



广东工业大学

专项设计（论文）

基于 Vue 的音乐播放器

学 院 计算机学院

专 业 软件工程

年级班别 2015 级（3）班

学 号 3115005331

学生姓名 徐静涛

指导教师 郝彦军

2018 年 10 月

摘 要

随着生活水平的提高，人们越来越重视娱乐，每天下班、茶余饭后都喜欢听听音乐，放松自己，陶冶情操。各种音乐播放器 APP 层出不求。本系统旨在运用移动 Web 技术，使音乐播放器更加简洁实用。本系统基于 MVVM 框架，采用 Vue 全家桶开发，由 Vue-router 控制页面跳转，Vuex 管理状态，Axios 向服务器发送请求。该系统实现了登陆注册、搜索音乐、添加音乐、播放控制等基本功能。

关键词： MVVM, Vue, Vue-router, Vuex, Axios

Abstract

With the improvement of living standards, people pay more and more attention to entertainment. They like to listen to music every day after work and after tea and meal, relax themselves and cultivate their sentiment. All kinds of music player app emerge in endlessly. The purpose of this system is to use mobile Web technology to make the music player more concise and practical. This system is based on MVVM framework and is developed by Vue family barrel. Vue - router controls page jumping, Vuex manages status, and Axios sends requests to the server. The system has realized the basic functions of login registration, music search, music addition, and playback control.

Key words: MVVM, Vue, Vue-router, Vuex, Axios

目录

专项设计（论文）	1
1. 需求分析	7
1.1. 引言	7
1.1.1. 编写目的	7
1.1.2. 项目背景	7
1.1.3. 开发应用技术介绍	7
1.2. 任务概述	8
1.2.1. 目标	8
1.2.2. 用户的特点	9
1.2.3. 假定和约束	9
1.3. 需求规定	9
1.3.1. 对功能的规定（功能性需求）	9
1.3.2. 对性能的规定（非功能性需求）	16
1.4. 运行环境规定	17
2. 概要设计	18
2.1. 引言	18
2.1.1. 编写目的	18
2.1.2. 背景	18
2.1.3. 术语定义	18
2.2. 总体设计	18
2.2.1. 需求规定	18
2.2.2. 运行环境	18
2.2.3. 系统流程图	19
2.2.4. 系统功能模块图	19
2.2.5. 功能模块说明	19
2.3. 系统数据结构设计	20
2.3.1. 逻辑结构设计要点	20
2.3.2. 物理结构设计要点	22
2.3.3. 数据结构与程序的关系	22
2.4. 系统出错处理设计	23
2.4.1. 出错信息	23
2.4.2. 补救措施	23
2.4.3. 系统维护设计	23
3. 详细设计	24
3.1. 引言	24
3.1.1. 编写目的	24
3.1.2. 术语定义	24
3.2. 界面	24
3.2.1. 系统登录	24

3.2.2. 注册.....	25
3.2.3. 个人中心.....	26
3.2.4. 推荐.....	27
3.2.5. MV 推荐.....	28
3.3. 系统划分.....	28
3.4. 功能模块详细设计.....	29
3.4.1. 注册.....	29
3.4.2. 登陆.....	29
3.4.3. 添加歌单.....	30
3.4.4. 搜索音乐.....	31
3.4.5. 添加歌曲.....	32
3.4.6. 添加歌曲到播放列表.....	33
3.4.7. 歌曲推荐.....	34
3.4.8. 进度控制.....	35
3.5. 全局数据结构.....	36
3.5.1. Vuex 状态管理.....	36
3.5.2. 全局路由卫士.....	36
4. 测试.....	37
4.1. 测试概述.....	37
4.1.1. 编写目的.....	37
4.1.2. 测试范围.....	37
4.2. 测试计划执行情况.....	37
4.2.1. 测试类型.....	37
测试人员.....	37
4.2.2. 测试问题总结.....	38
4.3. 测试总结.....	38
测试用例执行结果.....	38
测试问题解决.....	38
参考文献.....	39

1. 需求分析

1.1. 引言

1.1.1. 编写目的

本文档是对本项目第一阶段的内容进行分析,对需求细节和实现方式进行了较为详细的阐述。该文档主要为了确定项目的系统功能,文档需求,方便以后开发过程中按照需求完成项目的开发,以达到预期的效果。

1.1.2. 项目背景

随着计算机的广泛应用,手机市场迅速发展,各种音频资源在网上也是广为传播。于是,各种手机音乐播放器 APP 也如雨后春笋般兴起。但是,很多的音乐播放器却一味追求外观和庞大的功能,对用户的手机资源,例如 CPU、内存等造成了很多浪费,用户反映,有很多功能,他们并不需要。而且,有些用户的手机存储空间比较紧俏,庞大的音乐播放器 APP 会占用他们大量的存储空间。

手机上网已经成为重要的上网方式,移动互联网的时代已经开启。

近几年,Android 和 iOS 手机兴起,技术进步,加速了 HTML5 在移动设备的普及,移动浏览器升级换代,给 HTML5 在移动 web 放下的发展带来了无限的可能。移动设备性能的提高,也给移动 web 在在功能逼近客户端应用 APP 上提供了源源不断的动力。

1.1.3. 开发应用技术介绍

1、MVVM

MVVM 分为三个部分,分别是 M (模型层), V (视图层), VM (控制层), M 层主要负责业务相关,V 层主要负责视图相关,VM 层是 V 和 M 层之间的桥梁,负责监听 V 和 M 层的修改^[1]。

MVVM 支持双向绑定,当 M 层数据进行修改时,VM 层会监测到变化,并且通知 V 层进行相应的修改,反之修改 V 层则会通知 M 层数据进行修改,以此

也实现了视图与模型层的相互解耦。

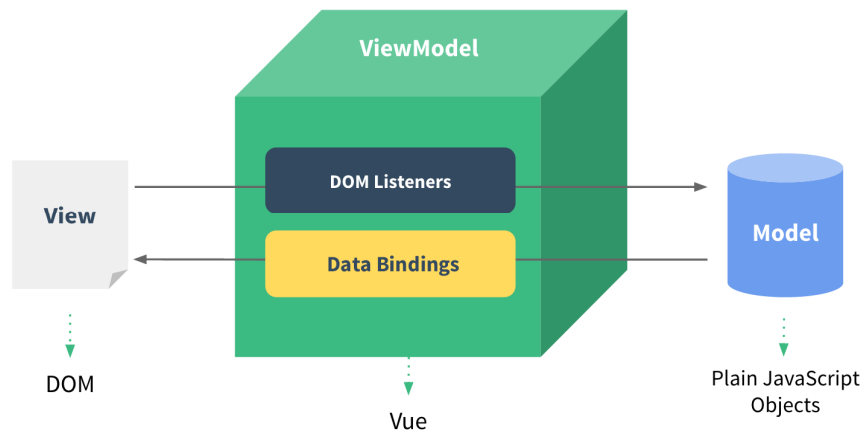


图 1.1 MVVM 模式图

2、Vue

Vue 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

Vue 的两大核心思想：组件化和数据驱动。组件化：把整体拆分为各个可复用的个体；数据驱动：通过数据变化直接影响 DOM，避免操作 DOM^[4]。

3、Axios

Axios 是基于 Promise 的 HTTP 请求客户端，可同时在浏览器和 node.js 中使用，可以在浏览器中发送 XMLHttpRequests 请求，具有以下功能特性：拦截请求和响应、转换请求和响应数据、自动转换 JSON 数据、客户端支持保护安全免受 XSRF 攻击^[5]。

1.2. 任务概述

1.2.1. 目标

本系统主要为了满足大多数普通用户播放音乐的需求。

用户可以在播放器网站上进行账号注册以及登陆，创建自己的歌单，往歌单里面添加歌曲，还能搜索歌曲，添加到播放列表，并且还能浏览到新歌、新 MV 和热门歌单。

系统的基本功能：

- 1、注册、登陆
- 2、创新歌单
- 3、搜索音乐
- 4、添加音乐到播放列表
- 5、音乐播放、暂停以及播放进度的控制
- 6、热门歌单、新歌、新 MV 推荐

1.2.2. 用户的特点

本平台面向普通用户，用户使用手机上网是日常行为，他们具有基本的操作技能，有播放音乐的需求，而传统手机播放器 APP 过于庞大臃肿，占用存储空间，他们渴望一个清新简约的播放器。

1.2.3. 假定和约束

假定本系统软件开发工作的开发限期三周。

采用 HTML5，基于 Vue 全家桶编写。

1.3. 需求规定

1.3.1. 对功能的规定（功能性需求）

1、用例图

（1）普通用户用例图

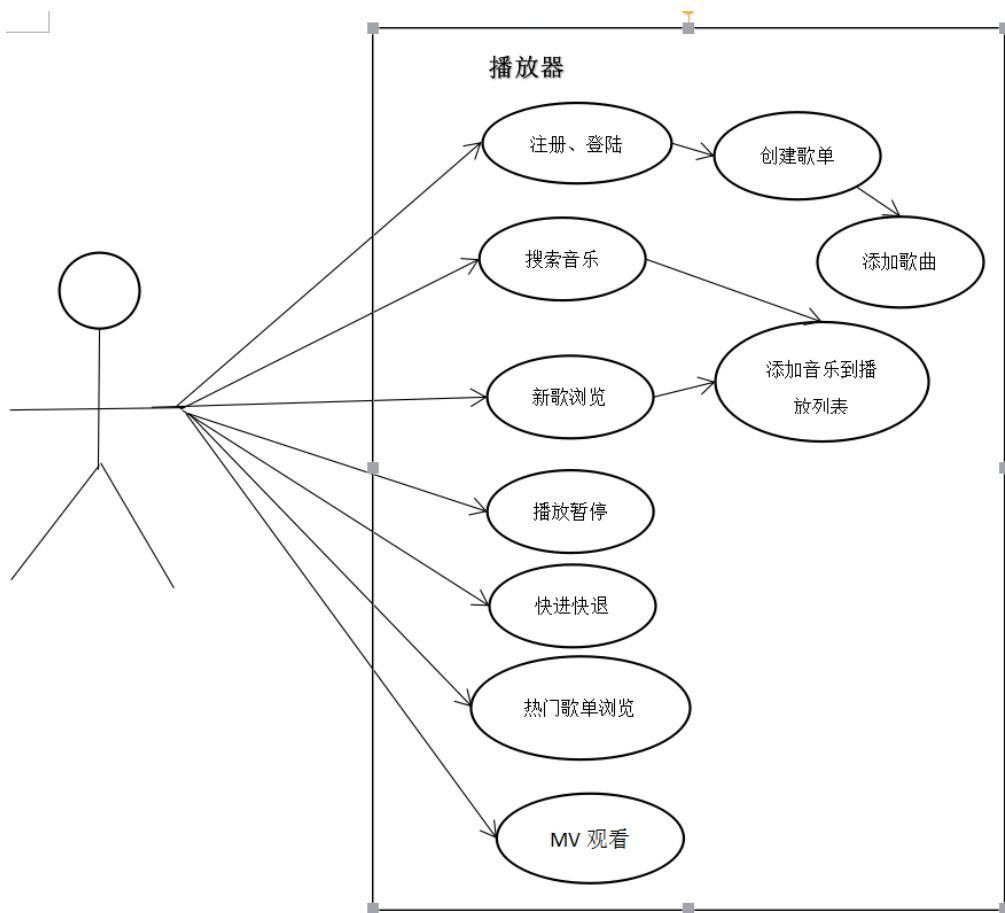


图 1.2 普通用户用例图

2、系统划分

该网站建设内容包含以下模块：

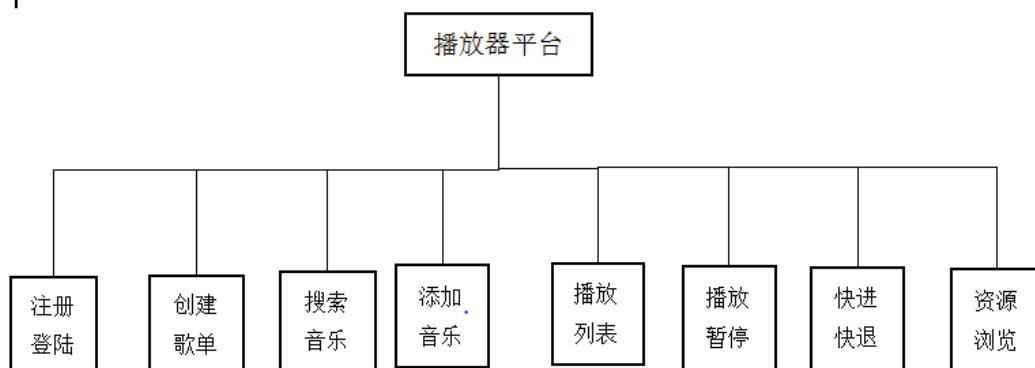


图 1.3 播放器功能模块图

3、 用户角色划分

(1) 普通用户（注册、登录的用户）

该用户可以创建歌单，添加歌曲，搜索歌曲，播放歌曲，管理播放队列，播放音乐，浏览新资源。

4、 用例分析

(1) 注册

表 1.1 用户注册信息表

信息	类型	描述
用户名	char	登陆账号
密码	char	第一次输入的密码
确认密码	char	第二次输入的秒
手机号码	char	绑定手机
验证码	char	手机接收的验证码

表 1.2 注册用例分析表

内容	说明
用例编号	C_01
用例名称	用户注册
用例说明	对应没有账号的用户， 可以进行注册
参与者	普通用户未注册的用户
前置条件	用户名唯一
后置条件	数据库存入数据
基本路径	1. 点击注册，进入注册页面 2. 填写相应数据 3. 点击确定，跳转到登陆界面
异常事件	1. 填写格式错误，显示提示相应问题

(2) 登陆

表 1.3 用户登陆信息表

信息	类型	描述
用户名或手机号码	char	登陆账号
密码	char	登陆密码
验证码	char	登陆的验证信息

表 1.4 登陆用例分析表

内容	说明
用例编号	C_02
用例名称	用户登录

用例说明	用户登录到主页面
参与者	用户
前置条件	已有注册账号
后置条件	输入的数据有效
基本路径	1. 用户输入用户名、密码、验证码，点击登陆 2. 系统验证验证码 3. 验证用户名、密码 4. 验证成功，跳转到主页面
异常事件	1. 验证码输入有误，提示验证码有误 2. 用户名，密码输入有误，提示用户名或密码有误

(3) 创建歌单

表 1.5 歌单信息表

信息	类型	描述
歌单名	char	
歌单封面（封面路径）	char	

表 1.6 创建歌单用例分析表

内容	说明
用例编号	C_03
用例名称	创建歌单
用例说明	创建歌单
参与者	用户
前置条件	用户已成功登陆网站
后置条件	用户信息成功导出到页面，创建的歌单写入数据库
基本路径	1. 用户点击个人中心 2. 系统判定用户是否已经登陆 3. 系统显示信息页面 4. 用户在信息页面，创建歌单 5. 提示创建成功
异常事件	1. 未登录用户点击，提示请登录 2. 跳转到登陆界面

(4) 添加歌曲

表 1.7 歌曲信息表

信息	类型	描述
歌名	char	
歌手	char	
路径	char	

封面（封面路径）	char	
----------	------	--

表 1.8 添加歌曲用例分析表

内容	说明
用例编号	C_04
用例名称	添加歌曲
用例说明	用户添加歌曲
参与者	普通用户
前置条件	用户已经登陆网站
后置条件	
基本路径	1. 用户点击添加到指定歌单 2. 添加歌曲到歌单中 3. 提示添加成功
异常事件	1. 未登录用户点击，提示请登录 2. 跳转到登陆界面

（5）搜索歌曲

表 1.9 歌曲信息表

信息	类型	描述
歌名	char	
歌手	char	
路径	char	
封面（封面路径）	char	

表 1.10 搜索歌曲用例分析表

内容	说明
用例编号	C_05
用例名称	搜索歌曲
用例说明	用户搜索想要听的歌曲
参与者	普通用户
前置条件	
后置条件	
基本路径	1. 输入歌曲名，点击搜索 2. 根据用户输入检索歌曲 3. 展示检索结果
异常事件	

（6）浏览新歌

表 1.11 歌曲信息表

信息	类型	描述
歌名	char	
歌手	char	
路径	char	
封面（封面路径）	char	

表 1.12 搜索歌曲用例分析表

内容	说明
用例编号	C_06
用例名称	浏览新歌
用例说明	浏览最近发行的新歌
参与者	普通用户
前置条件	
后置条件	
基本路径	1. 用户进入推荐页面 2. 网站获取新歌数据并展示
异常事件	

(7) 添加歌曲到播放列表

表 1.13 列表信息表

信息	类型	描述
歌曲信息	data	

表 1.14 添加歌曲到播放列表用例分析表

内容	说明
用例编号	C_07
用例名称	添加歌曲到播放列表
用例说明	用户搜索歌曲或浏览新歌，将想要听的歌曲添加到播放列表
参与者	普通用户
前置条件	用户搜索了歌曲或浏览新歌
后置条件	
基本路径	1. 用户点击歌曲添加到播放列表按钮 2. 将用户选中的歌曲添加到播放列表
异常事件	

(8) MV 观看

表 1.15 MV 信息表

信息	类型	描述
MV 名字	char	
封面（封面路径）	char	
歌手	data	
描述	char	对该 MV 的相关描述

表 1.16 MV 观看用例分析表

内容	说明
用例编号	C_08
用例名称	MV 观看
用例说明	观看 MV

参与者	普通用户
前置条件	
后置条件	
基本路径	1. 用户点击 MV 链接 2. 网页跳转 3. 播放 MV
异常事件	

(9) 热门歌单浏览

表 1.17 MV 信息表

信息	类型	描述
名字	char	
创建人名字	char	
封面（封面路径）	char	

表 1.18 热门歌单浏览用例分析表

内容	说明
用例编号	C_09
用例名称	热门歌单浏览
用例说明	用户浏览推荐的热门歌单
参与者	用户
前置条件	
后置条件	
基本路径	1. 用户点击歌单链接 2. 网页跳转 3. 展示歌单内容
异常事件	

(10) 播放、暂停

表 1.19 播放、暂停用例分析表

内容	说明
用例编号	C_10
用例名称	播放、暂停
用例说明	音乐播放或暂停
参与者	用户
前置条件	
后置条件	
基本路径	1. 用户点击播放或暂停按钮 2. 歌曲播放或暂停
异常事件	

(11) 快进、快退

表 1.20 快进、快退用例分析表

内容	说明
用例编号	C_11
用例名称	快进、快退
用例说明	音乐播放进度加快或减慢
参与者	用户
前置条件	当前有歌曲在播放或暂停中
后置条件	
基本路径	1.用户点击快进或快退按钮 2.歌曲播放进度加快或减慢
异常事件	当前没有歌曲在播放或暂停中，提示用户播放一首歌

1.3.2. 对性能的规定（非功能性需求）

1、网站整体要求

网站结构简单明了，页面精美。版面方面的要求,则应该设计构思力求精巧、布局要求简捷合理。

2、对性能的规定

用户输入要按照，所给的提示，信息对应。输入有误，系统要对应指出相应的问题所在。

3、时间特性要求

响应时间：

响应时间应不超 5s.

4、安全保密性要求

无

1.4. 运行环境规定

手机浏览器。

2. 概要设计

2.1. 引言

2.1.1. 编写目的

本阶段完成系统的大致设计并明确系统的数据结构与软件结构。本概要设计说明书的目的就是进一步细化软件设计阶段得出的软件概貌，把它加工成在程序细节上非常接近与源程序开发的软件表示。

2.1.2. 背景

任务提出者：学校

开发者：本人

用户：移动上网用户

2.1.3. 术语定义

逻辑结构：简称数据结构，用来实现用户观点的系统结构的表示。

实体—联系图(E-R 图):包含实体(即数据对象)、关系和属性。作为用户与分析员之间有效的交流工具^[3]。

流程图：由一些特定意义的图形、流程线以及简要的文字说明构成，能清晰明确的表示程序的运行过程^[2]。

2.2. 总体设计

2.2.1. 需求规定

系统的运行速度要快

2.2.2. 运行环境

运行在本地 node 服务器上

2.2.3. 系统流程图

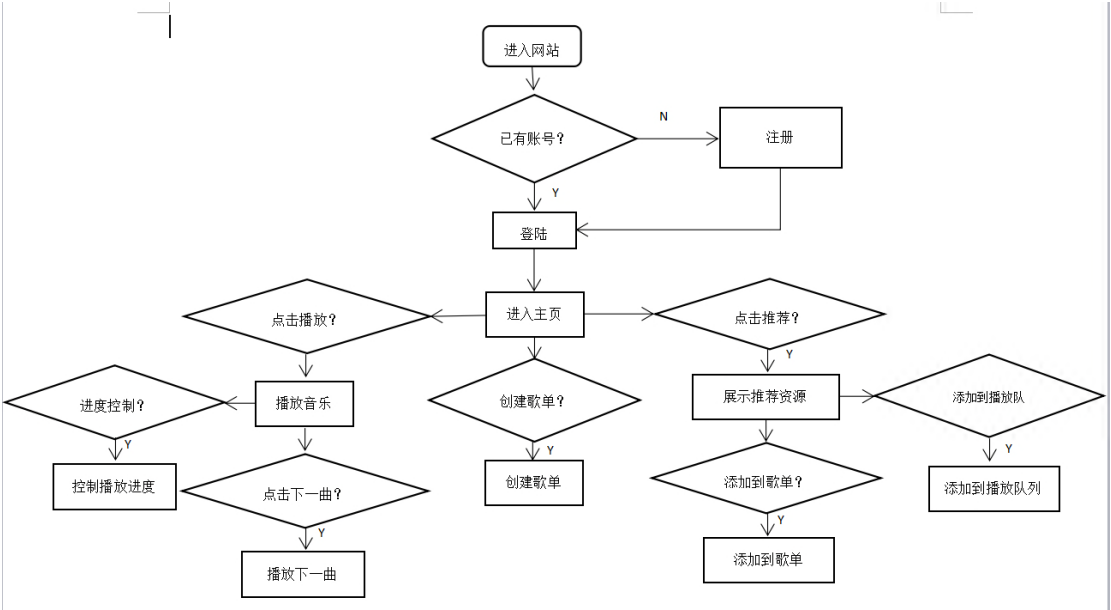


图 2.1 系统流程图

2.2.4. 系统功能模块图

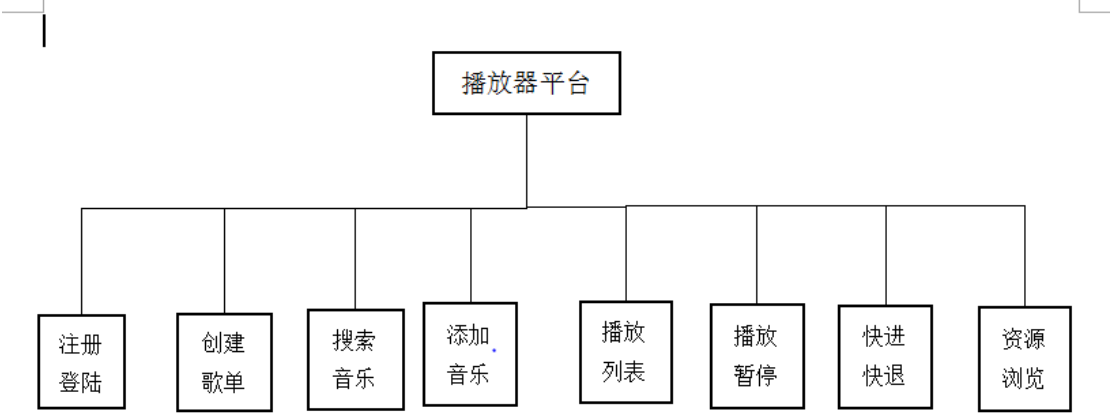


图 2.2 播放器功能模块图

2.2.5. 功能模块说明

1、注册登录：用户注册自己的账号，作为登录本凭条的唯一凭证，才能进行后续相关操作；用户可以输入自己的用户名或手机号和密码，验证成功可以跳转到个人主页，使用播放器相关功能。

- 2、创建歌单：用户在个人中心，可以创建自己的歌单，并进行命名。
- 3、搜索音乐：用户在个人中心，可以通过在搜索框中输入歌名，点击搜索，系统将调用 QQ 音乐的接口，将搜索到的第一首音乐添加到播放列表，供用户聆听。
- 4、添加音乐：用户可以将心仪的音乐添加到自己创建的歌单中
- 5、播放列表管理：用户可以往播放列表中添加音乐或者将播放列表中的音乐删除。
- 6、播放控制：用户可以对音乐进行播放、暂停、快进、快退、拖拽进度条控制进度等操作。
- 7、资源浏览：用户可以在推荐页面浏览到推荐的热门歌曲、热门歌单、新发布的 MV

2.3. 系统数据结构设计

2.3.1. 逻辑结构设计要点

1、实体及属性

用户（编号，用户名，密码，手机号码）

音乐（编号，歌名，歌手，封面，路径）

歌单（编号，歌单名，封面）

2、 整体 ER 图

用户实体 ER 图

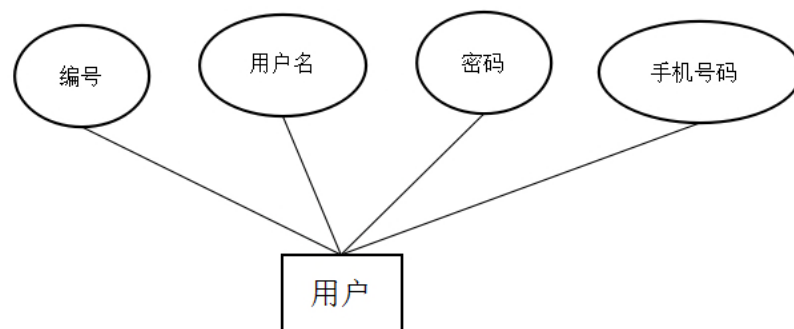


图 2.3 用户实体 ER 图

音乐实体 ER 图

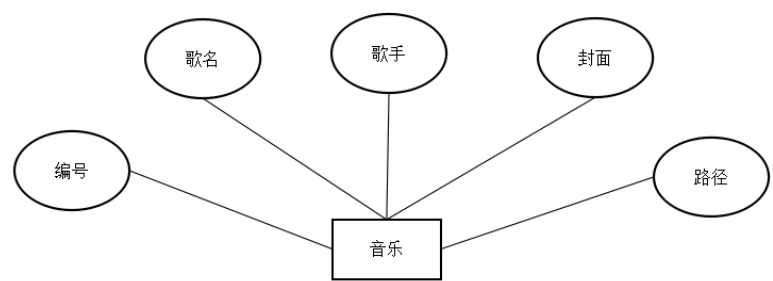


图 2.4 音乐实体 ER 图

歌单实体 ER 图

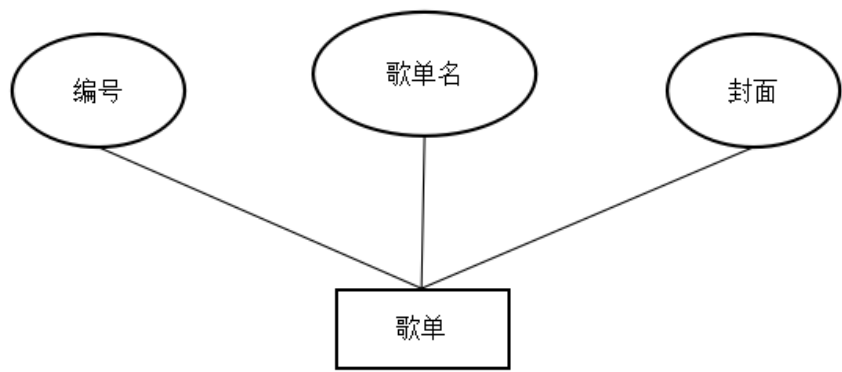


图 2.5 歌单实体 ER 图

总体 ER 图

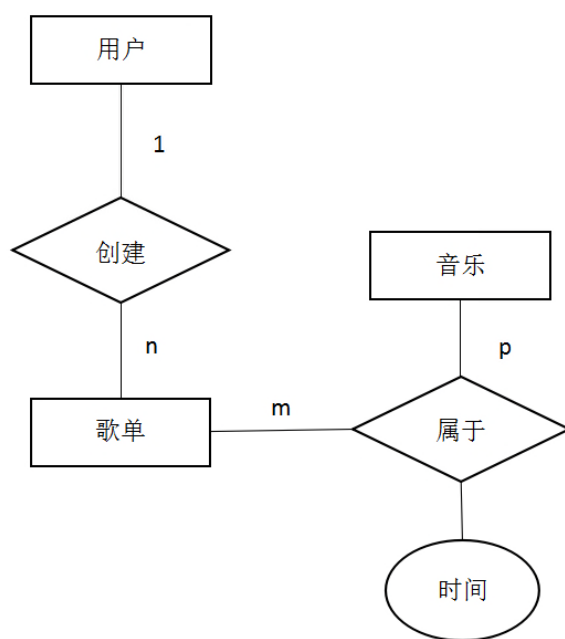


图 2.6 总体 ER 图

2.3.2. 物理结构设计要点

1、数据库文件和日志文件分盘存储

这样可以使得维护数据和检索数据速度更快。本系统数据数据文件和日志文件分别放在 D 盘和 C 盘下。

2、表和索引分开放

3、索引的建立

在音表上建立“音乐信息”非聚集索引，分盘存放，索引键列“音乐编号”“歌名”

4、备份方案

每周进行一次动态海量备份，每月进行一次静态海量备份，并妥善保存好日志文件。

2.3.3. 数据结构与程序的关系

1、数据库表设计

(1) 用户表 (user)，主键：编号

表 2.4 用户表设计

	列名	数据类型
--	----	------

用户编号	userid	int
用户名	username	varchar(12)
密码	password	varchar(12)
手机号码	tel	varchar(11)

(2) 歌单表 (disc), 主键: (编号, 用户编号)

表 2.5 歌单表设计

	列名	数据类型
用户编号	userid	int
歌单编号	discid	int
歌单名	discname	varchar(12)
封面	cover	varchar(20)

(3) 音乐表 (music), 主键: 音乐编号

表 2.6 音乐表设计

	列名	数据类型
音乐编号	musicid	int
歌名	name	varchar(12)
歌手	singer	varchar(12)
封面	cover	varchar(20)
路径	url	varchar(20)

(4) 属于表 (belong), 主键: (歌单编号, 音乐编号)

表 2.7 属于表设计

	列名	数据类型
音乐编号	musicid	int
歌单编号	discid	int
时间	time	daytype

2.4. 系统出错处理设计

2.4.1. 出错信息

1、手机丢失, 无法找回密码

2.4.2. 补救措施

1、联系网站管理员找回密码

2.4.3. 系统维护设计

要有组织、有计划地对系统进行必要的改动, 保证环境改变时, 系统处于最新最正确的工作状态, 在系统开发过程中的每个阶段都必须保存相关文档, 安排固定人员进行后期维护。

3. 详细设计

3.1. 引言

3.1.1. 编写目的

本文档编写目的是为了说明播放器各个层次中每个程序的设计考虑。把解法具体化，详细地设计每个模块，确定实现模块功能所需要的数据结构和算法。

3.1.2. 术语定义

PAD 图：一种主要用于描述软件详细设计的图形表示工具。

3.2. 界面

3.2.1. 系统登录



图 3.1 登录页面设计图

功能说明：

该界面为登录界面，提供三种登录方式，第一种是账号加密码，第二种是短信发送验证码，第三种是第三方账号登陆；若是还没有账号，则可以点击“快速注册”，进入注册页面；该页面还提供找回密码的服务，用户忘记密码的时候，可以点击“忘记密码”进行密码修改。

3.2.2. 注册



图 3.2 注册页面设计图

功能说明：

该页面为注册页面，用户输入手机号码和密码完成注册，也可以通过第三方账号注册，注册成功就跳转到登陆页面。

3.2.3. 个人中心



图 3.3 个人中心页面设计图

功能说明：

此界面为个人中心页面，用户可以创建歌单，播放音乐，搜索音乐。播放音乐的时候，可以控制音乐的播放进度，可以快进或快退，也可以拖拽进度条改变播放进度，能点击下一曲按钮播放下一首歌，也可以打开播放列表，选择喜欢的歌曲进行播放。

3.2.4. 推荐



图 3.4 推荐页面设计图

功能说明：

此两个页面是推荐页面，展示系统推荐的新歌以及热门歌单，用户可以将心仪的歌曲添加到播放列表，浏览热门歌单。

3.2.5. MV 推荐



图 3.5 MV 推荐页面设计图

功能说明：
此页面为新 MV 展示页面，展示 6 类 MV，用户可以点击 MV 链接观看 MV

3.3. 系统划分

该网站建设内容包含以下模块：

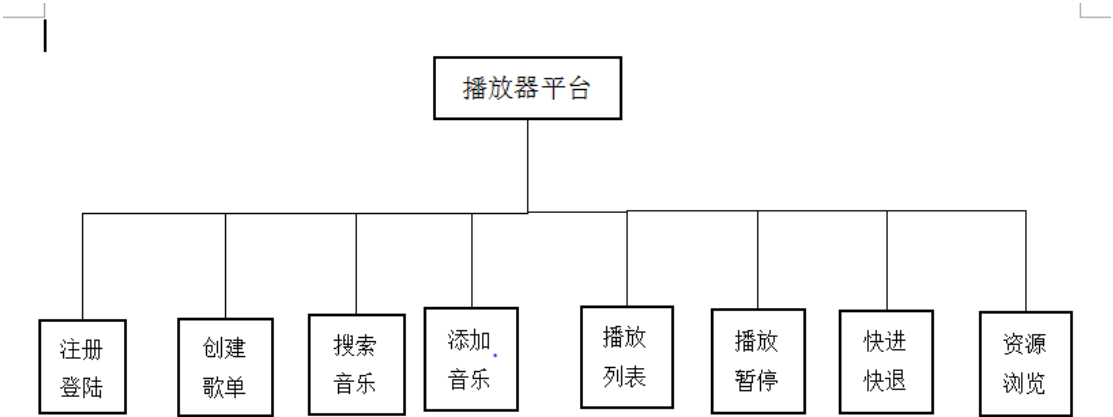


图 3.6 系统功能模块图

3.4. 功能模块详细设计

3.4.1. 注册

1、功能描述

用户填写相关数据，提交到后台写入数据库，作为用户登陆网站的凭证。

模块名：注册	模块号：001
上层调用模块：无	下层调用模块：登陆
处理过程：用户输入手机号、密码、确认密码，过滤不合法输入，将数据封装，加密，传输到服务端，写入数据库，注册成功就跳转到登陆页面；	

2、输入项

手机号、密码、确认密码。

3、输出项

登陆成功或者出错信息。

4、流程图

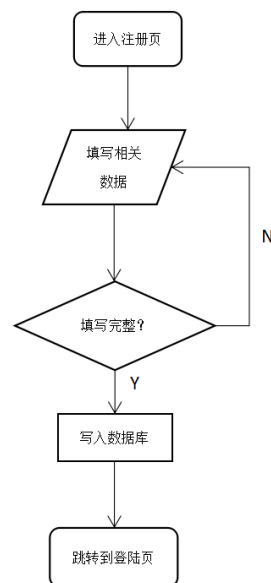


图 3.7 注册流程图

3.4.2. 登陆

1、功能描述

用户填写用户名、密码，提交，验证通过，登入网站。

模块名：登陆	模块号：002
上层调用模块：无	下层调用模块：无
处理过程：用户选择登陆方式，填写相关数据，提交，如果验证通过，就登入网站，进入个人中心；如果验证失败，就提示用户登陆失败，重新填写数据，重新验证。	

2、输入项

用户名，密码或者手机号，验证码

3、输出项

登陆成功提示或者出错提示

4、流程图

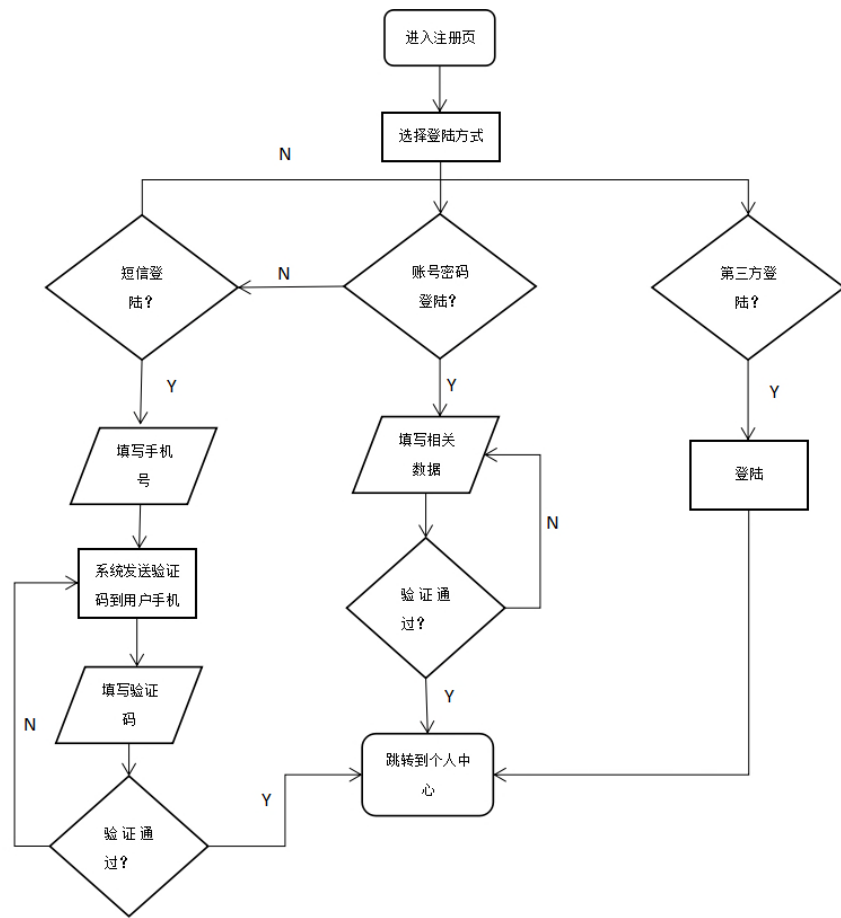


图 3.8 登录流程图

3.4.3. 添加歌单

1、功能描述

用户点击创建歌单按钮，输入歌单名，创建属于自己的歌单。

模块名：添加歌单	模块号：003
----------	---------

上层调用模块：无	下层调用模块：无
处理过程：用户点击 3 “+” 按钮，输入歌单名，创建歌单。	

2、输入项

歌单名。

3、输出项

创建成功，新歌单。

4、流程图

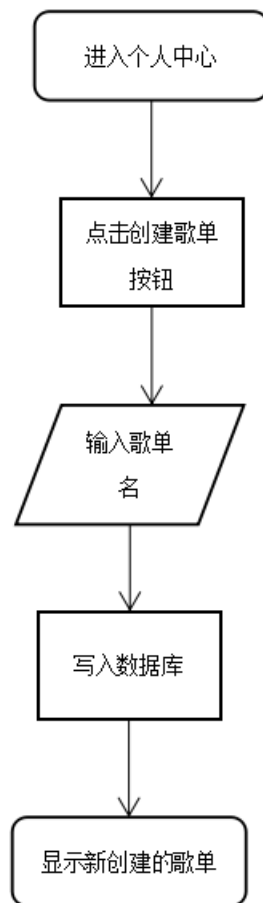


图 3.9 创建歌单流程图

3.4.4. 搜索音乐

1、功能描述

用户在搜索框中输入歌曲或 MV 名字，系统查找数据库，将查找结果展示给用户。

模块名：搜索音乐	模块号：004
上层调用模块：无	下层调用模块：无
处理过程：用户在个人中心搜索框中输入歌曲或 MV 的名字，点击搜索按钮，系统查找数据库，如果有结果，将查找结果展示给用户；如果没有结果，就提示用户暂无查询结果，请输入重新搜索。	

2、输入项

歌名或 MV 名。

3、输出项

查找成功后暂无结果，展示查找结果。

4、流程图

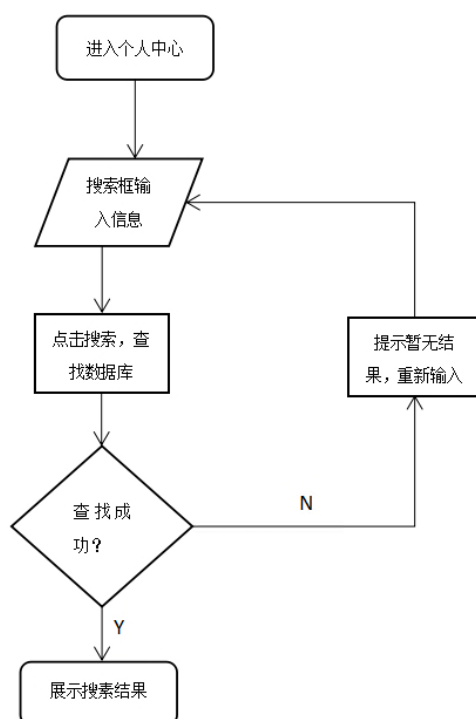


图 3.10 搜索音乐流程图

3.4.5. 添加歌曲

1、功能描述

用户根据搜索结果或系统推荐音乐，将歌曲添加到指定的歌单。

2、输入项

歌曲信息。

3、输出项

添加成功。

4、流程图

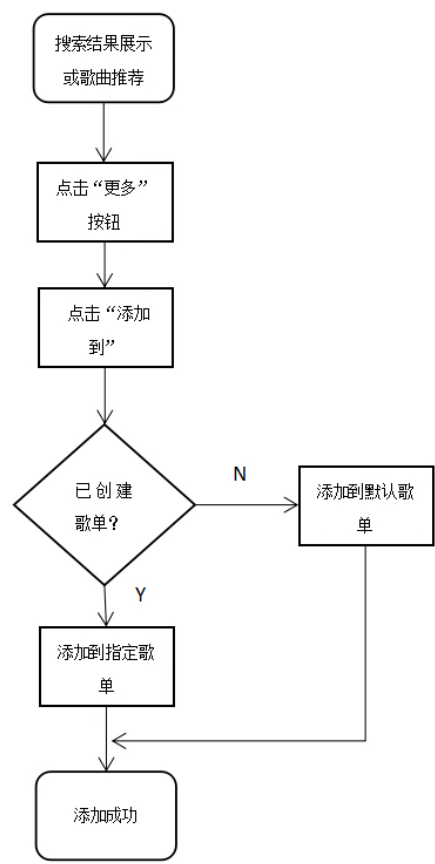


图 3.11 添加歌曲流程图

3.4.6. 添加歌曲到播放列表

1、功能描述

将心仪的歌曲添加到播放列表。

模块名：添加歌曲到播放列表	模块号：006
上层调用模块：搜索音乐，歌曲推荐	下层调用模块：无
处理过程：在搜索结果展示或歌曲推荐页面，用户点击“+”按钮，将指定的歌曲添加到播放列表，由于 Vue 是响应式的，添加成功会立马渲染到 DOM，将结果展示给用户	

2、输入项

无

3、输出项

添加成功

4、流程图



图 3.12 添加歌曲到播放列表流程图

3.4.7. 歌曲推荐

1、功能描述

用户进入歌曲推荐页，系统展示推荐的歌曲。

模块名：歌曲推荐	模块号：007
上层调用模块：	下层调用模块：无
处理过程：用户进入歌曲推荐页，系统展示推荐的歌曲	

2、输入项

无

3、输出项

推荐歌曲列表

4、流程图

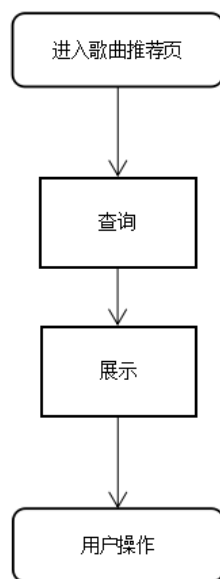


图 3.13 歌曲推荐流程图

3.4.8. 进度控制

1、功能描述

模块名：进度控制	模块号：008
上层调用模块：无	下层调用模块：无
处理过程：用户点击播放按钮，播放音乐；点击暂停按钮，暂停播放；点击快进按钮，更改当前播放时间（curTime），快进 6s，Vue 是数据驱动的，相应的进度条长度也会改变，；点击快退按钮，快退 6s；拖拽进度条，改变进度。	

2、输入项

无

3、输出项

进度控制结果

4、流程图

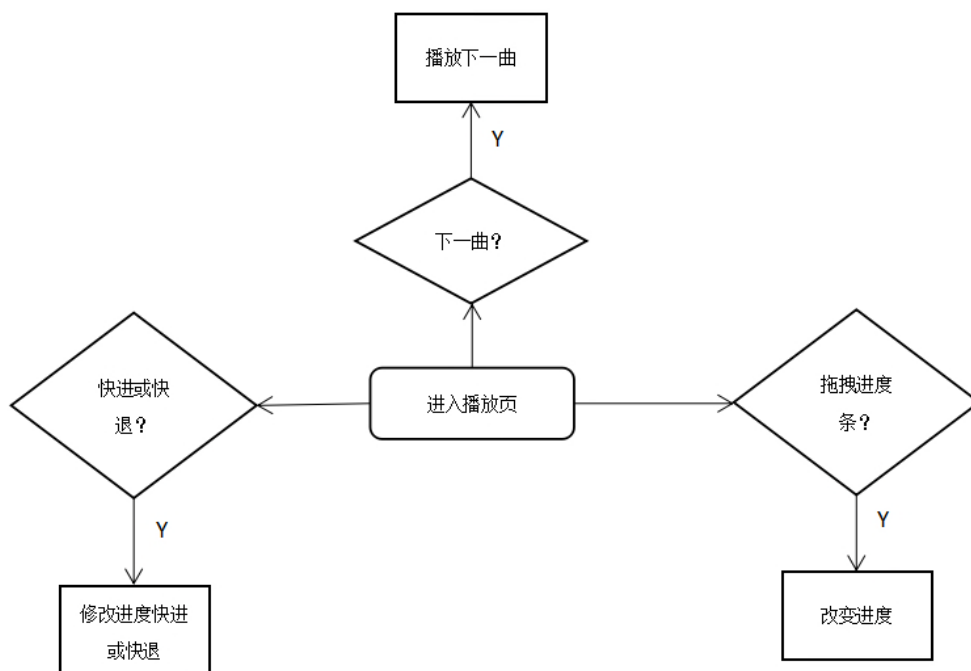


图 3.14 歌曲控制流程图

3.5. 全局数据结构

3.5.1. Vuex 状态管理

isLogin: 记录登陆状态;

playQueue[]: 播放列表

3.5.2. 全局路由卫士

当路由转到个人中心时，全局路由判断用户是否已经登陆，如果已经登陆，就允许跳转到个人中心，否则跳转到登陆页面。

4. 测试

4.1. 测试概述

4.1.1. 编写目的

本测试报告为本系统的测试报告，目的在于总结测试阶段的测试情况以及分析测试结果，描述系统是否符合用户需求，是否已达到用户预期的功能目标，并对测试质量进行分析。

测试报告参考文档提供给用户、测试人员、开发人员、项目管理者、其他管理人员和需要阅读本报告的高层经理阅读。

4.1.2. 测试范围

测试主要根据用户需求说明书和软件需求规格说明书以及相应的文档进行系统测试，包括功能测试、性能测试、安全性和访问控制测试、用户界面测试以及兼容性测试等，而单元测试和集成测试由开发人员来执行。

主要功能包括：用户登录、播放音乐。

4.2. 测试计划执行情况

4.2.1. 测试类型

测试类型	测试内容	测试目的	所用的测试工具和方法
功能测试	1、用户登陆 2、搜索音乐 3、进度控制	核实所有功能均已正常实现，即可按用户的需求使用系统。	采用黑盒测试方法，进行手工测试

测试人员

职务	姓名	E-mail	电话
开发工程师	***	***	***
测试人员	***	***	***

4.2.2. 测试问题总结

在整个系统测试执行期间，项目组开发人员高效地及时解决测试人员提出的各种缺陷，在一定程度上较好的保证了测试执行的效率以及测试最终期限

4.3. 测试总结

4.3.1. 测试用例执行结果

测试用例标识符	测试用例名称	状态	测试结果	备注
功能测试				
Testcase001	登录回车验证	已执行	测试通过	登录
Testcase002	登陆按钮验证	已执行	测试未通过	
Testcase003	二次登陆验证	已执行	测试未通过	
Testcase004	播放原列表歌曲	已执行	测试通过	播放音乐
Testcase005	播放新添加歌曲	已执行	测试未通过 通过	

4.3.2. 测试问题解决

测试用例标识符	错误或问题描述	错误或问题状态
Testcase002	在密码用户名都正确时，要点击两次登陆按钮才能登陆成功	已解决
Testcase003	登陆成功后，返回登陆，输入其他账号，在密码用户名都不正确时，还可以成功登陆	已解决
Testcase005	播放新添加的歌曲不成功	已解决

参 考 文 献

- [1] 阮一峰，深入响应式原理.
- [2] 张海潘，牟永敏，软件工程导论.
- [3] 王珊，萨师煊，数据库系统导论
- [4] 尤雨溪，Vue 2.0 渐进式前端解决方案
- [5] 科技帅戈，axios 使用详解