（三）、模板引擎- ThymeleafEng0ine

生成模板引擎是为了将用户的浏览界面与业务相关的内容分开。 它可以将数据呈现到模板，使用特定格式的数据生成文档，并且用于网站的模板引擎将生成符合标准要求的HTML文档。

（四）、结果文档- Result Documents

数据经过模板引擎的计算被渲染在模板上形成的特殊结构的文档，模板引擎就会生成一个符合格式要求的HTML文档。Template在Thymeleaf模板引擎具体运行原理如图2-2所示

图2-2：Thymeleaf模板引擎运行原理

**Template**

**Date**

**Thymeleaf**

**Engine**

**Result Documents**

Thymeleaf模板引擎可以在后台直接打开运行，方便后台程序员调试。也可以直接在有网络或者前后端分离的开发模式下，让前端工程师调试。大大的加快了开发速度。模板的数据引擎更适配了SpringBoot无需复杂的配置。只需引入JAR，开发者只关注Thymeleaf的数据引擎的语法就好。

* + 1. Thymeleaf处理模板的种类

Thymeleaf模板引擎也可以做到“Out of Box”开箱即用。Thymeleaf可以处理六中模板，而每一种模板我们会叫它模板模式。模式分为标记模板模式HTML和XML，文本模板模式TEXT，JavaScript，和CSS以及无操作模板模式RAW[[[10]](#endnote-10)]。

（一）、HTML模板模式

HTML模板模式是一种标记语言利用这种标记语言可以快速处理任何HTML数据的输入，包括HTML5，HTML4和XHTML，在这种模板模式下，将不会执行验证或格式检查，并且在输出中尽可能地遵守模板代码/结构。

（二）、XML模板模式

XML模板模式允许XML的输入，在这种模板框架下，代码的格式是良好的，不会出现未封闭的标签。属性的格式也是统一的。格式若是出现异常，解析器就会抛出异常。

（三）、TEXT模板模式

将允许对非标记性质的模板使用特殊语法。 这种模板的例子可能是特殊格式的文本或固定格式文档。对于HTML文件或XML文件也是可以作为TEXT文本处理，在这种情况下，它们不会被解析为标记，而每个标记，DOCTYPE或者注释等都将被视为纯文本。

（四）、JAVASCRIPT模板模式

将允许处理Thymeleaf应用程序中的JavaScript文件。这意味着能够像在HTML文件中一样使用JavaScript文件中的模型数据，但是使用特定于JavaScript的集成(例如专门转义或自然脚本)。 JAVASCRIPT模板模式被认为是文本模式，因此使用与TEXT模板模式相同的特殊语法。

（五）、CSS模板模式

将允许处理Thymeleaf应用程序中涉及的CSS文件。类似于JAVASCRIPT模式，CSS模板模式也是文本模式，并使用TEXT模板模式中的特殊处理语法。

（六）、RAW模板模式

根本不会处理模板。它意味着用于将未触及的资源(文件，URL响应等)插入正在处理的模板中。例如，可以将HTML格式的外部非受控资源包含在应用程序模板中，从而安全地知道这些资源可能包含的任何Thymeleaf代码都不会被执行 [[[11]](#endnote-11)]。

* 1. SSM框架

SSM框架由Spring，SpringMVC和Mybatis的缩写组成。它由Spring和Mybatis的两个开源框架集成[[[12]](#endnote-12)]。

* + 1. Spring

（一）、Spring发展历史

Spring是一个成熟的开源框架，为了降低企业级应用的创建难度创建Spring框架。其高效开发的基础主要依赖于Spring框架的快速配置，多种可选注解以及AOP编程[[[13]](#endnote-13)]。

（二）、Spring框架的特点

Spring通过DI（Dependency Injection）减少对象之间的耦合。AOP切面编程可以轻松实现对公共功能的修改，从而更加专注于系统的核心功能。SpringMVC是Spring开源框架的组成部分，MVC模式是Spring对后端的分层处理，以减少耦合；MVC设计模式是分为Model层，接收Controller请求。View层，即是视图层是经过model处理之后返回的数据结果。Controller层，即是接收http请求，调用model层，返回view层。

a. Dependency Injection

Dependency Injection（依赖注入）也称为Inversion of Control（控制反转），即是利用JAVA的高级特性反射原理为基础实现的将对象的生命周期托管给Spring容器。在Spring相关应用使用bean时自动创建，从而减少对象的创建减低对内存的占用率。

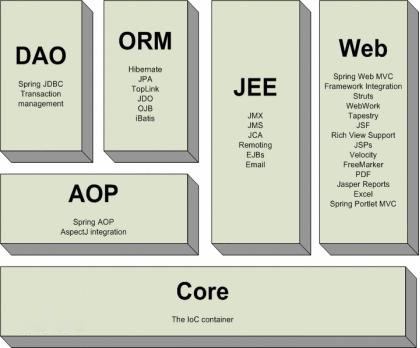
b. Aspect Oriented Programming

AOP（面向切面编程）作为Spring重要的编程思想，其对Object Oriented Programming（面向对象程序设计）做了补充和完善，AOP能够对分散的对象做公共行为的引入。将程序中共同业务逻辑封装成一个切面类的方法中，然后织入到的业务逻辑中。

* + 1. Spring MVC

（一）、Spring MVC简介

SpringMVC是实现了MVC设计模式的成熟开源框架，主要作用于J2EE搭建的WEB系统的web层，简化web层的开发难度。Spring MVC 是基于SpringFrameWork构造而成的后续产品，可以说Spring MVC就是Spring的一个模块，所以无无需整合SpringMVC就和Spring完美兼容了。下图为Spring MVC的架构图。

图2-3：Spring MVC的架构设计图。

（二）、SpringMVC原理解析

Spring MVC有六个主要的组件DispatcherServlet（前端控制器），HandlerMapping（处理器映射器），Handler（处理器），HandlAdapter（处理器适配器），ViewResolver（视图解析器），View（视图）。SpringMVC通过组件之间的相互调用配合完成接收前端请求和返回处理结果。

a. DispatcherServlet（前端控制器）

前端传入的请求数据会被DispatcherServlet拦截，在DispatcherServlet前端控制器接收消息后调用其他解析器进行处理，DispatcherServlet只是作为所有访问请求的节点。

b. HandlerMapping（处理器映射器）

DispatcherServlet(前端控制器)会请求HandlerMappering（处理器映射器）去查找Handle（处理器）处理请求，HandlerMapping（处理器映射器） 根据URL找到具体的controller(控制器)生成处理器对象。

c. Handler（处理器）

Handler处理器又被称为controller控制器。主要处理各个业务逻辑并返回处理结果。

d. HandlerAdapter（处理器适配器）

由DispatcherServlet 根据返回的Handler对象，经过适配HandlerAdapter处理器适配器调用Handler具体的方法。HandlerAdapter将Handler执行结果ModelAndView返回给DispatcherServlet

e. ModelAndView（视图解析器）

DispatcherServlet将ModelAndView传给ViewResolver进行解析。ViewResolver解析后返回具体View给DispatcherServlet，DispatcherServlet对View进行渲染视图后响应给客户。

具体的运行原理入下图所示；

DispatcherServlet

中央控制器

HandlerMapping

处理器映射器

HandlerAdapter处理器适配器

Handler

处理器

ModelAndView

视图解析器

1.http请求

2.查询处理器

3.返回Handler

4.调用适配器

5.调用处理器

6.返回ModelAndView

7.返回ModelAndView

8.调用解析器

9.返回view

图2-4：Sprin MVC运行原理

* + 1. Mybatis

（一）、MyBatis简介

Mybatis作为基于Java的对象关系映射框架，支持定制化SQL功能，高级映射功能以及存储过程配置功能，Mybatis可以通过配置文件实现Java的任意POJO和接口映射到数据库中的数据[[[14]](#endnote-14)]。让开发者减少代码书写不用关心数据库与应用的链接与否。另外MyBatis还有更加强大的功能；支持动态SQL能去完成更加复杂的SQL查询。

（二）、Mybatis原理简介

Mybatis框架的运行时主要分为三个功能层；基础支撑层，数据处理层和API接口层。功能层之间的运行逻辑如下所示。

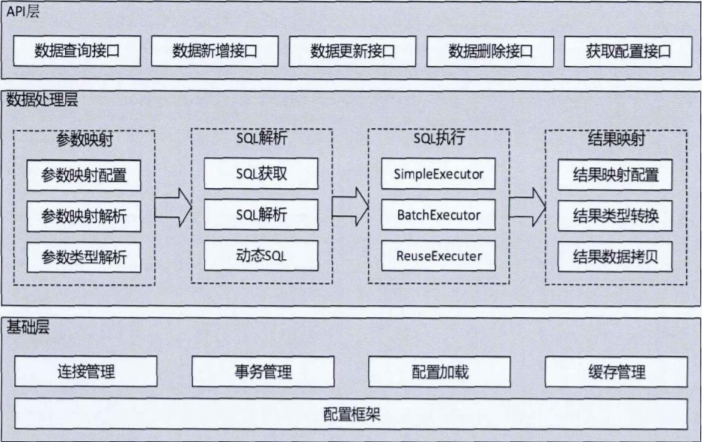


图3-5：Mybatis架构

基础层负责处理配置文件的加载、实现与数据库的连接、事务管理及缓存管理的服务；数据关系层负责生成和执行 SQL 语句，实现 ORM 机制；API层提供封装好的接口 API，方便开发者调用[[[15]](#endnote-15)]。

Mybatis的主要特点在于可以根据XML配置文件简化JDBC代码的书写，忽略SQL语句的书写。而XML配置文件主要有两类：配置文件和映射文件。配置文件主要是用来配置数据库连接信息，并声明各个配合文件。映射文件主要用来声明POJO和数据库中表的对应关系，SQL语句等详细信息。程序通过创建会话，加载主配置文件，创建实例。相关程序将映射配置文件中声明的 SQL 语句作为参数，传达给实例，即可执行的 SQL 语句，最终达到访问的目的。

* 1. Layui框架简介

Layui是一款自主研发模块化前端设计框架，完美兼容原生前端代码的书写与组织形式，门槛极低，拿来即用。体积轻盈，组件丰盈，非常适合界面的快速开发。

Layui主要面向后端开发者，提供了大量丰富的内置模块。在加载时会按需加载，主要模块有layer、layDate、table和form等

* 1. 数据库

在线电子数字音乐销售网站是基于MySQL数据库搭建的， MySQL不会过多消耗内存，运转平稳可靠，是中小型网站开发数据库系统的首选。同时利用Navicat与数据库连接，直接在Navicat中操作MySQL数据库。

在同类型的系统中其 使用最为流行，同时它还是在最适于用在 WEB 应用上的一个应用软件。MySQL 所配备的 SQL 语言是访问数据库时最为常用最为标准的查询语言。其实施双授权政策，涵括商业版与社区版两个版本，具备资源占用少，响应速度快，使用成本不高，源码呈开放性等诸多优势，通常在开发中小型网站时， MySQL 都是搭建数据库的首选。

* + 1. MySQL简介
    2. MySQL运行原理

MySQL服务器逻辑架构分为三层：客户端，核心功能层以及存储引擎层。三成架构之间相互配合，处理客户端的数据请求。

（一）、客户端

客户端是用来提供服务或者使用工具。主要有连接管理，授权管理和安全管理等。

（二）、核心功能层

MySQL的核心功能层，其中包括SQL语句解析，内置函数调用，存储函数调用，视图存储等[[[16]](#endnote-16)]。

（三）、存储引擎层

主要实现的数据的查询和存储，数据库服务器通过应用程序接口与核心功能层进行数据交换，而核心功能层提供的应用程序接口屏蔽了存储引擎的差异性[[[17]](#endnote-17)]。

客户端

**连接/线程处理**

**解析器**

**查询**

**缓存**

**优化器**

存储引擎

图3-6：MySQL运行原理

* 1. Redis技术

Redis作为日志型key-value数据存储系统，支持多种字段类型。Redis有着极高的反应性能，多样的字段类型。作为运行在内存中的数据存储系统，Redis支持数据持久化。

Redis支持多种数据结构，支持事务和不同级别的磁盘持久性存储，并通过Redis Sentinel和Redis Cluster自动分区提供了高可用性[[[18]](#endnote-18)]。

Redis 处理数据按下面两种方式：

1、对于重要的数据会先存到数据库，然后存到 Redis 缓存中，这样对于重

要的数据不会有丢失的风险。

2、对于要求响应速度很高的数据会先写 Redis 缓存中，然后再通过消息

队列再写入到数据库中[[[19]](#endnote-19)]。

* 1. 本章总结

本章主要叙述了在线数字音乐销售系统采用的主要实现技术和各种理论知识，首先第一章讲述了国内外研究现状，分析数字音乐网站的共同点和部分缺点，在接下来的第二章中概述了本系统开发将会使用的关键技术包含Spring相关技术等。

1. 需求分析设计

上一章节介绍了本文的选题背景和用到的相关技术，本章节会详细阐述在线数字音乐销售网站的系统功能需求，整体架构，并且会对用户需求做详细分析与论述。在分析需求之后会依托于用户需求做数字音乐销售网站的功能设计，资源整合以及系统架构优化。

* 1. 需求分析
     1. 功能性需求分析

从功能上说，在线数字音乐销售网站作为销售数字音乐的电子商务网站，其具有数字音乐商品展示页面，数字音乐订单管理界面，销售收益管理界面，后台商品管理界面，数字音乐在线试听界面，购物车界面。在数字音乐展示界面分别展示数字音乐专辑封面，数字音乐基本信息，在线播放试听，用户基本信息维护，上传数字音乐信息维护功能。后台数字音乐管理界面用以管理后台订单系统，数字音乐信息维护，前台页面维护等功能。购物车界面主要是用户加入购物车尚未购买的数字音乐。

从用户角色上分析，有管理员角色，普通用户角色和游客角色，管理员可以通过登录后台管理前台页面，用户注册之后能够登录数字音乐展示界面对数字音乐进行试听和购买，而游客只能在数字音乐网站进行收听无法购买。角色分类如下图3-1所示。

图3-1：角色分类

1）管理员

管理员作为本网站维护角色的重要职位，其所拥有的权限应是最大的，管理员必须保证用户上传的数字音乐是正版且该用户拥有版权以及数字音乐必须符合中华人民共和国相关法律，所以管理员必须审核后方能让数字音乐进行发表，同时及时管理订单，反馈用户订单问题。在线数字音乐不涉及第三方物流，而是直接通过网络资源访问权限进行控制。

图3-2：管理员用例

2）普通用户

网站的普通用户是本站的重要消费人群，他们会有多元化的需求与问题。下面主要从用户基本信息维护，用户原创数字音乐信息维护，用户权益维护三方面分析用户需求。

a.用户基本信息维护

游客要成为本站用户要进行基本信息提交，也就是注册，在线数字音乐网站需要用户提供电话号码，账户，账户密码，昵称，身份证号，邮箱作为账号方便用户在丢失忘记密码后找回密码。用户在注册成功后登录在线数字音乐销售网站。可以维护用户基本信息，包括用户头像和用户昵称，查看最近收听的数字音乐，查看购买的所有数字音乐。

b.用户原创数字音乐信息维护

用户上传原创数字音乐必须经过管理员的认证与审核，通过后方可在发表在本网站上进行销售或者免费让用户试听，上传数字音乐时必须要用户的基本信息完整，若不完善基本信息会让用户完善基本信息后发布原创数字音乐。用户可以查看相关作品，对数字音乐进行下架，重传，以及设置是否收费等操作。

c.用户权益维护

用户的上传成功并且设置为收费后，可以在其他本站用户购买后在这里查看收益，在线数字音乐网站会收取数字音乐单价的10%会用作本站的运营维护费用，随着用户上传的原创数字音乐数量和质量的增加，收费比率会逐步降低。在用户权益维护界面用户可以改变对数字音乐的定价，也可以查看收听已购的数字音乐产品。

图3-3：用户用例

3）游客

在线数字音乐销售网站作为一种电子商务网站，必须保持部分内容完全公开以吸引更多的用户，游客是潜在的用户，有着很高的用户转化率，为了能够更好地挖掘用户，系统会在每次游客登录后在游客PC或者移动端留下cookie，方便记录游客基本喜好与曾经收听数字音乐的历史记录。为用户带来更为人性化的体验。游客必须注册为用户后方能和用户享有同样的功能。

图3-4：游客用例

在以上的用例分析中可以概括为下面的在线数字音乐销售网站系统总体功能架构；

图3-5：网站系统总体功能架构

* + 1. 非功能性分析

非功能性分析会展开讨论系统的具体设计是否合理，用户体验是否满意以及业务实现难度。本系统的非功能性需求主要有以下几点。

（一）、设计合理性

在线数字音乐销售网站的界面设计是否有利于用户操作，本系统的主要用户是善于使用互联网资源的群体，所以本系统的前端网页设计新颖时尚，操作简单。本系统整体操作流程顺畅，操作按钮明亮鲜艳，可以快速的找到系统的相关功能，快速上手。

（二）、安全性

阿里云服务器的整体运行状态稳定，系统存在多个安全数据备份，特别对SQL注入，密码保护，身份识别做了相关设计与建设可以避免大多数数据安全问题。

* 1. 系统功能设计
     1. 网站架构设计

本站是基于WEB实现的全部功能，其中PC端，移动端均通过WEB浏览器进入本站，与本站进行资源访问，本站的全部资源均放在云服务器上进行配置，平台结构如下图所示。

PC端

移动端

其他入口

在线数字音乐平台

在线数字音乐平台

在线数字音乐平台



Redis

MySQL数据库

文件资源

图3-6网站架构设计

本次设计采用B\S模式。浏览器会作为提供服务的主要入口，通过URL进入网站的前端后可以体验，操作以及享受这个网站的全部功能。浏览器入口主要针对PC端和移动端设计，考虑到浏览器的适配性暂时仅支持市场上流行的主流浏览器有谷歌浏览器和火狐浏览器。Server服务器将使用MySQL行存储数据库，优势在于行存储式数据库有利于做联机事务处理过程，可以短时间内做出快速反应将处理结果返回给用户[[[20]](#endnote-20)]。而在中间业务处理层将会使用SpringBoot框架处理逻辑，SpringBoot的大量注释，快速开发出后台业务处理服务系统。

考虑到本系统要实现快速开发与上线应用，故采用租赁服购买服务器的方式，租赁阿里云服务器，系统的测试环境，生产环境都将在云服务器上实现，而开发环境在本地电脑端。

* + 1. 总体功能设计

通过上文的分析可以得出本系统可以的分为音乐展示，用户信息，榜单推荐，商城四大类

在线数字音乐销售平台

音乐展示模块

用户信息模块

商城模块

榜单推荐模块

图3-7：功能模块分类

针对以上四大模块又有以下的各个细分模块，如下分析；

（一）、音乐展示模块

音乐展示模块的功能有发现音乐，音乐人，推荐音乐个子功能。其中发现音乐是根据近一周播放量较多的音乐筛选出来的。音乐人是随机筛序出来展示，推荐音乐是根据下载量和播放量累积最高才会推荐到用户界面。

音乐展示模块

发现音乐

音乐人

推荐音乐

图3-8：音乐展示模块

（二）、用户信息管理

用户信息包括用户注册信息，用户登录管理，用户基本信息维护，平台安全退出，播放记录，购买记录，数字音乐发布维护。如下图所示；

用户信息模块

用户注册

用户登录

信息维护

退出平台

播放记录

购买记录

发布维护

图3-9：用户信息模块

（三）、榜单推荐模块

榜单推荐包含三部分：热歌排行，热销排行和音乐人排行。通过给定的算法计算各个音乐或者音乐人的权重大小然后显示出来。

图3-10：榜单推荐模块

榜单推荐模块

热歌排行

热销排行

音乐人排行

（四）、商城模块

商城模块是整个在线数字音乐销售系统的核心，它包括订单管理，购物车管理，后台收益管理。

商城模块

订单管理

购物车管理

后台收益管理

图3-11：商城模块

* + 1. 网站资源管理设计

资源数据是网站得以运行的根本，网站的所有信息全部来自网站的资源管理系统，其中有用户的信息，用户上传的数字音乐信息两个模块。

对资源管理进一步细分又可将用户信息模块分为；用户基本身份信息，用户收听记录，用户购买历史，用户收益总计，用户购物车总览，用户上传数字音乐管理七个部分。用户信息模块对用户的重要信息进行保存与分类，用于计算出用户的兴趣爱好，以推荐用户偏爱类型的数字音乐。分析用户上传的数字音乐计算出用户应抵扣网站维护的费用，网站维护费用计算方式为一首数字音乐分别计算；一首数字音乐应收费用 = 数字音乐定价单价 \* 数字音乐售出数量 \* 10%。

数字音乐信息模块又包括数字音乐基本信息，数字音乐类型信息，数字音乐文件资源，数字音乐审核信息五个模块。

资源系统架构如下图所示，下面会对子模块的功能需求进行详细的分析和设计

图3-12：资源系统架构

（一）用户信息模块

用户信息模块的资源管理分为用户基本身份信息，用户收听记录，用户购买历史，用户收益总计，用户购物车总览，用户上传数字音乐管理六个部分。

1）用户基本身份信息

用户在注册成功时，后台资源管理系统会记录下用户的基本身份信息；包括用户头像，昵称，性别，邮箱号，密码，电话号码，身份证号等信息，用户必须在提交注册信息时接收本站使用协议，同时对邮箱的正确性进行验证，点击注册按钮，邮箱验证完成即会注册成功。

在填写用户注册信息时用户的昵称将作为以后用户登录本站的账号，必须保证用户昵称的符合规范性，在登记信息时进行验证，当输入框输入完成时会返回用户此时注册的昵称是否符合规范。昵称作为重要的登录凭证之一，必须符合以下具体规范；

唯一性：填写的用户昵称必须是在本网站未被注册过的昵称

正确性：昵称必须在2到18位字符之间的昵称

合法性：昵称不能包含违反国家法律，反社会的字词。

注册信息的填写除昵称的要求外，还有下表列出的具体要求：

|  |  |  |  |
| --- | --- | --- | --- |
| 填写项目 | 是否必填 | 填写规范 | 错误警告 |
| 头像 | 否 | —— | —— |
| 昵称 | 是 | 符合唯一性，正确性，合法性要求 | 昵称已被使用、昵称不合规。 |
| 密码 | 是 | 必须为8位以上的密码 | —— |
| 密码验证 | 是 | 和密码相同 | 密码验证不一致 |
| 电话号码 | 否 | 必须是11位数字 | 电话号码不合规 |
| 邮箱 | 是 | 必须是规范可用的邮箱 | 请填写正确的邮箱地址 |
| 身份证号码 | 是 | 必须是正确的号码 | 号码格式有误 |

表3-1：注册信息要求

2）用户收听记录

资源管理系统会自动的记录用户收听的数字音乐历史记录，包括听歌累积时长，听歌历史歌名，数字音乐基本信息，数字音乐播放列表。用户可以通过这些收听信息再次收听自己曾经听过的数字音乐，也可以将未听过的数字音乐加入播放列表等待下一曲播放。在收听信息列表里可以试听数字音乐，同时也可以直接在列表中进入数字音乐的购买界面。

3）用户购买记录

在这里用户可以看到自己的购物车信息，历史购买记录。

4）用户收益信息

作为本网站的原创歌手，上传通过审核的原创数字音乐用户就能够在展示页面展示自己上传的原创数字音乐，在他人购买之后会有部分收益到用户的账户上。另部分收益会被本网站收取作为网站的基本运营维护费用。如果用户原创数字音乐的收听量和销售量是本站的月榜第一，会触发本站的奖励制度。这些奖励会在收益明细中展示。

5）用户购物车信息

购物车会作为电子商务网站重要的组成部分，为用户提供了很多的方便。购物车会记录用户加入购物车而未付款的商品，待用户付款后会自动清空购物车，或者在用户使用商品删除功能时也会清除商品。

6）用户上传信息

原创数字音乐用户需要在这里上传原创数字音乐，上传时需要用户填写数字音乐基本信息；包括数字音乐名称，原创作者信息，数字音乐封面，数字音乐时长，数字音乐文件资源，上传之后会在审核状态，后台管理员审核通过后会标记数字音乐审核通过。未通过的会标记未通过审核。

（二）数字音乐信息

数字音乐信息模块又包括数字音乐基本信息，数字音乐类型信息，数字音乐文件资源，数字音乐审核信息四个模块。数字音乐的基本信息是由用户进行填写制作的。是根据管理员审核贴的标签。而对于数字音乐文件资源是存储于服务器资源管理器，本站主要通过管理文件资源路径管理文件。

1）数字音乐基本信息

数字音乐的基本信息包含数字音乐的名称，基本封面，原创作者，上传用户，文件资源路径。这些信息有上传用户直接填写并直接录入数据库。

2）数字音乐类型

是数字音乐在管理员审核时管理员贴上的标签，通过这些类型标签更好地为数字音乐分类识别。以及方便通过标签推荐给有相同爱好的用户，提升用户浏览体验。

3）数字音乐文件资源

作为数字音乐的直接载体，以二进制文件的格式保存在阿里云服务器上，通过文件资源路径访问进而访问数字音乐文件资源，对于设置需要购买后方能访问的数字音乐会设置一定的访问权限以保护原创用户的基本权益。

4）数字音乐审核信息

在用户提交上传数字音乐后，后台管理员会对数字音乐的合规性进行检查。检查的内容有

a.音乐是否为原创，即对比云端服务器是否有相同的音乐文件，

b.音乐内容是否含有违法内容。

c.是否为有效的数字音乐格式。

审核步骤如下图所示：

用户上传数字音乐

格式验证

原创验证

合法验证

修改音乐审核状态

录入数据库

结果反馈

图3-13：审核流程图

* 1. 系统数据库设计

数据库是数据的重要存储位置，数据格式在数据基础结构中占有十分重要的位置，只有根基牢靠上层的建筑才能越搭越高。巧妙的数据库设计不仅能提升系统性能，还能提升效率、降低成本、增强可扩展性

* + 1. 数据库部分实体关系设计

在线数字音乐销售平台部分实体关系图如下所示

数字音乐

用户

订单

发布

拥有

1

n

1

n

图3-14：在线数字音乐销售平台部分实体关系图

* + 1. 逻辑设计

将ER图装换为适用在线数字音乐销售网站的逻辑结构，使之能够在系统中稳定运行。本网站的逻辑结构设计如下所示。

1. 数字音乐表（编号，名称，上传用户，时长，专辑编号，上传时间，地址，价格，审核标志）
2. 用户表（编号，名称，昵称，密码，邮箱，电话，身份证号，级别）
3. 订单表（编号，用户编号，数字音乐编号，创建时间，完成标志）
4. 用户播放列表（编号，用户编号，数字音乐编号，添加时间，播放次数，购买标志）
5. 播放购买记录表（编号，数字音乐编号，标志，时间）
6. 用户收益表（编号，用户编号，数字音乐编号，销售次数，金额）
7. 购物车表（编号，用户编号，数字音乐编号，创建时间，完成标志）
8. 权限表（编号，权限级别，权限路径）
   * 1. 数据库物理设计

一下会详细介绍MySQL数据库物理设计实现的相关实体表结构。

数字音乐表（d\_music）用于存储数字音乐的基本信息。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 是否为空 | 备注 |
| ID | int(16) |  | NO | 数字音乐编号 |
| NAME | varchar(50) | 50 | NO | 名称 |
| USER\_ID | varchar(50) | 50 | NO | 用户编号 |
| TIME | int(16) |  | YES | 音乐时长，单位：秒 |
| COLLE\_ID | int(16) |  | YES | 专辑编号 |
| UP\_TIME | datetime |  | NO | 上传时间 |
| PATH | varchar(50) | 50 | NO | 存放地址 |
| FEE | varchar(1) | 1 | YES | 收费标志：0-免费，1-收费 |
| PRICE | decimal(10,2) |  | YES | 单价，单位：元 |
| PASS | varchar(1) | 1 | YES | 审核标志：0-待审核，1-审核中，2-审核通过，  3-拒绝发布，4-修改后发布 |

表3-2：数字音乐表

用户表（user）存储用户的基本信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 是否为空 | 备注 |
| ID | int(16) |  | NO | 编号 |
| NAME | varchar(50) | 50 | YES | 姓名 |
| NICK\_NAME | varchar(100) | 100 | NO | 昵称 |
| PASSWORD | varchar(50) | 50 | NO | 密码 |
| EMAIL | varchar(50) | 50 | YES | 邮箱 |
| TEL | varchar(20) | 20 | YES | 电话 |
| CARD | varchar(20) | 20 | YES | 身份证号 |
| LEVEL | varchar(1) | 1 | YES | 1-管理员，2-普通用户 |

表3-3：用户表

订单表（order）存储订单相关信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 是否为空 | 备注 |
| ID | int(50) |  | NO | 编号 |
| USER\_ID | int(50) |  | NO | 用户编号 |
| MUSIC\_ID | int(50) |  | NO | 音乐编号 |
| CREATE\_TIME | datetime |  | YES | 订单创建时间 |
| FLAG | varchar(1) | 1 | YES | 0-未付款，1-完成交易，2-销毁订单 |

表3-4：订单表

用户播放列表（play）存储用户播放历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 是否为空 | 备注 |
| ID | int(50) |  | NO | 编号 |
| USER\_ID | int(50) |  | YES | 用户编号 |
| MUSIC\_ID | int(50) |  | YES | 音乐编号 |
| ADD\_TIME | datetime |  | YES | 添加时间 |
| TIMES | int(10) |  | YES | 播放次数 |
| PURCHASE | varchar(1) | 1 | YES | 0-未购买，1-已购买，2-免费 |

表3-5：播放记录表

播放购买记录表（paly\_buy\_his）存储音乐购买播放记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 是否为空 | 备注 |
| ID | int(50) |  | NO | 编号 |
| MUSIC\_ID | int(50) |  | YES | 音乐编号 |
| FLAG | varchar(1) | 1 | YES | 0-购买，1-播放 |
| TIME | datetime |  | YES | 购买或者播放时间 |

表3-6：播放购买记录表

用户收益表（profit）存储用户收益相关数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 是否为空 | 备注 |
| ID | int(50) |  | NO | 编号 |
| USER\_ID | int(50) |  | YES | 用户编号 |
| MUSIC\_ID | int(50) |  | YES | 音乐编号 |
| TIMES | int(10) |  | YES | 售出次数 |
| TITTLE | decimal(50,2) |  | YES | 歌曲售出总价 |

表3-7：用户收益表

购物车（cart）存储用户添加到购物车信息相关数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 是否为空 | 备注 |
| ID | int(50) |  | NO | 编号 |
| USER\_ID | int(50) |  | YES | 用户编号 |
| MUSIC\_ID | int(50) |  | YES | 音乐编号 |
| CREATE\_TIME | datetime |  | YES | 创建时间 |
| PRICE | decimal(20,2) |  | YES | 价格 |
| FLAG | varchar(1) | 1 | YES | 0-未完成，1-已结算，3-异常 |

表3-8：购物车表

权限表（authority）用于限制用户权限

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 是否为空 | 备注 |
| ID | int(50) |  | NO | 编号 |
| LEVEL | varchar(1) | 1 | YES | 级别:1-管理员，2-普通用户 |
| PATH | varchar(100) | 100 | YES | 文件资源路径 |

表3-9：权限表

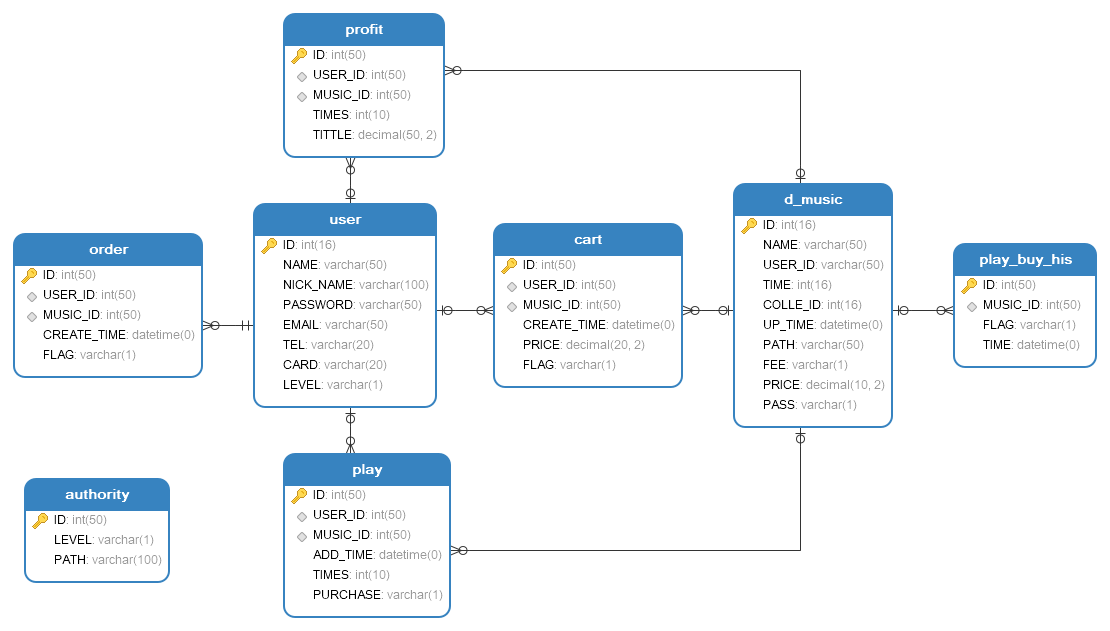
经过以上的数据物理结构设计现在实体关系通过Navict生成，在线数字音乐销售平台数据库设计如下

图3-15：在线数字音乐销售平台数据实体关系

* 1. 本章总结

本章主要论述了在线数字音乐平台的需求分析，从功能性需求和非功能性需求分析，然后阐述了本站的系统架构，为了使数据和平台能快速被应用，详细论述了实体关系，数据库中数据结构的设计以及物理设计。

1. 网站功能实现与测试

上一个章节中，本文对网站的不同用户做了用例分析，根据不同用户的不同需求设计了网站适配的数据物理结构，并对网站用户功能性和非功能性需求进行分析划分出网站建设有四大模块：音乐展示模块，用户信息模块，榜单推荐模块和商城模块。本章会对各个模块做具体实现进行阐述并做部署，测试和结果展示。

* 1. 音乐展示模块

音乐展示模块包含发现音乐，音乐人和推荐歌单三个部分。因三部分的具体实现类似，所以仅介绍发现音乐的实现思路，其他大致简述略过。

* + 1. 发现音乐具体实现

音乐展示模块在网站中作为音乐馆展示，位于网站的侧边常驻导航栏，展示如图4-1，收起后展示如图4-2。



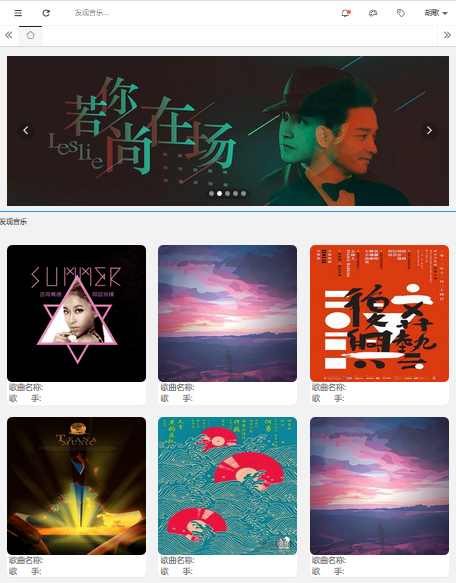
图4-1：音乐馆展示

图4-2：音乐馆收起

在线数字音乐销售网站会预购买一些数字音乐存放于本系统的数据库作为原始积累歌曲，在音乐人逐渐入住后会积累更多的平台音乐人自主上传的音乐。然后存入到数据库的d\_music表里。

新用户成功登录后系统会发送AJAX请求到后台Handler，而后台Handler会调用start()方法查询d\_music表里面的数据，然后随机选取某些音乐直接推荐在用户的“音乐馆”首页。移动端展示效果如下图4-3。

图4-3：发现音乐效果展示



* 1. 用户信息模块

用户信息模块有新用户注册登录和个人信息展示，历史播放数据，购买下载记录等前端展示。本节着重介绍历史播放数据部分实现功能。

* + 1. 历史播放数据

用户播放数字音乐的历史记录会通过AJAX请求记录到后台数据库play表中。每播放一次就会记录一次，以方便统计音乐的热度和个人喜好。历史播放数据展示如下图4-4

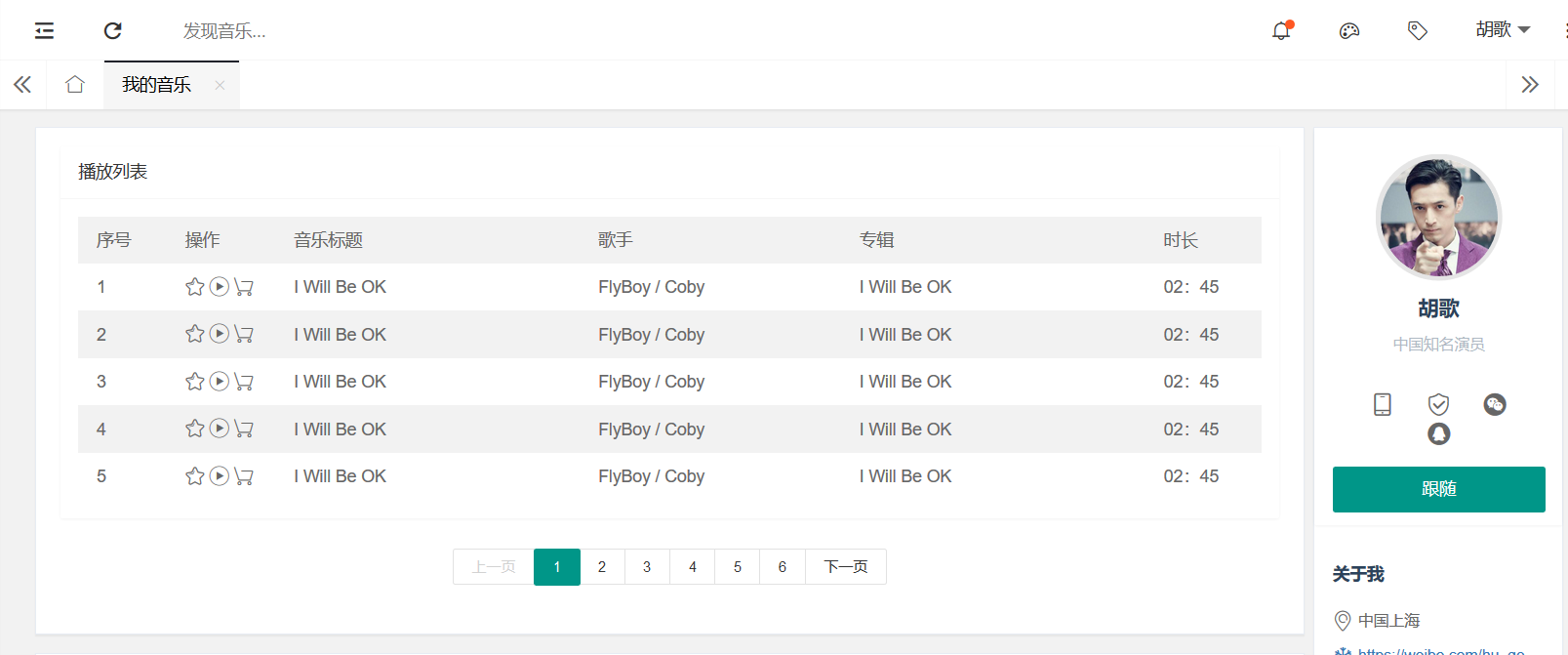


图4-4：历史播放数据展示

* 1. 榜单推荐模块

榜单推荐会根据play\_buy\_his表查询歌曲在某段时间内播放总量和购买总次数，播放总量最大的会放在热歌排行榜，购买总次数最大的会推荐到热销排行榜，而相关音乐人会根据公式（播放总量\*70%+购买总次数\*20%+音乐总售卖金额\*10%）推送到音乐人排行榜。

* 1. 商城模块

商城模块主要用户购买数字音乐的地方，用户可以在音乐馆试听数字音乐，然后跳转到商城模块进行购买。商城模块主要关联了order表，cart表。以下流程图4-5介绍购买流程。

浏览器

Controller

Database

加入购物车

addCart(music\_id,user\_id)

移出购物车

remCart(music\_id,user\_id)

结算购物车

cleCart(music\_id,user\_id)

cart表新增

cart表FLAG状态为3

cart表FLAG状态为1

order表新增

JSON

SQL

SQL

SQL

JSON

JSON

图4-5：购物车处理流程

* 1. 关键逻辑实现

本系统采用前后端分离技术，访问前端主要依赖AJAX控制，业务逻辑在后端controller控制器实现页面访问入口控制，Service实现业务逻辑处理，mapper接口主要映射后台数据库与POJO类。

* + 1. 注册登录实现

登录功能是通过前端发送AJAX请求完成的，具体步骤为

1. 前端发送AJAX请求

|  |
| --- |
| <script>  $("#register").bind("click",function(){  $.ajax({  type:"post",  url:"http://localhost:8888/signIn",//注册地址  async:true,  data:{ username:$("#register-username").val(),  password:$("#register-password").val()  },//准备提交的数据  dataType:"JSON",  success:function(result){//正常回调  alert("success"+result.msg)  if(result.code===0){  alert("准备跳转")  window.location.href=url;  }else {  alert(result.msg);  }  },  error:function(result){  alert("error"+result.msg)  }  });  })  </script> |

1. 后端SignController接收请求以及传输来的数据

|  |
| --- |
| @PostMapping(value = **"signIn"**) @ResponseBody **public** Result signin (@RequestParam(value = **"username"**) String username, @RequestParam(value = **"password"**)String password) {  Result result = **signService**.signIn(username, password);  **return** result; } |

1. 调用具体的SignService处理请求，然后返回JSON数据

|  |
| --- |
| @Service **public class** SignService {  @Autowired  **private** UserMapper **userMapper**;  *//登录* **public** Result signIn(String name, String password){  Result result = **new** Result();  UserExample example = **new** UserExample();  UserExample.Criteria criteria = example.createCriteria();  criteria.andNameEqualTo(name);  criteria.andPasswordEqualTo(password);  User user = **new** User();  **try** {  List<User> users = **userMapper**.selectByExample(example);  **if** (users != **null**) {  user = users.get(0);  result.set(0, **"查询成功"**, 0, user);  }**else**{ result.set(500, **"查询用户不存在"**, 0, **""**);}  }**catch**(Exception e){  result.set(500, **"查询异常"**, 0, **""**);  e.printStackTrace();  }  **return** result;  } |

* + 1. 页面数据展示

页面数据展示也是用AJAX实现前后端数据交互，通过用户唯一ID标识传输到后台查询前端JavaScript代码与登录代码类似，主要展示使用layui框架的table展示。

|  |
| --- |
| <table id="playList" lay-filter="playList"></table>  <script>  layui.use('table', function() {  var table = layui.table;  table.render({  id: "playList",  elem: '#playList',  cellMinWidth: 80 ,  height: 312,  url: 'http://localhost:8888/playList' ,//数据接口  page: true ,//开启分页  cols: [[ //表头  {field: 'id',title: '序号',sort: true,fixed: 'left'},  {field: '',title: '操作'},  {field: 'title',title: '音乐标题',sort: true},  {field: 'userName',title: '歌手'},  {field: 'collection',title: '专辑'},  {field: 'time',title: '时长',sort: true}  ]]  });  });  </script> |

数据表格可以动态的渲染数据，通过layui模板调用数据引擎接收后端数据接口传来的JSON格式的数据渲染上模板。

参考文献

致谢

在进行本次大学生毕业设计时，本人在诸多问题上得到了杨老师的深切的关怀和不断的指正，感谢杨老师提供的各种毕业设计的资源和资料，不仅让我可以顺利完成基于J2EE的数字音乐销售网站设计，更让我进一步拓展了知识面。感谢为我提供帮助的实习同事，给我提供技术上的帮助，助我解决各种技术难题。也在此感谢所有帮助过我的领导，老师，同学和同事，感谢培养我四年的学校。

1. []易观. 中国移动音乐市场年度综合分析2018[J/OL]. [↑](#endnote-ref-1)
2. []应妮. 《中国首份权威数字文化产业报告出炉》[N]. 中国新闻, 2019.08.05. [↑](#endnote-ref-2)
3. []张千绘.网易云音乐:丛林法则下的流媒体音乐平台突围之路[J].卷宗,2019,9(34):370. [↑](#endnote-ref-3)
4. []唐政,熊华报,张停.基于B/S结构的文件管理系统的开发[J].冶金动力,2019,(8):78-80. [↑](#endnote-ref-4)
5. []孟祥双.前后端分离式WEB应用开发研究[J].电子元器件与信息技术,2019,3(6):40-43. DOI:10.19772/j.cnki.2096-4455.2019.6.011. [↑](#endnote-ref-5)
6. []刘敏.基于SpringBoot框架社交网络平台的设计与实现[D].湖南:湖南大学,2018. [↑](#endnote-ref-6)
7. []Spring官网 [↑](#endnote-ref-7)
8. []吕宇琛.SpringBoot框架在web应用开发中的探讨[J].科技创新导报,2018,(8):168,173. DOI:10.16660/j.cnki.1674-098X.2018.08.168. [↑](#endnote-ref-8)
9. []https://www.jianshu.com/p/837087ee2ca6 [↑](#endnote-ref-9)
10. []https://www.bbsmax.com/A/kjdwAbYOJN/ [↑](#endnote-ref-10)
11. []易百教程.Thymeleaf简介[EB/OL]. https://www.yiibai.com/thymeleaf/thymeleaf-instroduce.html [↑](#endnote-ref-11)
12. []郝佳.SPRING源码深度解析 第2版[M]. 北京:人民邮电出版社,2019.01. [↑](#endnote-ref-12)
13. []张泉森.基于Spring框架的高校教学评估系统的设计与实现[D].湖南:湖南大学,2018. [↑](#endnote-ref-13)
14. []陈光剑.Spring Boot开发实战[M].北京:机械工业出版社,2018.08. [↑](#endnote-ref-14)
15. []杨帆.基于Spring和MyBatis框架的物业管理系统的设计与实现[D].湖北:华中科技大学,2016. DOI:10.7666/d.D01309444. [↑](#endnote-ref-15)
16. []BARON SCBWARTZ,PETER ZAITSEV,VADIM TKACBENKO,宁海元,周振兴,彭立勋.高性能MySQL 第3版[M].北京:电子工业出版社,2013.05. [↑](#endnote-ref-16)
17. []BARON SCBWARTZ,PETER ZAITSEV,VADIM TKACBENKO,宁海元,周振兴,彭立勋.高性能MySQL 第3版[M].北京:电子工业出版社,2013.05. [↑](#endnote-ref-17)
18. []王成光. 分布式实时计算框架原理及实践案例[M].北京:电子工业出版社，2016.08. [↑](#endnote-ref-18)
19. []欧阳文臣.基于Redis技术的分布式缓存电商平台设计与实现[D].江西:南昌大学,2018. [↑](#endnote-ref-19)
20. []康旗.大数据资产化[M].北京;人民邮电出版社，2016.12 [↑](#endnote-ref-20)