|  |  |
| --- | --- |
| 231 | 123 |
| **PINGDINGSHAN UNIVERSITY** |

**毕业设计(论文)**

**题 目:**基于SSM的影视推荐系统的设计

    与实现

**院(系):** 计算机学院（软件学院）

**专业年级:** 软件工程 2017级

**姓 名:** 杜嘉伟

**学 号:** 171530206

**指导教师:** 张国平 副教授

**2020年04月26日**

原 创 性 声 明

本人郑重声明：本人所呈交的毕业论文，是在指导老师的指导下独立进行研究所取得的成果。毕业论文中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究成果做出重要贡献的个人和集体，均已在文中以明确方式标明。

本声明的法律责任由本人承担。

论文作者签名：   日 期：

关于毕业论文使用授权的声明

本人在指导老师指导下所完成的论文及相关的资料（包括图纸、试验记录、原始数据、实物照片、图片、录音带、设计手稿等），知识产权归属平顶山学院。本人完全了解平顶山学院有关保存、使用毕业论文的规定，同意学校保存或向国家有关部门或机构送交论文的纸质版和电子版，允许论文被查阅和借阅；本人授权平顶山学院可以将本毕业论文的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业论文。如果发表相关成果，一定征得指导教师同意，且第一署名单位为平顶山学院。本人离校后使用毕业论文或与该论文直接相关的学术论文或成果时，第一署名单位仍然为平顶山学院。

论文作者签名： 日 期：

指导老师签名： 日 期：

基于SSM的影视推荐系统的设计与实现

摘 要

近年来,随着网络智能平台的迅速发展,用户可以更方便地观看或下载网络影视作品。然而,面对网络中百万级的影视节目内容,如何准确的表达用户偏好,为用户从海量数据中推荐其想要的视频节目,仍无较好的解决方案。对于商家来说,如何针对不同用户推荐合适的影视节目,对于提高节目收视率及购买率也具有重要的作用。面对海量的影视作品，服务于用户去选择的影视推荐系统就显得尤为重要

本文介绍了影视推荐系统的设计和实现，主要包含五大模块，用户信息管理模块、电影信息管理模块、热片推荐管理模块、讨论帖管理模块和团约管理模块。使用QQ与微信接口，便于用户登陆，系统采用B/S 模式设计，无需用户下载客户端，选取了SpringBoot、Spring、MyBatis框架技术作为应用开发平台后端的支撑，前端采用Umi框架、React、Ant Design、TypeScript等新型技术实现，采用组件化思想前后端分离，研究了通用权限系统的RBAC模型，控制应用开发平台的功能权限。对选取的前后端框架技术进行集成，使用MySQL作为数据库保证了数据的安全和系统的性能。

本论文阐述了基于SSM的影视推荐系统的开发背景、研究现状及发展趋势，介绍了系统开发所使用的环境、工具、开发语言以及相关的开发技术。按照软件开发的生命周期论述了需求分析、系统分析、概要设计、详细设计和代码实现的过程。

该系统经测试，各项功能运行稳定，性能良好，界面设计合理友好，简单易操作。通过这样一个影视推荐系统，有利于用户更快更简洁的获取到自己想看的影视作品

**关键词：**影视推荐系统，Umi框架，React

Design and implementation of film recommendation system based on SSM

Abstract

In recent years, with the rapid development of network intelligent platforms, users can more easily watch or download online movies and TV works. However, there is still no good solution for how to accurately express users' preferences and recommend the videos they want from the mass data in the face of millions of TV programs on the Internet. For businesses, how to recommend appropriate movie and television programs to different users also plays an important role in improving program viewing rate and purchase rate. In the face of massive film and television works, the film and television recommendation system that serves users to choose becomes particularly important

This paper introduces the design and implementation of the film recommendation system, which mainly includes five modules: user information management module, film information management module, hot film recommendation management module, discussion post management module and group appointment management module. Use QQ with WeChat interface, making it easy for users to log in, system adopts B/S pattern Design, without user download client, select the SpringBoot, Spring, MyBatis - Plus framework technology as the backend application development platform support, front end USES Umi frame, React, Ant Design, TypeScript, such as new technology, the idea of modular separation before and after the end, to study the general permission system of RBAC model, control the function of the application development platform. Integrate the front and back end framework technology and use MYSQL as the database to ensure the security of data and the performance of the system.

This paper describes the development background, research status and development trend of the film and television recommendation system based on SSM, and introduces the environment, tools, development language and related development technologies used in the system development. According to the life cycle of software development, the process of requirement analysis, system analysis, outline design, detailed design and code realization is discussed.

The system has been tested, the function is stable, the performance is good, the interface design is reasonable and friendly, simple and easy to operate. Through such a film and television recommendation system, it is beneficial for users to get the film and television works they want to watch more quickly and succinctly

**Keywords: Video Recommendation System, Umi Framework, React**

**目 录**

[1 绪论 1](#_Toc35977231)

[1.1 项目开发背景意义 1](#_Toc35977232)

[1.2 研究现状 2](#_Toc35977233)

[1.3 论文的结构安排 2](#_Toc35977234)

[1.4 本章小结 3](#_Toc35977235)

[2 相关技术的介绍 4](#_Toc35977236)

[2.1 SpringBoot 4](#_Toc35977237)

[2.2 MyBatis-Plus 4](#_Toc35977238)

[2.3 FreeMarker 5](#_Toc35977239)

[2.4 Shiro 5](#_Toc35977240)

[2.5 Layui 5](#_Toc35977241)

[2.6 Redis 6](#_Toc35977242)

[2.7 本章小结 6](#_Toc35977243)

[3 需求分析 7](#_Toc35977244)

[3.1 业务描述 7](#_Toc35977245)

[3.2 系统设计目标 8](#_Toc35977246)

[3.3 用户需求描述 8](#_Toc35977247)

[3.4 性能需求 15](#_Toc35977248)

[3.5 本章小结 16](#_Toc35977249)

[4 总体设计 17](#_Toc35977250)

[4.1 系统结构设计 17](#_Toc35977251)

[4.2 软件功能设计 20](#_Toc35977252)

[4.3 数据库设计 22](#_Toc35977253)

[4.3.1 数据库概念结构设计 22](#_Toc35977254)

[4.3.2 数据库逻辑结构设计 28](#_Toc35977255)

[4.3.3 数据库表的结构设计 28](#_Toc35977256)

[4.4 本章小结 32](#_Toc35977257)

[5 详细设计与实现 33](#_Toc35977258)

[5.1 功能设计 33](#_Toc35977259)

[5.1.1 用户登录模块的设计 33](#_Toc35977260)

[5.1.2 用户管理模块的设计 40](#_Toc35977261)

[5.1.3 角色管理模块的设计 46](#_Toc35977262)

[5.1.4 菜单管理模块的设计 54](#_Toc35977263)

[5.1.5 部门管理模块的设计 60](#_Toc35977264)

[5.1.6 在线用户模块的设计 65](#_Toc35977265)

[5.1.7 系统日志模块的设计 68](#_Toc35977266)

[5.1.8 代码生成模块的设计 72](#_Toc35977267)

[5.2 系统的界面和接口说明 78](#_Toc35977268)

[5.3系统的具体实现 79](#_Toc35977269)

[5.3.1 用户登录模块的实现 79](#_Toc35977270)

[5.3.2 用户管理模块的实现 81](#_Toc35977271)

[5.3.3 角色管理模块的实现 82](#_Toc35977272)

[5.3.4 菜单管理模块的实现 84](#_Toc35977273)

[5.3.5 部门管理模块的实现 85](#_Toc35977274)

[5.3.6 在线用户模块的实现 87](#_Toc35977275)

[5.3.7 系统日志模块的实现 88](#_Toc35977276)

[5.3.8 代码生成模块的实现 89](#_Toc35977277)

[5.4 本章小结 90](#_Toc35977278)

[6 测试 91](#_Toc35977279)

[6.1 测试的意义 91](#_Toc35977280)

[6.2 测试环境 91](#_Toc35977281)

[6.3 功能测试 91](#_Toc35977282)

[6.3.1 用户登录测试 92](#_Toc35977283)

[6.3.2 修改用户信息测试 93](#_Toc35977284)

[6.3.3 删除角色信息测试 93](#_Toc35977285)

[6.3.4 查询菜单信息测试 94](#_Toc35977286)

[6.3.5 新增部门信息测试 94](#_Toc35977287)

[6.3.6 新增系统日志测试 95](#_Toc35977288)

[6.3.7 查询系统日志测试 96](#_Toc35977289)

[6.3.8 代码生成测试 97](#_Toc35977290)

[6.4 WeTest页面测试 97](#_Toc35977291)

[6.5 服务器响应速度测试 98](#_Toc35977292)

[6.6 本章小结 99](#_Toc35977293)

[7 结束语 101](#_Toc35977294)

[7.1 总结 101](#_Toc35977295)

[7.2 展望 101](#_Toc35977296)

[附 录 103](#_Toc35977297)

[参考文献 106](#_Toc35977298)

[致 谢 107](#_Toc35977299)

1 绪论

随着计算机的普及和互联网的发展，越来越多的影视作品进入了人们的生活，如何从数以百万计的作品中挑选出自己喜欢的作品成为每个人的难题，影视推荐系统就应运而生。本系统为所有热爱观影的用户提供方便，对所有作品进行筛选。本章介绍了项目开发的背景和意义，基础应用开发平台的研究现状，最后对论文的结构进行了阐述。

## 1.1 项目开发背景意义

在当今社会，人类科技技术正在飞速发展，一方面，互联网已经进入人类生活的方方面面，社会进入网络时代，计算机已经成为社会发展的强大动力。另一方面，人们的电影拍摄技术也在不断的进步，越来越多的电影使用眼花缭乱的特效，因此人们对影视作品提出了越来越高的要求。推荐系统的出现，来自于一个很普遍的现象：人在做一个决策的时候，往往会参考或者依赖别人的建议，例如在商场买服饰时会参考导购的推荐；去书店买书时会依赖老师或者同学的推荐。同理，观看影视作品时，人们大多也会倾向于浏览评分和评论的信息。因此，影视推荐系统就应运而生。

为节约寻找优质影视作品的时间、减轻用户在网上盲目寻找这类问题，开发与设计一个影视资源推荐系统，可以实现低成本、网络化、高性能的网络线上影视资源详细查询。用户可以根据自己的具体需要，选择查看具体的影视类型或者相关方面的作品，在网络上通过影视推荐，减少了用户费时费力去找优质作品的过程，极大的提升了用户的观看体验。本平台还提供了讨论帖功能，用户可以对自己喜欢的影视作品发起讨论帖，方便兴趣相同的用户在一起交流，让更多喜欢此作品的人参与讨论，一方面可以提高该作品的热度，另一方面拓宽了用户的交际圈。

推荐系统中的推荐技术主要包括基于内容的过滤和协同过滤两种。协同过滤推荐技术是推荐系统中应用最早和最为成功的技术之一。近几年电子商务的快速发展推动了推荐系统的发展，诸如互联网上最大的书店Amazon.corn,Web.上最大的CD商店CDNow.corn,全球最大的C2C网站一淘宝网等等。由微软研究院开发的协同过滤工具己被集成在微软的Commerce Server产品中，并被许多站点使用。其运作模式逐渐转变为跨地区、跨领域的扁平化管理，从而获得了显著的经济和社会效益。但仍旧不能对提高效率、降低成本等方面起到至关重要的作用。本课题正是基于如何开发简单高效的影视推荐管理系统，力求尽量的简洁实用地解决问题。

## 1.2 研究现状

Java技术跨平台，拥有极大的广泛性，近年来不断推出各种支持Java EE应用开发的技术、以及相关的框架的技术，例如：SpringBoot、SpringMVC、Spring、MyBatis等。这些基于Java EE应用的优秀开源框架各自都拥有自身的特点，适用于不同的场景。可以根据自身的的实际需求来决定在开发Web应用时所选择的技术。

在前端方面，经历过三种变革，从一站式开发，所有代码杂合在一起到后来的前后端分离，再到现如今的组件式开发，前端的技术变更迅速，如前端框架React、Vue，采用MVVM思想，MVVM是Model-View-ViewModel的简写。视图模型mvvm模式的核心，它是连接view和model的桥梁。模型指的是后端传递的数据。视图指的是所看到的页面。

它有两个方向：  
一是将【模型】转化成【视图】，即将后端传递的数据转化成所看到的页面。实现的方式是：数据绑定。二是将【视图】转化成【模型】，即将所看到的页面转化成后端的数据。实现的方式是：DOM 事件监听。这两个方向都实现的，我们称之为数据的双向绑定。  
 MVVM的框架下视图和模型是不能直接通信的。它们通过ViewModel来通信，ViewModel通常要实现一个observer观察者，当数据发生变化，ViewModel能够监听到数据的这种变化，然后通知到对应的视图做自动更新，而当用户操作视图，ViewModel也能监听到视图的变化，然后通知数据做改动，这实际上就实现了数据的双向绑定。并且MVVM中的View 和 ViewModel可以互相通信。

目前国内基于JavaEE 应用框架的整合开发架构的典型代表是EasyJWeb。用于JavaEE应用程序的MVC框架。实现了页面及代码完全分离的mvc开发框架。但Facebook认为MVC无法满足他们的扩展需求，由于他们非常巨大的代码库和庞大的组织，使得MVC很快变得复杂，每当需要添加一项新功能或者特性时，系统的复杂就成级数的增长，致使代码变得脆弱而不可预测，结果导致他们的MVC正在土崩瓦解。认为MVC不适合大规模的应用。当系统中有很多模型和相应的视图时，其复杂度就会迅速扩大，非常难以理解和调试，特别是模型和视图可能存在双向数据流动。  
 解决这个问题需要“以某种方式组织代码，使其更加可预测”，这通过Flux和React已经完成

本文借鉴了国内成熟的集成框架的思想和方法，采用了当前国内较为流行的SpringBoot、MyBatis框架技术作为程序后台设计的支撑，以及全球范围内较为前沿的React前端框架作为前台设计的基石，结合这些开源框架技术的特点和优势，搭建出一个具有较好扩展性和通用性的基础架构。

## 1.3 论文的结构安排

本论文主要介绍了系统进行设计开发的原因，以及系统的需求分析、总体设计、详细设计和功能 测试。整篇论文分为七章来写，具体如下：

第1章 绪论，简述项目开发的背景意义、项目的研究现状以及论文的结构安排。

第2章 相关技术的介绍，简述项目中使用到的关键技术。

第3章 需求分析，通过相关的业务描述、系统设计目标给出各部分的相关设计。

第4章 总体设计，根据需求分析进行了软件功能设计，并对数据库设计进行详细介绍。

第5章 系统的详细设计与实现，列举了系统中的重要模块，对每个模块的功能进行描述。

第6章 测试，分析测试出来的结果。

第7章 结语部分从总结和展望两方面叙述，总结了系统的实现过程，提出了系统的不足。

## 1.4 本章小结

本章主要是对项目开发的背景意义、项目的研究现状、论文的结构安排的简单描述，在背景中主要介绍了前端框架的开发历程，通过背景意义来说明本系统开发的意义以及技术的进步，在论文结构安排中介绍整篇论文的安排，主线清晰，便于阅读。

2 相关技术的介绍

## 2.1 React

React 起源于 Facebook 的内部项目，因为该公司对市场上所有 JavaScript MVC 框架，都不满意，就决定自己写一套，用来架设 Instagram 的网站。由于 React 的设计思想极其独特，属于革命性创新，性能出众，代码逻辑却非常简单。ReactJS是一套JavaScript Web库，由Facebook打造而成且主要用于构建高性能及响应式用户界面。React负责解决其它javascript框架所面对的一大常见难题，即对大规模数据集的处理。能够使用虚拟DOM并在发生变更时利用补丁安装机制只对DOM中的dirty部分进行重新渲染，React得以实现远超其它框架的速度表现。

1. 速度很快，它并不直接对DOM进行操作，引入了一个叫做虚拟DOM的概念，安插在javascript逻辑和实际的DOM之间，性能好。
2. 跨浏览器兼容。
3. 单向数据流，其实reactjs的核心内容就是数据绑定，所谓数据绑定指的是只要将一些服务端的数据和前端页面绑定好，使开发者只关注实现业务。
4. 一切都是component，代码更加模块化，重用代码更容易，可维护性高。

## 2.2 TypeScript

TypeScript是[微软](https://baike.baidu.com/item/%E5%BE%AE%E8%BD%AF/124767" \t "https://baike.baidu.com/item/typescript/_blank)开发的一个开源的[编程语言](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80/9845131" \t "https://baike.baidu.com/item/typescript/_blank)，通过在[JavaScript](https://baike.baidu.com/item/JavaScript/321142" \t "https://baike.baidu.com/item/typescript/_blank)的基础上添加静态类型定义构建而成。TypeScript通过TypeScript[编译器](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8/8853067" \t "https://baike.baidu.com/item/typescript/_blank)或Babel转译为JavaScript代码，可运行在任何[浏览器](https://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8/213911" \t "https://baike.baidu.com/item/typescript/_blank)，任何[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192" \t "https://baike.baidu.com/item/typescript/_blank)。

TypeScript扩展了[JavaScript](https://baike.baidu.com/item/JavaScript" \t "https://baike.baidu.com/item/typescript/_blank)的语法，所以任何现有的JavaScript程序可以运行在TypeScript环境中。TypeScript是为大型应用的开发而设计，并且可以编译为JavaScript。

TypeScript 支持为已存在的 JavaScript 库添加类型信息的头文件，扩展了它对于流行库的支持，如 [jQuery](https://baike.baidu.com/item/jQuery" \t "https://baike.baidu.com/item/typescript/_blank)，[MongoDB](https://baike.baidu.com/item/MongoDB" \t "https://baike.baidu.com/item/typescript/_blank)，[Node.js](https://baike.baidu.com/item/Node.js" \t "https://baike.baidu.com/item/typescript/_blank) 和 D3.js 等。这些第三方库的类型定义本身也是开源的，所有开发者都能参与贡献。

TypeScript 增加了代码的可读性和可维护性

1. 类型系统实际上是最好的文档，大部分的函数看看类型的定义就可以知道如何使用了。
2. 可以在编译阶段就发现大部分错误。
3. 增强了编辑器和 IDE 的功能，包括代码补全、接口提示、跳转到定义、重构等。

TypeScript 非常包容

1. TypeScript 是 JavaScript 的超集，.js 文件可以直接重命名为 .ts 即可。
2. 即使不显式的定义类型，也能够自动做出类型推论。
3. 可以定义从简单到复杂的几乎一切类型。
4. 即使 TypeScript 编译报错，也可以生成 JavaScript 文件。
5. 兼容第三方库，即使第三方库不是用 TypeScript 写的，也可以编写单独的类型文件供 TypeScript 读取。

## 2.3 Umi

UmiJS是一款插件化的企业级前端应用框架。基于路由，支持类似next.js的常规路由和各种高级路由功能，例如路由级别的按需加载。

它具有完整的插件系统，涵盖了从源代码到构建的每个生命周期。 Umi能够满足各种功能扩展和业务需求，目前Umi在社区和公司内部拥有近50多个插件。 Umi是蚂蚁金服的基本前端框架，已直接或间接为3000多个应用程序提供服务，包括Java、节点、移动应用程序、混合应用程序、纯前端资产应用程序、CMS应用程序等等。

1. 可扩展性，Umi 实现了完整的生命周期，并使其插件化，Umi 内部功能也全由插件完成。此外还支持插件和插件集，以满足功能和垂直域的分层需求。
2. 开箱即用，Umi 内置了路由、构建、部署、测试等，仅需一个依赖即可上手开发。并且还提供针对 React 的集成插件集，内涵丰富的功能，可满足日常 80% 的开发需求。
3. 企业级，经蚂蚁内部 3000+ 项目以及阿里、优酷、网易、飞猪、口碑等公司项目的验证，值得信赖。
4. 大量自研，包含微前端、组件打包、文档工具、请求库、hooks 库、数据流等，满足日常项目的周边需求。
5. 完备路由，同时支持配置式路由和约定式路由，同时保持功能的完备性，比如动态路由、嵌套路由、权限路由等等。
6. 面向未来，在满足需求的同时，我们也不会停止对新技术的探索。比如 modern mode、webpack@5、自动化 external、bundler less 等等。

## 2.4 Ant Design

Ant Design是蚂蚁集团体验技术部经过大量项目实践和总结，逐步打磨出一个服务于企业级产品的设计体系。基于「自然」、「确定性」、「意义感」、「生长性」四大[设计价值观](https://ant.design/docs/spec/values-cn)，通过模块化解决方案，降低冗余的生产成本，让设计者专注于更好的用户体验。

1. 原则，基于价值观衍生出的一般设计技巧，包括如何高效组织信息、帮助建立用户与界面的互动等原则性要求。
2. 全局规则，通过规范常见的互动行为，包括导航、数据录入、数据展示、反馈、操作、帮助等，呈现组件元素的标准用法和弹性空间，并了解如何将他们组织在一起来创建良好的体验。
3. 模板文档，基于蚂蚁中台的设计经验，我们总结了表单、列表、数据可视化、详情页、工作台、异常、结果、编辑器几类页面的典型解决方案，介绍每类页面的设计目标、设计技巧以及典型页面内容等，帮助广大设计者可以快速开始构建界面。

## 2.5 Yarn

Yarn的基本思想是将JobTracker的两个主要功能（资源管理和作业调度/监控）分离，主要方法是创建一个全局的ResourceManager（RM）和若干个针对应用程序的ApplicationMaster（AM）。这里的应用程序是指传统的MapReduce作业或作业的[DAG](https://baike.baidu.com/item/DAG/10765028" \t "https://baike.baidu.com/item/yarn/_blank)（有向无环图）。大大减小了 JobTracker（也就是现在的 ResourceManager）的资源消耗，并且让监测每一个 Job 子任务 (tasks) 状态的程序分布式化了，更安全、更优美。

## 2.6 Node

Node.js发布于2009年5月，由Ryan Dahl开发，是一个基于Chrome V8引擎的[JavaScript](https://baike.baidu.com/item/JavaScript/321142" \t "https://baike.baidu.com/item/_blank)运行环境，使用了一个[事件驱动](https://baike.baidu.com/item/%E4%BA%8B%E4%BB%B6%E9%A9%B1%E5%8A%A8/9597519" \t "https://baike.baidu.com/item/_blank)、非阻塞式I/O模型， 让JavaScript 运行在[服务端](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E7%AB%AF/6492316" \t "https://baike.baidu.com/item/_blank)的开发平台，它让JavaScript成为与[PHP](https://baike.baidu.com/item/PHP/9337" \t "https://baike.baidu.com/item/_blank)、[Python](https://baike.baidu.com/item/Python/407313" \t "https://baike.baidu.com/item/_blank)、[Perl](https://baike.baidu.com/item/Perl/851577" \t "https://baike.baidu.com/item/_blank)、[Ruby](https://baike.baidu.com/item/Ruby/11419" \t "https://baike.baidu.com/item/_blank)等服务端语言平起平坐的[脚本语言](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80/1379708" \t "https://baike.baidu.com/item/_blank)。

Node.js对一些特殊用例进行优化，提供替代的[API](https://baike.baidu.com/item/API/10154" \t "https://baike.baidu.com/item/_blank)，使得V8在非浏览器环境下运行得更好，V8引擎执行Javascript的速度非常快，性能非常好，基于Chrome JavaScript运行时建立的平台， 用于方便地搭建响应速度快、易于扩展的[网络应用](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%BA%94%E7%94%A8/2196523" \t "https://baike.baidu.com/item/_blank)。

## 2.7 本章小结

本章内容主要围绕快速Web应用开发平台的开发的相关技术来进行介绍，分别阐述了React、TypeScript、UmiJS、Ant Design、Yarn、Node等技术的含义与用法，方便设计过程的使用。

3 需求分析

为了满足用户所有需求，我们首先就需要知道用户真正想要的是什么、需求分析的主要目的就是去分析和思考我们系统软件将要实现的功能，深入分析用户的心理，明确系统要实现的具体功能，需要有大局意识，在此基础上节约开发成本，提高开发效率。这一章我们共分析平台的功能性需求和非功能性需求两部分，功能性需求主要是介绍系统需要实现哪些功能模块，实现哪些具体的操作等。非功能需求指的是软件的运行环境需求、系统运行的效率。通过研究这两个方面的内容，让我们对系统又一个清楚的认知，以便我们在开发完成后验证系统功能的完善与否。

## 3.1 业务描述

基于基于SSM的影视推荐系统从功能上分，主要有两个角色，管理员和一般用户。也可以根据后续的开发添加新的角色。

管理员的主要作用负责一般用户管理模块、影视资源管理模块、评论管理模块、讨论帖管理模块和团约管理模块，这五大模块组成了开发平台中的操作权限和数据权限。在用户管理中，因为本平台采用QQ、微信、支付宝授权等方式登陆，所以管理员并不拥有创建用户的权力，但有对某个账号违规信息进行封禁的权限，管理员也可以查看所有用户的详细信息，但不能对其进行修改。在电影管理中，管理员可以查看所有影视资源的详细信息，并拥有对影视资源进行上架、下架、修改的权限。在评论管理模块中，管理员可以看到所有用户的发言，并拥有对其删除的权限。在讨论帖管理模块中，管理员可以看到所有讨论帖，以及每个讨论帖的发起人的详细信息，每个参与的人发表的言论，并对其拥有删除的权限。在团约管理中，管理员能够查看到所有已发起的团约和已结束的团约，以及每个团约的发起人和参与用户的详细信息，并对每个团员拥有删除的权限。

一般用户在此系统可以查看个人信息，多数为授权平台上存储的信息，如头像、授权平台的登陆账号id等，一般用户也可以浏览所有的影视资源信息以及相关影视资源下的评论，也可以发表自己的评论。可以查看所有的讨论帖内容，并查看所有发表言论的用户的部分信息，可以查看所有已发起的团约以及查看团约内的用户，在授权登陆后可以对相关影视资源进行评论、在讨论帖中回复与别人交流以及参加或发起团约。

## 3.2 系统设计目标

本系统是基于React、TypeScript、UmiJS、Ant Design、Yarn、Node、SpringBoot、MyBatis技术所设计开发的影视推荐系统。希望能做到开发快速、系统稳定、设计合理，在此基础上设计优美、运行流畅，给用户好的使用体验，帮助用户更快的寻找优质影视资源，增加效率。

系统的具体功能实现的目标如下：

1.考虑到安全性，采用QQ、微信、支付宝授权的方式登陆，确保留有每个人正确的联系方式。

2.管理员拥有对所有对象删除的权限，但是能对所有的对象都拥有修改的权限，确保对不干涉用户的信息。

3.服务器开启日志检测，详细的记录每个用户的操作痕迹，用文件的方式储存，并实时监控。

## 3.3 用户需求描述

本影视推荐系统供系统管理员和一般用户两种人员使用，这两种人员都采用QQ、微信、支付宝等具有官方认证的平台授权登陆，即使不登陆，也可以使用本系统的查看功能。系统管理员拥有整个系统的最高权限，可以对用户进行封禁操作，对影视资源进行上架、下架、更新、查询等操作，对团约和讨论帖拥有删除操作。一般用户可以对自己的个人信息进行修改操作，对影视资源以及讨论帖拥有查看以及评论操作，对团约有发起、加入、退出、查看等操作。

系统管理员的用例图，如图3-1所示。

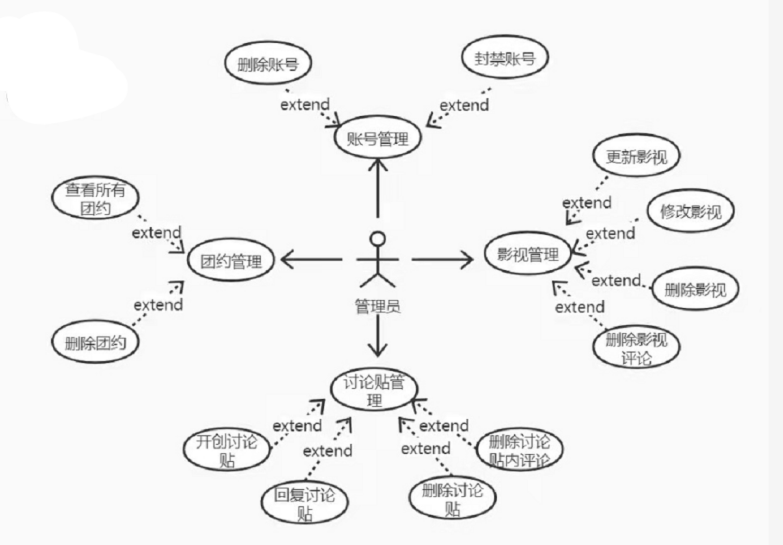


图3-1 系统管理员用例图

一般用户用例图，如图3-2所示。

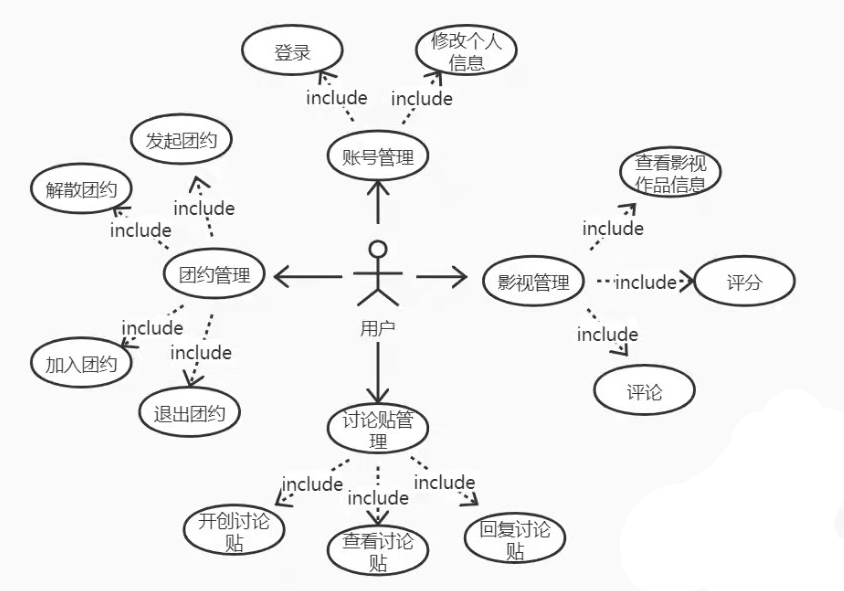


图3-2 一般用户用例图

在普通入口登陆，系统管理员与一般用户并无差异，拥有的权限都是对自己的个人信息进行修改操作，对影视资源以及讨论帖拥有查看以及评论操作，对团约有发起、加入、退出、查看等操作。系统的部分功能，使用用例表对其进行描述。

登录的用例描述如表3-1所示。

表3-1登录用例描述

|  |  |
| --- | --- |
| **用例名** | 登录 |
| **用例描述** | 系统管理员、一般用户利用该用例登录系统，通过身份认证后展示全部的操作界面 |
| **参与者** | 系统管理员、一般用户 |
| **涉众** | 系统管理员、一般用户：通过身份验证，展示相应的页面与操作 |

**续表3-1**

|  |  |
| --- | --- |
| **相关用例** | 无 |
| **前置条件** | 无 |
| **后置条件** | 如果登录成功，则展示相应的页面与操作 |
| **基本事件流**  （1）用例起始于用户进入系统首页  （2）系统拉起QQ、微信或支付宝的授权登录接口  （3）用户授权（D-1）  （4）系统验证用户名信息，同意用户登录系统（A-1），并返回登陆code  （5）系统展示全部的操作界面 | |
| **备选事件流**  A-\*用户在进入登录界面随时都可以终止用例  （1）系统提示关闭信息  （2）用户可以结束该用例或执行其他的操作  A-1 用户拒绝授权或者验证码超时  （1）系统提示用户已拒绝授权或者验证码超时，并进入第（2）步  （2）用户可以重新验证（B-1），也可以选择结束该用例 | |
| **补充约束-业务规则**  B-1 验证二维码每10s刷新一次  **补充约束-数据需求**  D-1用户要求：必须使用QQ、微信或支付宝已注册的账号  **补充约束-非功能需求**  安全性：使用QQ、微信或支付宝官方认证 | |

新增影视评论的用例描述如表3-2所示。

表3-2新增影视评论用例描述

|  |  |
| --- | --- |
| **用例名** | 新增影视评论 |
| **用例描述** | 一般用户可以利用该用例添加新的影视评论 |
| **参与者** | 一般用户 |
| **涉众** | 一般用户：会在相应影视评论区添加一条评论 |
| **包含的用例** | 查询影视资源，查询影视评论 |
| **前置条件** | 一般用户登录系统 |
| **后置条件** | 添加的评论信息正确的保存到系统中 |
| **基本事件流**  （1）用例起始于一般用户进入影视详细信息页面，准备添加评论  （2）包含“查询影视”来获取选中的影视资源详情列表（A-1）（D-1）  （3）显示选中影视的所有评论信息（A-3）（D-2）  （4）系统显示录入用户的评论信息表单页（B-1）（D-3）  （5）一般用户录入评论信息  （6）系统保存用户评论信息（A-2），用例结束 | |

**续表3-2**

|  |
| --- |
| **备选事件流**  A-\*一般用户在操作之前随时都可以终止用例  （1）系统提示关闭信息，刷新表单  （2）一般用户可以结束该用例或执行其他的操作  A-1 影视资源列表获取失败  （1）系统显示影视资源列表获取失败  （2）一般用户可以选择再次获取，也可以结束该用例  A-2 保存评论信息失败  （1）系统显示保存用户信息失败  （2）一般用户可以选择再次保存，也可以结束该用例  A-3 评论信息列表获取失败  （1）系统显示评论信息列表获取失败  （2）一般用户可以选择再次获取，也可以结束该用例 |
| **补充约束-业务规则**  B-1每次只允许添加一条评论数据  **补充约束-数据需求**  D-1显示的影视资源信息包括：影视资源ID、影视资源名称、影视资源封面、上映时间、时长、评分、导演、演员、影视资源类型、描述、上映地区、评论数量  D-2 显示的评论信息包括用户头像、用户名、具体内容、点赞数量  D-3 表单信息包括：评论具体内容 |

查询角色的用例描述如表3-3所示。

表3-3查询影视资源用例描述

|  |  |
| --- | --- |
| **用例名** | 查询影视资源 |
| **用例描述** | 其他用例可以包含该用例，以查询所需的影视资源列表信息 |
| **参与者** | 无 |
| **涉众** | 系统管理员、一般用户 |
| **基用例** | 查询影视资源，查询详细影视资源，查询影视评论 |
| **前置条件** | 无 |
| **后置条件** | 显示符合要求的查询结果 |
| **本事件流**  （1）用例起始于其他用例需要用过该用例来获取影视资源列表  （2）系统管理员、一般用户提交查询条件（D-1）  （3）系统根据设定的条件查询影视资源详细信息（A-1），并显示相关影视资源列表详细信息（D-2）、相关影视资源详细评论信息（D-3），该用例结束 | |
| **备选事件流**  A-\*系统管理员、一般用户在操作之前随时都可以终止用例  （1）系统提示关闭信息  （2）系统管理员、一般用户可以结束该用例或执行其他的操作  A-1 没有找到相关形式资源信息  （1）系统提示未找到相关影视资源信息  （2）系统管理员、一般用户可以再次进行查询，也可以结束用例 | |

**续表3-3**

|  |
| --- |
| **补充约束-数据需求**  D-1 查询条件包括：影视资源名称  D-2 影视资源信息包括：影视资源ID、影视资源名称、影视资源封面、上映时间、时长、评分、导演、演员、影视资源类型、描述、上映地区、评论数量  D-3 评论信息包括用户头像、用户名、具体内容、点赞数量 |

修改一般用户的用例描述如表3-4所示。

表3-4修改用户用例描述

|  |  |
| --- | --- |
| **用例名** | 修改用户 |
| **用例描述** | 一般用户通过该用例修改用户信息 |
| **参与者** | 一般用户 |
| **涉众** | 无 |
| **包含的用例** | 查询用户信息 |
| **前置条件** | 一般用户登录系统 |
| **后置条件** | 被修改用户的信息正确的保存到系统中 |
| **基本事件流**  （1）用例起始于一般用户进入用户管理页面，准备修改用户信息  （2）针对需要修改信息的内容，一般用户输入最新的用户信息  （3）一般用户提交修改后的用户信息  （4）系统将用户信息保存（A-1）  （7）处理完用户信息后（D-1），当前用例结束 | |
| **备选事件流**  A-\*一般用户在进入登录界面之前随时都可以终止用例  （1）系统提示关闭信息  （2）一般用户可以结束该用例或执行其他的操作  A-1 系统更新失败  （1）系统显示修改用户信息失败  （2）一般用户可以选择再次修改，也可以结束该用例 | |
| **补充约束-数据需求**  D-1 显示用户的信息包括：用户名、手机号、邮箱、QQ、微信、头像、性别 | |

上架影视资源的用例描述如表3-5所示。

表3-5上架影视资源用例描述

|  |  |
| --- | --- |
| **用例名** | 上架影视资源 |
| **用例描述** | 系统管理员修改系统中的角色信息 |
| **参与者** | 系统管理员 |
| **涉众** | 无 |
| **前置条件** | 系统管理员登陆 |
| **后置条件** | 影视资源查询 |

**续表3-5**

|  |
| --- |
| **基本事件流**  （1）用例起始于系统管理员登陆管理后台系统  （2）系统管理员进入影视资源管理界面  （3）包含“上架影视资源”来获取上架列表  （4）系统管理员输入影视资源的详细信息  （5）系统保存新上架的影视（A-1）  （6）返回所有影视资源列表（D-1）用例结束 |
| **备选事件流**  A-\*系统管理员在操作之前随时都可以终止用例  （1）系统提示关闭信息  （2）系统管理员可以结束该用例或执行其他的操作  A-1保存失败  （1）系统无响应  （2）系统管理员可以选择再次提交，也可以结束该用例 |
| **补充约束-数据需求**  D-1显示影视资源的信息包括：影视资源ID、影视资源名称、影视资源封面、上映时间、时长、评分、导演、演员、影视资源类型、描述、上映地区、评论数量  **补充约束-业务规则**  B-1每次只允许修改一个角色的信息 |

删除角色的用例描述如表3-6所示。

表3-6下架影视资源用例描述

|  |  |
| --- | --- |
| **用例名** | 下架影视资源 |
| **用例描述** | 系统管理员通过该用例下架已有的影视资源 |
| **参与者** | 系统管理员 |
| **涉众** | 无 |
| **包含的用例** | 查询影视资源，查询评论，删除评论 |
| **前置条件** | 系统管理员登录系统 |
| **后置条件** | 选中的影视资源被成功下架，相关评论删除 |
| **基本事件流**  （1）用例起始于系统管理员进入管理后台系统  （2）系统管理员进入影视资源管理界面（A-1）  （3）包含“查询影视”来获取可选的影视列表（A-2）（D-1）  （4）系统管理员选中一个影视资源，选择下架功能  （5）系统删除相关影视资源信息（A-3），用例结束。 | |

**续表3-6**

|  |
| --- |
| **备选事件流**  A-\*系统管理员在操作之前随时都可以终止用例  （1）系统提示关闭信息  （2）系统管理员可以结束该用例或执行其他的操作  A-1系统管理员无法进入角色管理界面  （1）重新获取请求，显示界面数据  （2）系统管理员也可以结束该用例  A-2 影视资源列表获取失败  （1）系统显示影视资源列表获取失败  （2）系统管理员可以选择再次获取，也可以结束该用例  A-3下架失败  （1）系统无响应  （2）系统管理员可以选择再次提交，也可以结束该用例 |
| **补充约束-数据需求**  D-1显示影视资源的信息包括：影视资源ID、影视资源名称、影视资源封面、上映时间、时长、评分、导演、演员、影视资源类型、描述、上映地区、评论数量 |

查询团约的用例描述如表3-7所示。

表3-7查询团约用例描述

|  |  |
| --- | --- |
| **用例名** | 查询团约 |
| **用例描述** | 系统管理员、一般用户查询团约 |
| **参与者** | 系统管理员、一般用户 |
| **涉众** | 一般用户 |
| **相关用例** | 无 |
| **前置条件** | 无 |
| **后置条件** | 页面上显示出团约数据 |
| **基本事件流**  （1）用例起始于系统管理员、一般用户进入系统  （2）系统管理员、一般用户进入团约界面（A-1）  （3）系统显示团约列表（D-1）  （4）系统管理员、一般用户查询条件（D-2）  （5）系统根据设定的条件查询相关团约信息（A-2），并显示详细信息，用例结束。 | |
| **备选事件流**   1. \*系统管理员、一般用户在操作之前随时都可以终止用例   （1）系统提示关闭信息  （2）系统管理员、一般用户可以结束该用例或执行其他的操作  A-1系统管理员、一般用户无法进入登录日志界面  （1）重新获取请求，显示界面数据  （2）系统管理员、一般用户也可以结束该用例  A-2没有找到相关团约信息  （1）系统提示读取中，显示空白列表  （2）系统管理员、一般用户重新输入查询条件，也可以结束该用例 | |

续表3-7

|  |
| --- |
| **补充约束-数据需求**  D-1显示的团约信息包括：发起人ID、成员ID、团约标题、内容描述、总人数、现有人数、  发起时间、结束时间  D-2查询条件包括：团约名称 |

加入团约的用例描述如表3-8所示。

表3-8加入团约用例描述

|  |  |
| --- | --- |
| **用例名** | 加入团约 |
| **用例描述** | 一般用户加入团约 |
| **参与者** | 一般用户 |
| **涉众** | 一般用户 |
| **相关用例** | 无 |
| **前置条件** | 一般用户登陆 |
| **后置条件** | 页面上显示出申请团约提示 |
| **基本事件流**  （1）用例起始于一般用户进入系统并授权登陆  （2）一般用户进入团约界面（A-1）  （3）系统显示团约列表（D-1）  （4）一般用户查看团约详细信息（D-2）  （5）系统根据设定的条件查询相关团约信息（A-2），并显示详细信息  （6）一般用户申请加入（A-3）  （7）系统提示消息发送，等待其他人同意，用例结束。 | |
| **备选事件流**   1. \*一般用户在操作之前随时都可以终止用例   （1）系统提示关闭信息  （2）一般用户可以结束该用例或执行其他的操作  A-1一般用户无法进入查询团约界面  （1）重新获取请求，显示界面数据  （2）一般用户也可以结束该用例  A-2没有找到相关团约信息  （1）系统提示读取中，显示空白列表  （2）一般用户重新输入查询条件，也可以结束该用例  A-3一般用户申请加入失败  （1）系统提示申请繁忙  （2）一般用户重新申请，也可以结束该用例 | |
| **补充约束-数据需求**  D-1显示的团约信息包括：发起人ID、成员ID、团约标题、内容描述、总人数、现有人数、  发起时间、结束时间  D-2查询条件包括：团约名称 | |

退出团约的用例描述如表3-9所示。

表3-9退出团约用例描述

|  |  |
| --- | --- |
| **用例名** | 退出团约 |
| **用例描述** | 一般用户退出团约 |
| **参与者** | 一般用户 |

**续表3-9**

|  |  |
| --- | --- |
| **涉众** | 一般用户 |
| **相关用例** | 无 |
| **前置条件** | 一般用户登陆，用户在当前团约中 |
| **后置条件** | 页面上显示出退出团约提示 |
| **基本事件流**  （1）用例起始于一般用户进入系统并授权登陆  （2）一般用户进入我的团约界面（A-1）  （3）系统显示我的团约列表（D-1）  （4）一般用户查看我的团约详细信息（A-2）（D-2）  （5）一般用户申请退出当前团约（A-3）  （6）系统提示消息发送，等待其他人同意，用例结束。 | |
| **备选事件流**   1. \*一般用户在操作之前随时都可以终止用例   （1）系统提示关闭信息  （2）一般用户可以结束该用例或执行其他的操作  A-1一般用户无法进入我的团约界面  （1）重新获取请求，显示界面数据  （2）一般用户也可以结束该用例  A-2没有找到相关团约信息  （1）系统提示读取中，显示空白列表  （2）一般用户重新点击进入，也可以结束该用例  A-3一般用户申请退出失败  （1）系统提示申请繁忙  （2）一般用户重新申请，也可以结束该用例 | |
| **补充约束-数据需求**  D-1显示的团约信息包括：团约标题、内容描述、总人数、现有人数  发起时间、结束时间  D-2显示的团约信息包括：发起人ID、成员ID、团约标题、内容描述、总人数、现有人数、  发起时间、结束时间 | |

## 3.4 性能需求

实现用户需求前不能缺少对所需性能的预算，为了保证软件能长久有效的运行，该系统需要满足以下几点要求：

1.对广大用户来说，软件的性能主要体现系统交互时相应的时间。我们应该采用懒加载的方式控制系统页面响应时间，要极力避免页面抖动。

2.系统数据库中数据的存储也是重中之重，如果服务器发生意外停机或数据丢失，对用户的使用体验是毁灭性的。为了避免这种情况，我们应该针对具体的业务，制定详细的数据库备份计划，保证系统的稳定与数据的安全。

3.随着系统影响力的扩大，使用的人数越多。会使数据执行的效率降低，我们应该在后端简化业务逻辑，前端控制资源文件加载的优先级，减少重排（Reflow），并且利用浏览器缓存将网络资源存储在本地，提高软件的效率。

## 3.5 本章小结

本章中，我们主要是围绕需求分析对系统的业务进行了大致的描述，并以此为根据阐述了系统设计的目标。为了使读者更容易阅读与使用，在用户需求描述这一小节，给出了各个用户角色的用例图和系统的用例图。使用用例表对特定的用例进行描述，使系统更具有逻辑性。为后期的总体设计起到一个铺垫的作用。

4 总体设计

在需求分析阶段我们对影视推荐系统的开发做了详细的分析工作，确立了整体架构与功能。本章主要进行系统的总体设计，包括系统结构设计、软件功能设计、数据库设计和页面设计。

## 4.1 系统结构设计

系统后台主体架构采用SpringBoot+Spring+MyBatis框架搭建。系统使用MyBatis作为数据访问层框架。mybatis是对jdbc的封装，它让数据库底层操作变的透明。MyBatis的操作都是围绕一个sqlSessionFactory实例展开的。MyBatis通过配置文件关联到各实体类的Mapper文件，Mapper文件中配置了每个类对数据库所需进行的sql语句映射。在每次与数据库交互时，通过sqlSessionFactory拿到一个sqlSession，再执行sql命令。使用Spring的控制反转管理整个项目中各个类之间的依赖关系，从而为各层之间的代码进行解耦合，使用Spring面向切面编程的思想对项目的日志进行处理。在Spring中整合开源数据库连接池HikariCP进行数据库连接，并进行事务的相关配置，通过切面编程使用自定义注解实现对事务方法的调用。用户提交的请求通过Controller接收，调用Service层、Dao层最终把数据持久化到数据库中。系统架构图，如图4-1所示。

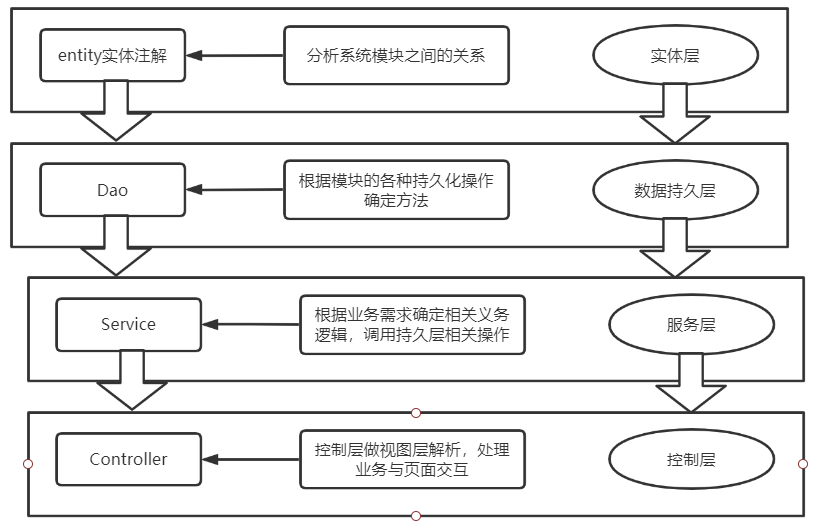


图4-1 系统架构图

系统前端主要使用的是React+Ant Design+TypeScript+Umi等多个前端技术搭建，React主要用于构建高性能及响应式用户界面。能够使用虚拟DOM并在发生变更时利用补丁安装机制只对DOM中的dirty部分进行重新渲染。用户基于页面进行的所有向服务器发送的请求，都是Axios异步请求的模式，不仅可以提高整个系统的通用性，而且也能大大降低服务器端的压力。

前台通过Axios进行数据访问，后台以Json的数据格式进行响应，数据交互示意图，如图4-2所示。

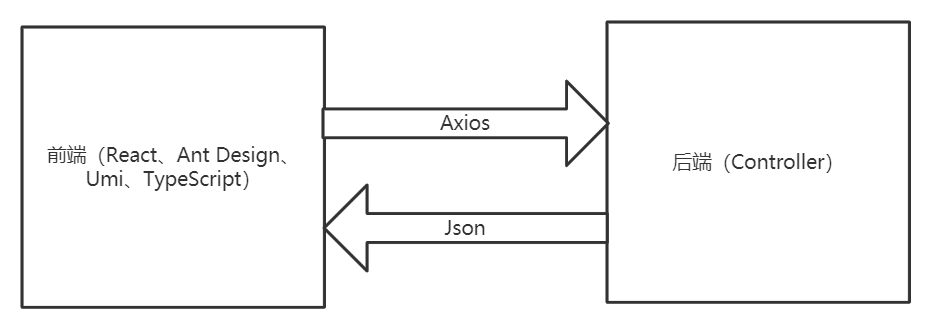


图4-2 数据交互示意图

## 4.2 软件功能设计

本系统主要涉及到系统管理员与一般用户这两种角色。系统管理员有用户账号管理，影视资源管理，影视评论管理，讨论帖管理，团约管理，对于一般用户有个人信息管理，影视资源管理，影视评论管理，讨论帖管理，团约管理，如图4-3所示。

接下来我们分别阐述系统中各个角色中每个管理模块的具体功能。

管理员可以对违规用户账号进行封禁操作，因为本平台采用QQ、微信、支付宝授权等方式登陆，所以管理员并不拥有创建用户的权力。管理员也可以查看所有用户的详细信息，但不能对其进行修改。在电影管理中，管理员可以查看所有影视资源的详细信息，并拥有对影视资源进行上架、下架、修改的权限。在评论管理模块中，管理员可以看到所有用户的发言，并拥有对其删除的权限。在讨论帖管理模块中，管理员可以看到所有讨论帖，以及每个讨论帖的发起人的详细信息，每个参与的人发表的言论，并对其拥有删除的权限。在团约管理中，管理员能够查看到所有已发起的团约和已结束的团约，以及每个团约的发起人和参与用户的详细信息，并对每个团员拥有删除的权限。

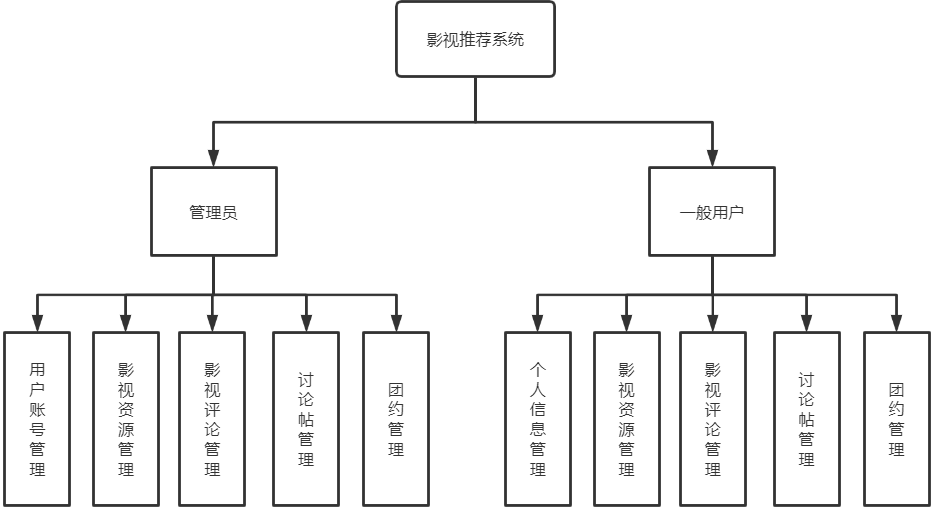


图4-3 软件功能结构图

一般用户在此系统可以查看个人信息，多数为授权平台上存储的信息，如头像、授权平台的登陆账号id等，一般用户也可以浏览所有的影视资源信息以及相关影视资源下的评论，也可以发表自己的评论。可以查看所有的讨论帖内容，并查看所有发表言论的用户的部分信息，可以查看所有已发起的团约以及查看团约内的用户，在授权登陆后可以对相关影视资源进行评论、在讨论帖中回复与别人交流以及参加或发起团约。

## 4.3 数据库设计

数据库设计在整个软件开发的起到的举足轻重的作用，数据库与需求是相辅相成的，我经常把软件开发比作汽车制造。汽车制造会经过图纸设计，模型制作，样车制造，小批量试生产，最后是批量生产等步骤。整个过程环环相扣，后一过程是建立在前一过程正确的前提基础之上的。如果在图纸设计阶段发现了一个纰漏，我们可以重新进行图纸设计，如果到了样车制造阶段发现这个错误，那么我们就要把从图纸设计到样车制造的阶段重来，越到后面发现设计上的问题，所付出的代价越大，修改的难度也越大。通常开发人员在设计数据库时有两个重要方面：分析数据库、设计和实现数据库。

### 4.3.1 数据库概念结构设计

在数据库的开发之前，我们需要思考使用系统的角色都具有什么属性（即概念模型），只有明确了用户的真正需求，我们才能真正的设计推导出概念模型。

概念模型的主要特点是：

1.对现实世界的一个真实模型，能真实、充分地反映现实世界中事物和事物之间的联系，以此来满足用户对数据的处理要求。

2.易于理解，哪怕是不熟悉计算机的用户，也能简单的理解并交换意见，沟通十分方便。数据库设计成功的关键与否很大程度上取决于用户是否积极参与。

3.易于更改，当应用的场景和用户的需求发生改变的时候，概念模型能够更加容易的修改或扩充。

4.易于向关系、网状、层次等各种数据模型转换。

相比于数据模型，概念模型更加独立、稳定、和抽象。我们将使用E-R图来

描述概念模型。

根据影视推荐系统需求分析，可以确定拟开发的此系统平台被用户分为两大类：系统管理员、一般用户。他们所涉及的模块也不尽相同。通过分析我们确定拟开发的系统主要包括以下实体：用户实体、影视资源实体、评论实体、讨论帖实体、讨论帖评论详情志实体、团约实体、团约个人实体。各实体具体介绍如下：

1.用户实体

用户实体中，属性包括用户ID、用户名、用户头像、性别、账号是否可用、电话、QQ、微信、邮箱。用户实体属性图，如图4-4所示。

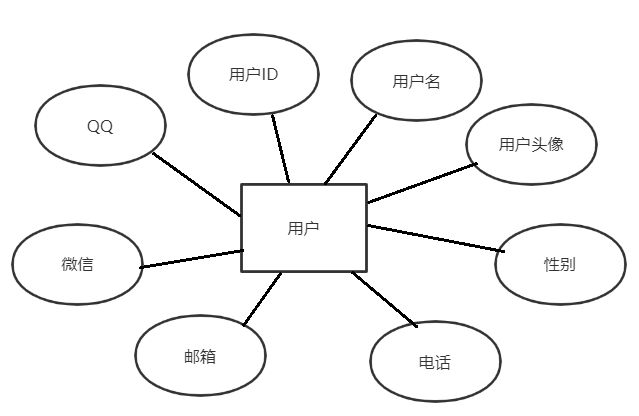


图4-4 用户实体属性图

2.影视资源实体

影视资源实体中，属性包括影视ID、影视名称、影视封面、上映时间、时长、评分、导演、演员、影视类型、详细描述、上映地区、评论数量、评分人数。影视资源实体属性图，如图4-5所示。

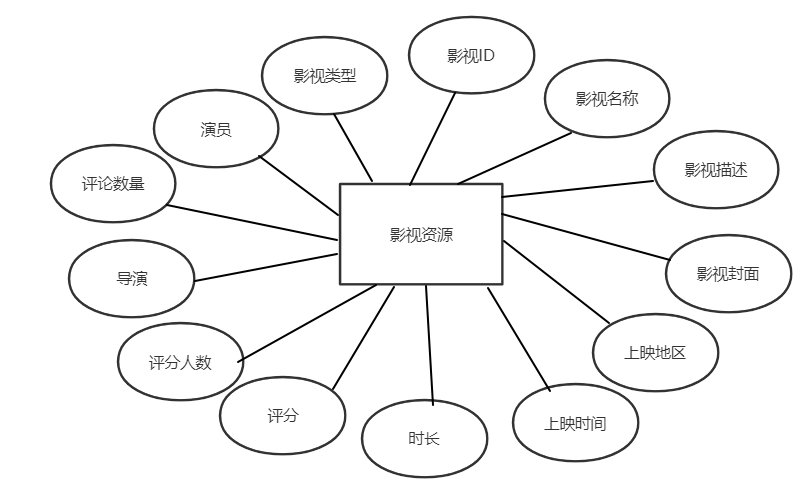


图4-5 影视资源实体属性图

3.评论实体

评论实体中，属性包括评论ID、评论用户ID、影视资源ID、用户头像、评论内容、评分、评论时间、点赞。评论实体图，如图4-6所示。

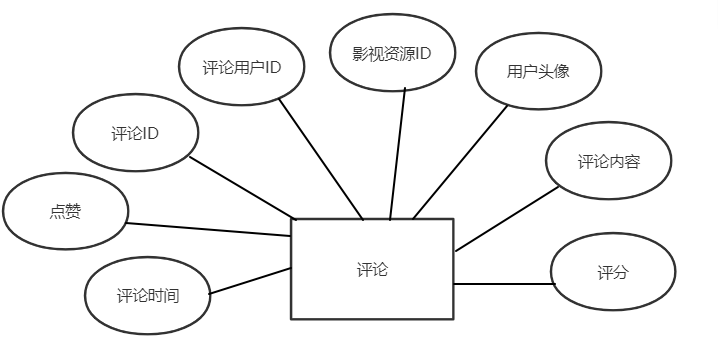


图4-6 评论实体属性图

4.团约实体

团约实体中，属性包括团约ID、团约用户ID、团约标题、团约内容、发起时间、结束时间、总人数、现有人数。团约实体属性图，如图4-7所示。

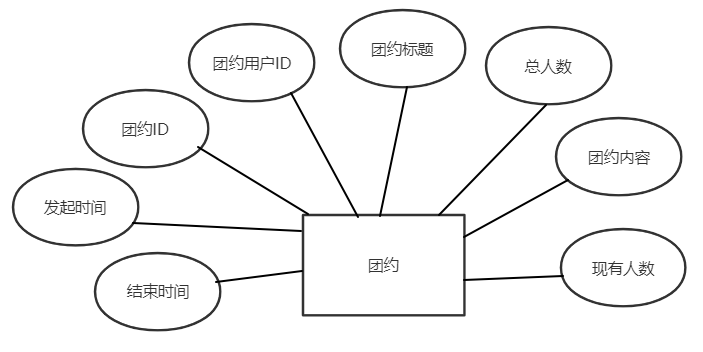


图4-7 部门实体属性图

5.讨论帖实体

讨论帖实体中，属性包括讨论帖ID、发起人ID、讨论帖标题、内容描述、讨论信息条数、发起时间、发起人头像、发起人名称。讨论帖实体属性图，如图4-8所示。

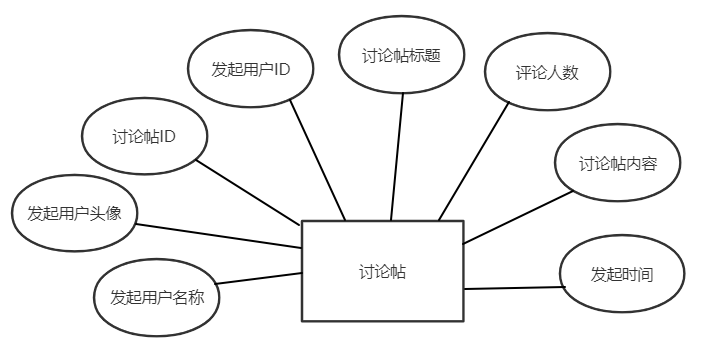


图4-8 讨论帖实体属性图

6.讨论帖评论详情实体

讨论帖评论详情实体，属性包括讨论帖评论详情ID、用户ID、讨论帖ID、内容、发布时间、用户头像、用户名。讨论帖评论详情实体属性图，如图4-9所示。

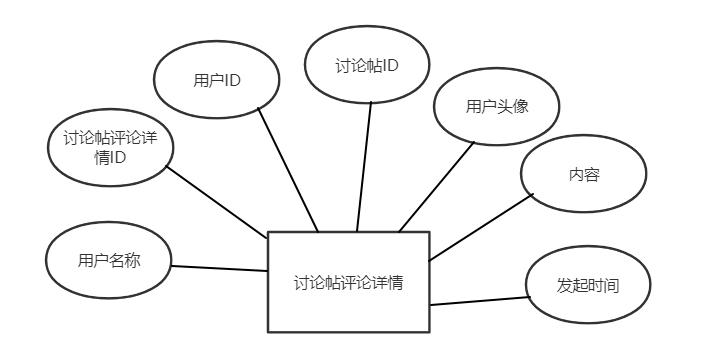


图4-9 讨论帖评论详情实体属性图

7.团约个人实体

团约个人实体，属性包括团约ID、用户ID、用户名、用户头像、性别、QQ、微信、电话、加入时间、加入状态。团约个人实体属性图，如图4-10所示。

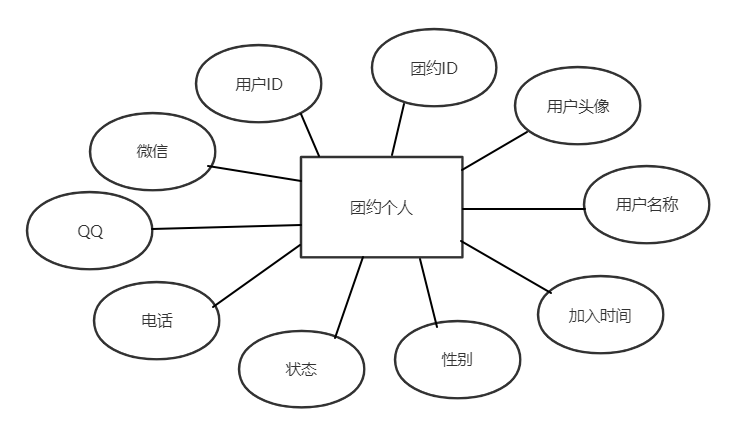


图4-10 团约个人实体属性图

根据系统的功能，确定了各个实体间的关系，画出系统E-R图，如图4-14所示。

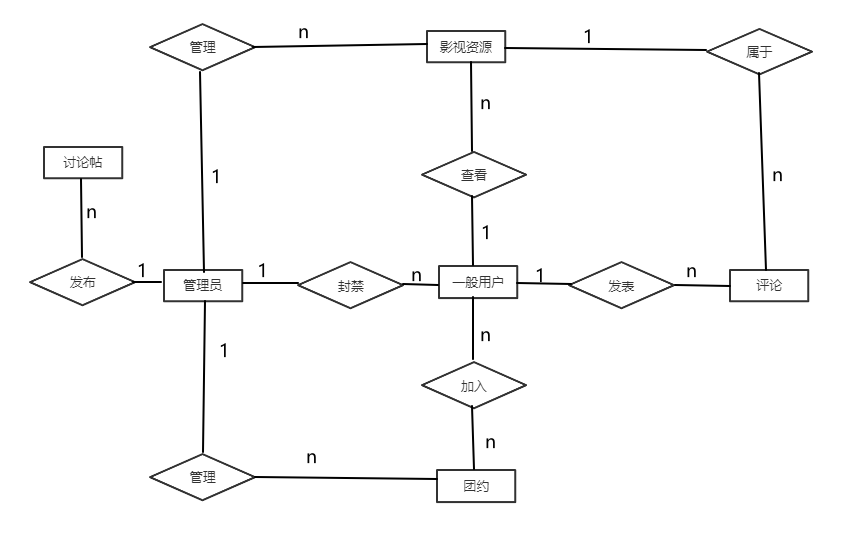


图4-14 系统E-R图

### 4.3.2 数据库逻辑结构设计

概念结构是独立于任何一种数据模型的信息结构，逻辑结构设计就是把概念结构设计阶段设计好的基本实体-关系图转换为与选用的数据库管理系统产品所支持的数据模型相符合的逻辑结构。

在概念结构设计阶段我们已经设计出了本系统数据库的实体-关系图，下面就是把实体-关系图转换为逻辑结构设计中对应的的关系模式（主键用下划线着重标出）。

1.用户信息（用户ID、用户名、用户头像、性别、账号是否可用、电话、QQ、微信、邮箱）

此为用户信息关系模式，用户ID作为关系主键。

2.影视资源信息（影视ID、影视名称、影视封面、上映时间、时长、、评分、导演、演员、影视类型、详细描述、上映地区、评论数量、评分人数）

此为影视资源关系模式，角色ID作为关系主键。

3.评论信息（评论ID、评论用户ID、影视资源ID、用户头像、评论内容、评分、评论时间、点赞、创建时间）

此为评论关系模式，评论ID作为关系主键。

4.团约信息（团约ID、团约用户ID、团约标题、团约内容、发起时间、结束时间、总人数、现有人数）

此为团约关系模式，团约ID作为关系主键。

5.讨论帖信息（讨论帖ID、发起人ID、讨论帖标题、内容描述、讨论信息条数、发起时间、发起人头像、发起人名称）

此为讨论帖关系模式，讨论帖ID作为关系主键。

6.讨论帖评论详情信息（讨论帖评论详情ID、用户ID、讨论帖ID、内容、发布时间、用户头像、用户名）

此为讨论帖评论详情关系模式，讨论帖评论详情ID作为关系主键。

7.团约个人信息（团约ID、用户ID、用户名、用户头像、性别、QQ、微信、电话、加入时间、加入状态）

此为登团约个人信息关系模式，团约ID和用户ID作为关系联合主键。

### 4.3.3 数据库表的结构设计

本系统在实现的过程中主要创建的表有：user（用户）、moves（影视资源）、comments（评论）、spellGroups（团约）、discussion（讨论帖）、discussionItem（讨论帖评论）、spellGroupsUser（团约个人）。

1. user（用户）信息表，如表4-1所示。

表4-1 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据含义 | 数据类型 | 数据长度 | 是否允许空值 |
| userId | 用户ID | bigint |  | NO |
| username | 用户名 | varchar | 50 | NO |
| userAvatar | 头像 | varchar | 128 | NO |
| gender | 性别 | varchar | 20 | NO |
| email | 邮箱 | varchar | 128 | YES |
| mobile | 联系电话 | varchar | 20 | NO |
| QQ | QQ号码 | varchar | 20 | NO |
| wechat | 微信 | varchar | 20 | NO |

2. moves（影视资源）信息表，如表4-2所示。

表4-2 影视资源信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据含义 | 数据类型 | 数据长度 | 是否允许空值 |
| moveId | 电影ID | bigint |  | NO |

续表4-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| moveName | 影视名称 | varchar | 100 | NO |
| moveAvatar | 影视封面 | varchar | 100 | NO |
| startData | 上映时间 | varchar | 100 | NO |
| time | 时长 | varchar | 100 | NO |
| score | 评分 | int | 20 | YES |
| director | 导演 | varchar | 100 | NO |
| actors | 演员 | varchar | 100 | NO |
| type | 类型 | varchar | 100 | NO |
| describe | 影视描述 | varchar | 100 | NO |
| place | 上映地区 | varchar | 100 | NO |
| counts | 评论数量 | bigint |  | YES |
| scoreNum | 评分人数 | bigint |  | YES |

3. comments（评论）信息表，如表4-3所示。

表4-3 评论信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据含义 | 数据类型 | 数据长度 | 是否允许空值 |
| commentsId | 评论ID | bigint |  | NO |
| userId | 用户ID | bigint |  | NO |
| moveId | 影视ID | bigint |  | NO |
| uesrAvatar | 用户头像 | varchar | 100 | NO |
| textValue | 评论内容 | varchar | 100 | NO |
| score | 评分 | int |  | YES |
| stars | 点赞 | bigint |  | YES |
| commentsTime | 评论时间 | bigint |  | NO |
| username | 用户名 | varchar | 100 | NO |

4.spellGroups（团约）信息表，如表4-4所示。

表4-4 团约信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据含义 | 数据类型 | 数据长度 | 是否允许空值 |
| spellGroupId | 团约ID | bigint |  | NO |
| userId | 用户ID | bigint |  | NO |

续表4-4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| title | 团约名称 | varchar | 100 | NO |
| content | 内容描述 | varchar | 100 | NO |
| count | 总人数 | int |  | NO |
| realCount | 现有人数 | int |  | NO |
| startTime | 发起时间 | bigint |  | NO |
| endTime | 结束时间 | bigint |  | NO |

5. discussion（讨论帖）信息表，如表4-5所示。

表4-5 讨论帖信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据含义 | 数据类型 | 数据长度 | 是否允许空值 |
| discussionId | 讨论帖ID | bigint |  | NO |
| mainUserId | 发起人ID | bigint |  | NO |
| title | 讨论帖标题 | varchar | 100 | NO |
| content | 内容描述 | varchar | 100 | NO |
| count | 信息条数 | int |  | YES |
| startTime | 发起时间 | bigint |  | NO |
| uesrAvatar | 发起人头像 | varchar | 100 | NO |
| username | 发起人名称 | varchar | 100 | NO |

6. discussionItem（讨论帖评论）信息表，如表4-6所示。

表4-6 讨论帖评论信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据含义 | 数据类型 | 数据长度 | 是否允许空值 |
| itemId | 讨论帖评论ID | bigint |  | NO |
| userId | 用户ID | bigint |  | NO |
| discussionId | 讨论帖ID | bigint |  | NO |
| content | 内容描述 | varchar | 100 | NO |
| time | 发布时间 | bigint |  | NO |
| uesrAvatar | 发起人头像 | varchar | 100 | NO |
| username | 发起人名称 | varchar | 100 | NO |

7.spellGroupsUser（团约个人）信息表，如表4-7所示。

表4-7 团约个人信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据含义 | 数据类型 | 数据长度 | 是否允许空值 |
| spellGroupId | 团约ID | bigint |  | NO |
| userId | 用户ID | bigint |  | NO |
| username | 用户名 | varchar | 50 | NO |
| userAvatar | 头像 | varchar | 128 | NO |
| gender | 性别 | varchar | 20 | NO |
| email | 邮箱 | varchar | 128 | YES |
| mobile | 联系电话 | varchar | 20 | NO |
| QQ | QQ号码 | varchar | 20 | NO |
| wechat | 微信 | varchar | 20 | NO |
| joinTime | 加入时间 | bigint |  | NO |
| isInGroup | 状态 | varchar | 20 | NO |

## 4.4 本章小结

本章进行了总体设计，首先，我们从系统的结构、数据的交互方式、实现原理入手，使用系统架构图、服务器端页面渲染图、前后端的数据交互图，展示了系统架构的实现原理，同时也介绍了数据在系统内的交互过程。在对软件功能的设计过程中，使用软件功能结构图对系统模块的功能进行了划分和展示，并对每个功能做了功能性的描述。接着开始进行数据库设计。通过设计物理结构模型将数据表结构存入关系数据库中，保证数据独立性。

5 详细设计与实现

本章主要介绍系统功能设计、方法设计和软件具体实现的过程。通过对软件功能进行详细的逻辑描述，使用活动图的方式展示具体的流程。系统的实现效果在下文将使用系统截图展示。

## 5.1 功能设计

### 5.1.1 用户登录模块的设计

用户登录是整个应用开发平台系统的入口，任何用户需要进入本系统，必须先在首页点击登录，前端会跳转到相应的授权平台，本系统暂时只支持QQ、微信、支付宝等平台的认证，用户在授权平台选择是否授权，拒绝授权的用户将会返回系统首页，选择授权的用户，相应的授权平台将会相应的账号检测，如果账号有违规操作或者其他不合法行为被封禁，将无法成功授权，成功授权之后生成唯一code供给后台判断，每个code仅可使用一次，使用一次后会过期以保证系统的安全，如果已被使用过，系统将会提示‘code被使用，请不要重复使用’code通过后台检测后，将相关账号的数据通过第三方授权平台发送到服务器，服务器将相关数据存储起来，包括用户的账号、头像等部分公开信息，前台给出登陆成功的响应并给出相关的提示信息。用户登录的活动图，如图5-1所示。

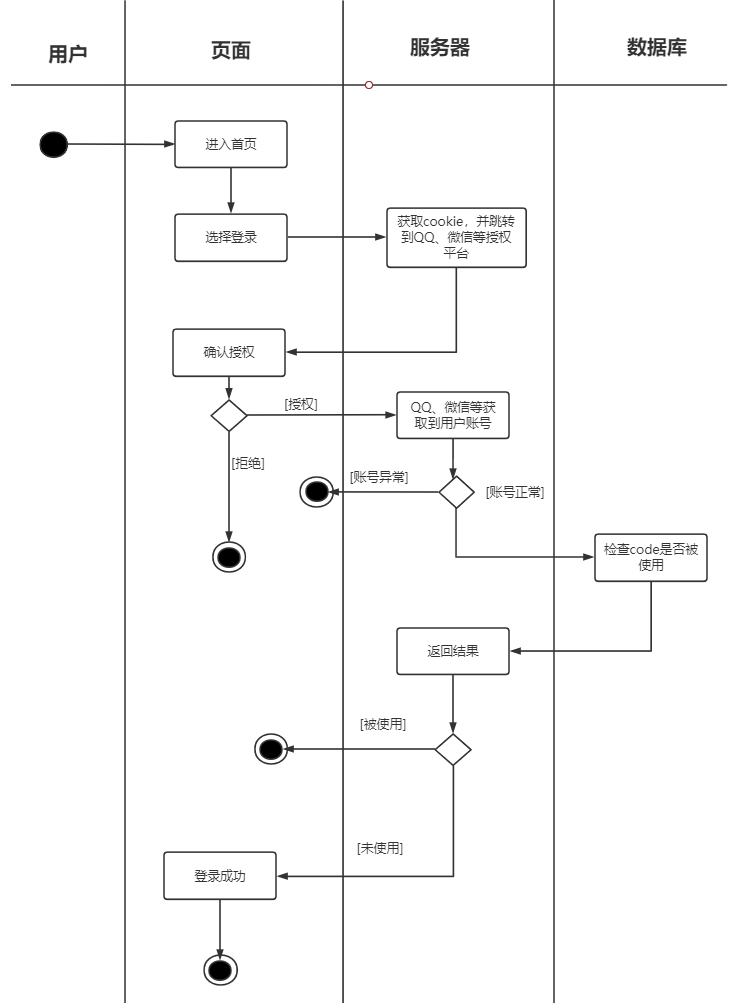


图5-1用户登录活动图

### 5.1.2 影视资源管理模块的设计

影视资源管理模块是为了系统管理员为了管理各类影视资源设计的。系统管理员在成功授权登陆之后，点击影视资源管理菜单，系统将会检索所有的影视资源将不同类的影视资源分类显示，如电影、电视剧、综艺等不同类型的影视资源。显示的数据为每个影视资源的的部分信息，包括影视资源封面、影视资源名称、影视资源上映时间、影视资源评分。系统管理员可以在此页面上架、下架、更新、条件查询系统内的所有影视资源。

当进行查询操作的时候，系统管理员需要在搜索框中搜索目标影视资源，点击标题或者查看详情进入当前目标影视资源的详细信息界面，包括影视ID、影视名称、影视封面、上映时间、时长、评分、导演、演员、影视类型、详细描述、上映地区、评论数量、评分人数。

影视资源管理模块（查询）的活动图，如图5-3所示。

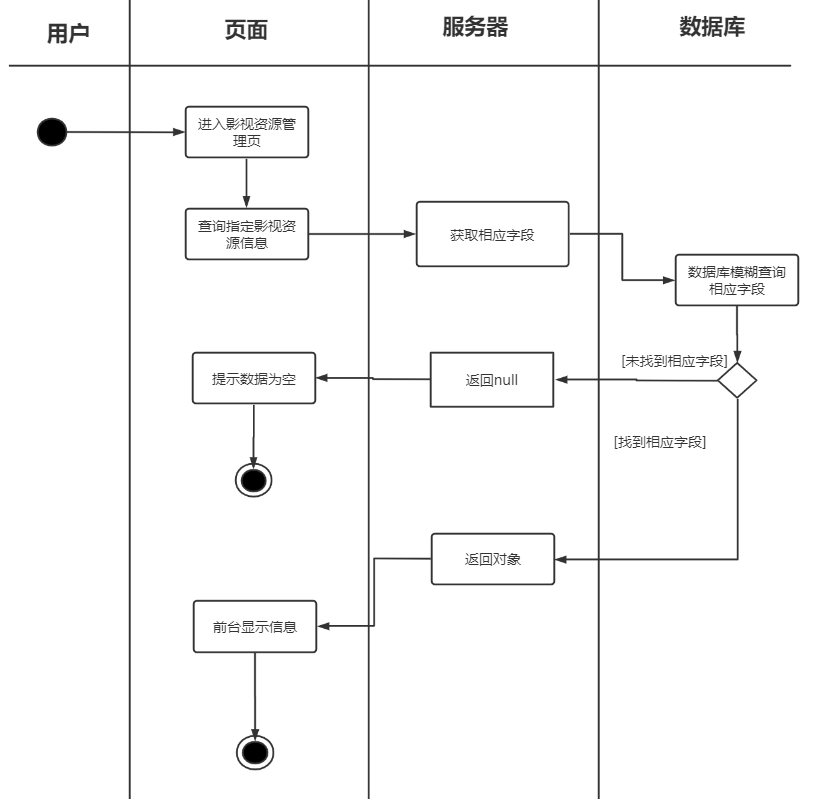


图5-3 影视资源管理活动图（查询）

当进行更新操作的时候，系统管理员需要在此页面或者在查看电影详情页面点击更新按钮，系统会弹出一个Drawer弹窗，上面会显示电影的部分信息，如影视名称、影视封面、上映时间、时长、导演、演员、影视类型、详细描述。系统管理员可以对这些属性进行修改，但是影视ID、评分、评论数量、评分人数这些属性管理员无法修改。

影视资源管理模块（更新）的活动图，如图5-4所示。

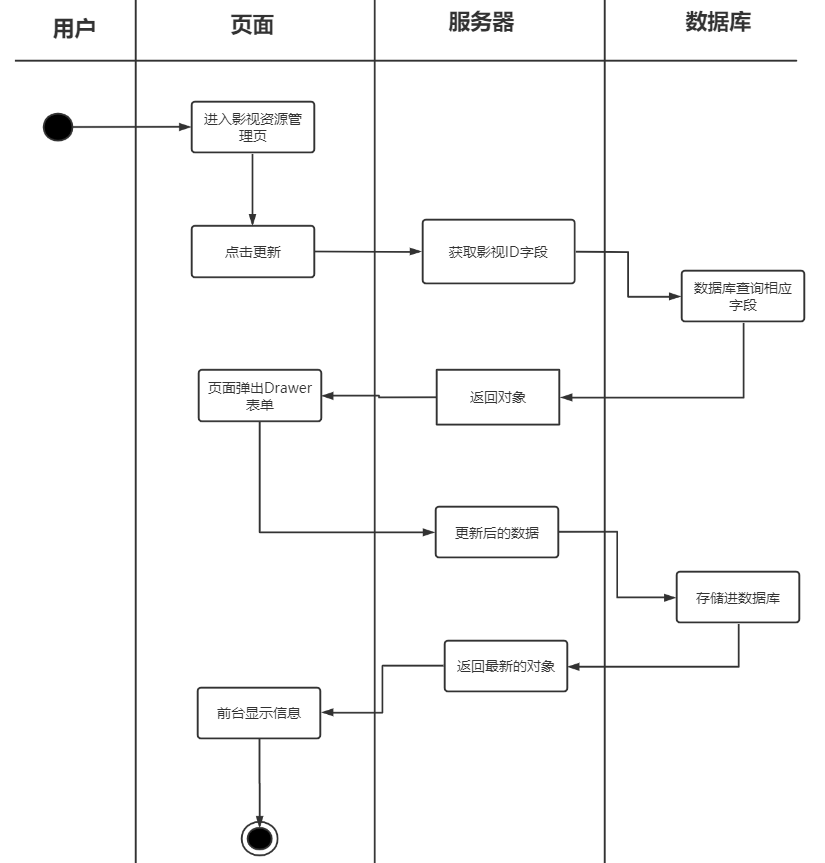


图5-4 影视资源管理活动图（更新）

当进行上架操作的时候，系统管理员需要在此页面点击上架按钮，系统会弹出一个Drawer弹窗，上面含有上架影视资源所需要的所有信息，如影视名称、影视封面、上映时间、时长、导演、演员、影视类型、详细描述。这类信息需要管理员自行添加，而影视ID、评分、评论数量、评分人数这些属性需要用户来根据自己的感受添加，管理员无法改变。

影视资源管理模块（上架）的活动图，如图5-5所示。

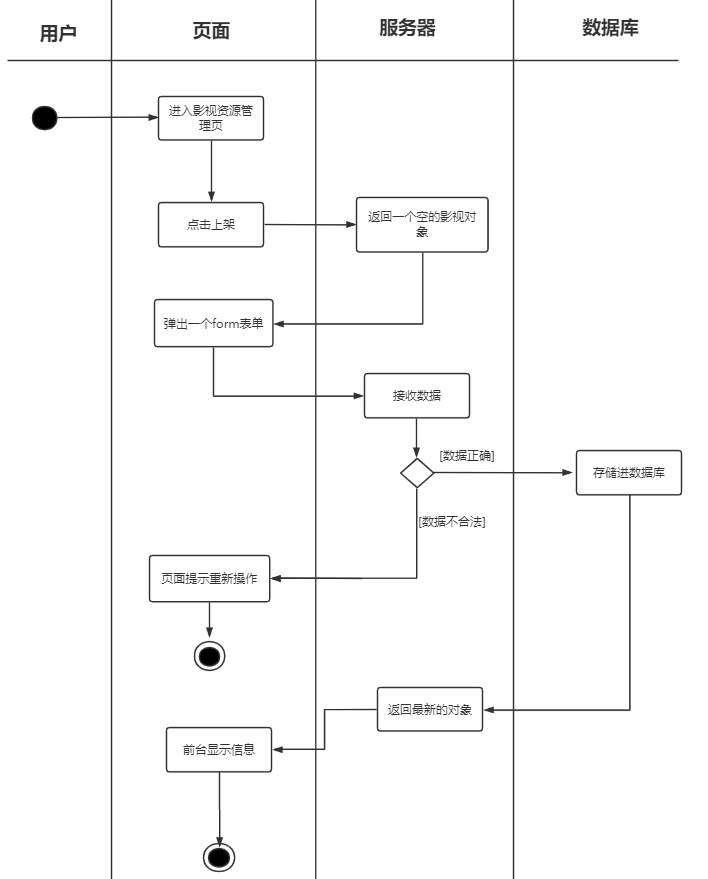


图5-5 影视资源管理活动图（上架）

当进行下架操作的时候，系统管理员需要在此页面或者在查看电影详情页面点击下架按钮，系统会弹出一个气泡选择框，如果管理员选择确认，系统成功将相关资源下架，并删除数据库的相关信息，包括此影视资源的评论信息，也将一并删除。

影视资源管理模块（下架）的活动图，如图5-6所示。

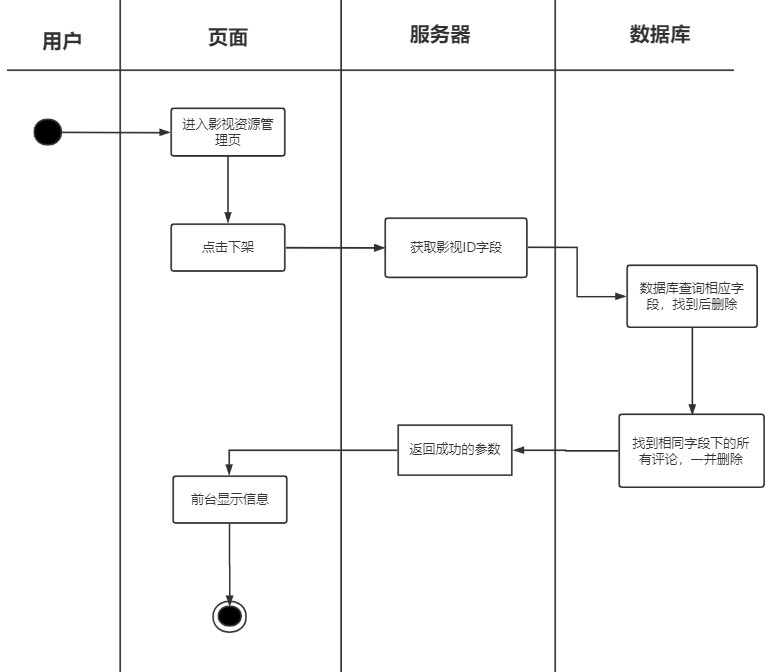


图5-6 影视资源管理活动图（下架）

### 5.1.3 讨论管理模块的设计

讨论管理模块为了所有角色设计的。一般用户在成功授权登陆之后，点击查看影视资源详细信息，系统将会检索选中影视资源的所有信息，包括选中影视资源的所有评论信息，评论信息包括以发表评论的用户的头像、名称、发表时间、评论内容、点赞数量等部分信息，一般用户也可以在页面添加自己的评论，页面会及时的刷新。管理员在成功授权登陆之后，在管理讨论界面可以查看所有影视资源的评论信息，并可以在此页面进行删除评论操作。

当进行删除评论操作的时候，管理员需要在管理评论页面点击删除评论按钮，系统会弹出一个气泡选择框，如果管理员选择确认，系统成功将相关评论删除，并删除数据库的相关信息。

讨论管理模块（删除）的活动图，如图5-7所示。

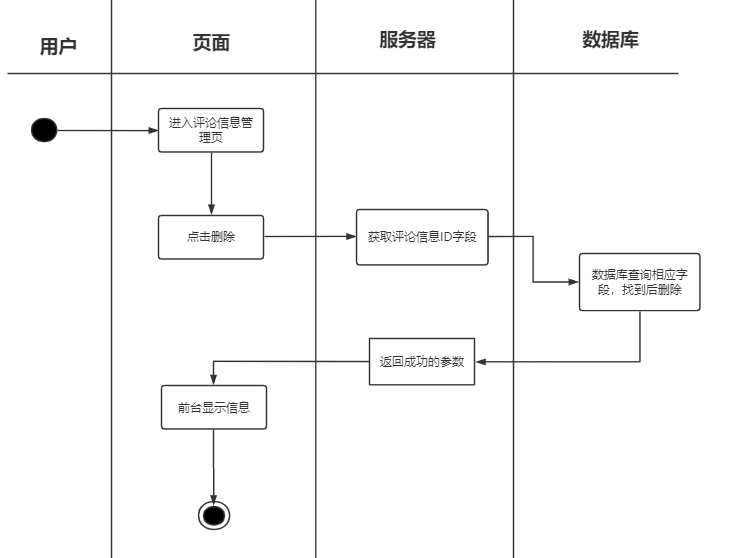


图5-7 讨论管理活动图（删除）

当进行发表评论操作的时候，一般用户需要在进入某个影视资源的详细界面，在讨论区的模块中点击“我要评论”，系统弹出增加评论的Drawer，用户填写完自己的评论后，点击发表，信息会以评论对象（Object）的形式提交到服务器，系统会自动补充用户的部分信息，如评论ID、评论用户ID、影视资源ID、用户头像、评论时间，并及时刷新页面进行展示。

讨论管理模块（发表）的活动图，如图5-8所示。

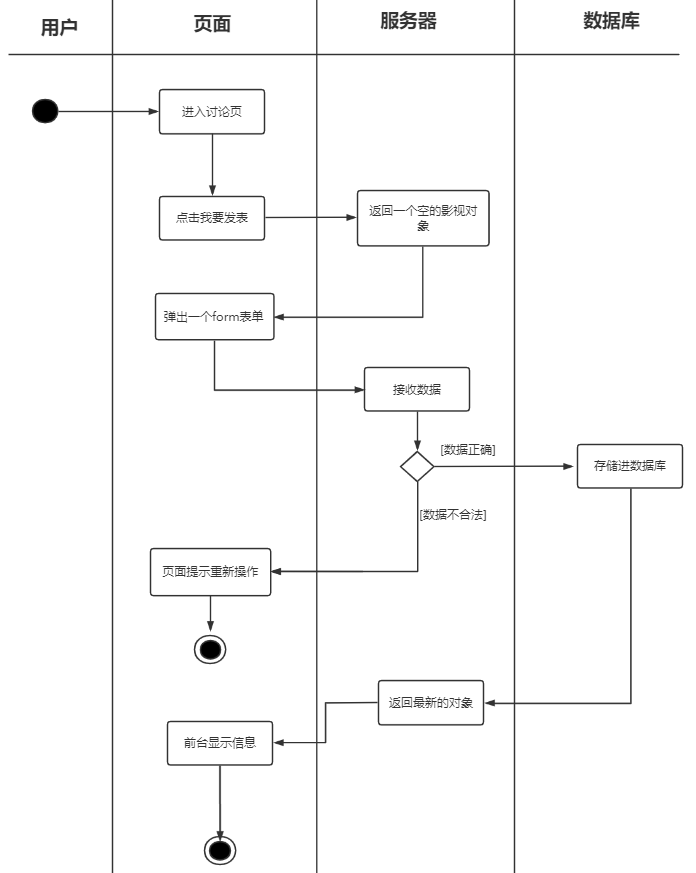


图5-8 讨论管理活动图（发表）

### 5.1.4 讨论帖管理模块的设计

讨论帖管理模块管理模块为了所有角色设计的。一般用户在成功授权登陆之后，点击查看所有讨论帖，系统将会检索讨论帖资源的所有信息，在此页面通过卡片（Card）的方式将每个讨论的展示出来，但只展示部分信息，如讨论帖标题，发起人，发起时间，回复人数，一般用户可以点击讨论帖标题或者点击查看详情按钮进入讨论帖页面，展示出该讨论帖的所有信息，如讨论帖标题、内容描述、讨论信息条数、发起时间、发起人头像、发起人名称、讨论帖内别人的回复内容、回复人的头像、回复人的名称,回复人回复的时间，一般用户可以在页面添加自己的回复，页面会及时的刷新。管理员在成功授权登陆之后，在管理讨论帖界面可以查看所有讨论帖的详细信息，并可以在此页面进行删除评论操作。

当进行查询操作的时候，一般用户可以搜索框中搜索讨论帖标题，该字段会通过服务器发送给数据库，在数据库进行模糊查询，将相似的标题全部返回到前端页面，点击标题或者查看详情进入当前目标讨论帖的详细信息界面，包括讨论帖标题、内容描述、讨论信息条数、发起时间、发起人头像、发起人名称。

讨论帖管理模块活动图（查询），如图5-9所示。

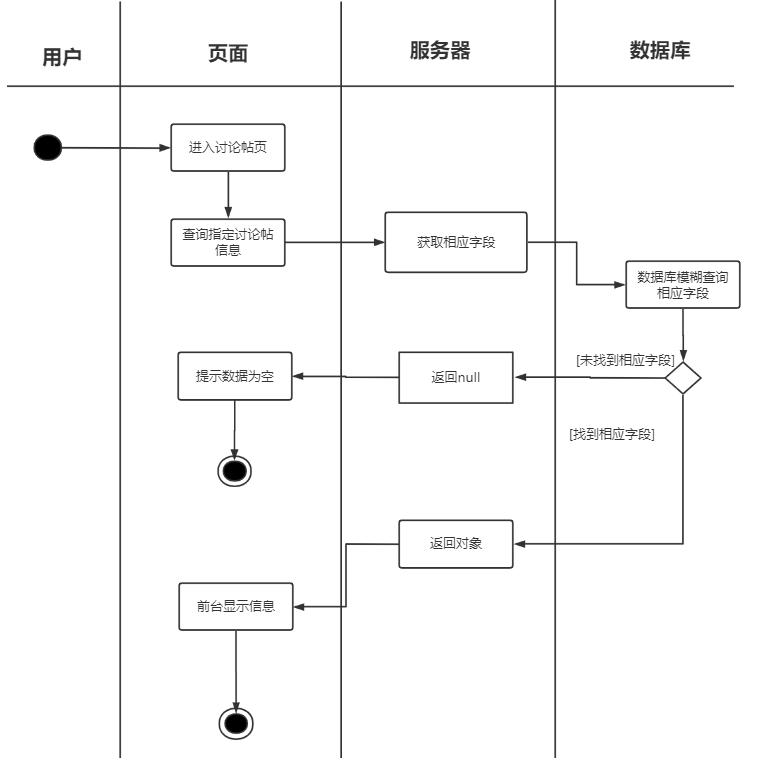


图5-9 讨论帖管理活动图（查询）

当进行回复操作的时候，一般用户需要在进入某个讨论帖的详细信息界面，在讨论区的模块中点击“我要评论”，系统弹出回复的Drawer，用户填写完自己的回复后，点击发表，信息会以对象（Object）的形式提交到服务器，系统会自动补充用户的部分信息，如讨论帖评论详情ID、用户ID、讨论帖ID、内容、发布时间、用户头像、用户名，并及时刷新页面进行展示。

讨论帖管理模块活动图（回复），如图5-10所示。

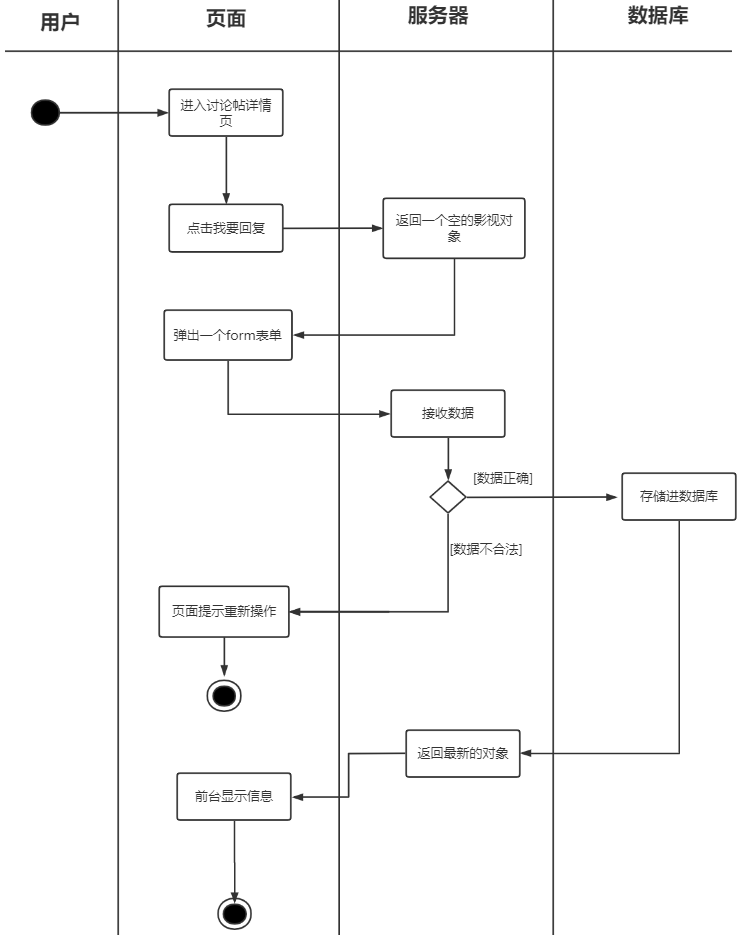


图5-10 讨论帖管理活动图（回复）

当进行删除讨论帖操作的时候，管理员需要在管理讨论帖页面点击删除讨论帖按钮，系统会弹出一个气泡选择框，如果管理员选择确认，系统成功将相关讨论帖删除，并删除数据库的相关信息，包括此讨论帖内的回复信息，也将一并删除。

讨论帖管理模块活动图（删除），如图5-11所示。

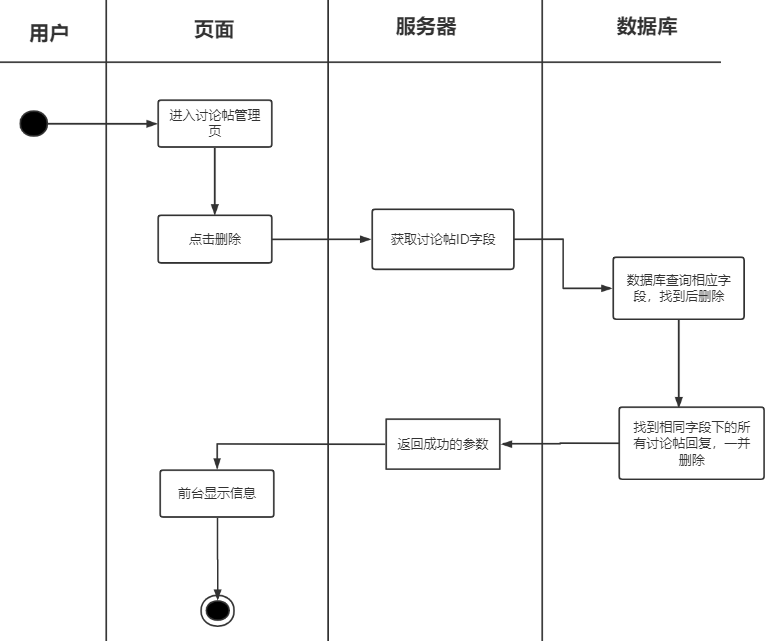


图5-11 讨论帖管理活动图（删除）

### 5.1.5 团约管理模块的设计

团约管理模块是为了所有角色设计的。一般用户在成功授权登陆之后，点击查看所有团约，系统将会检索团约资源的所有信息，在此页面通过卡片（Card）的方式将每个团约信息的展示出来，但只展示部分信息，如团约标题、团约内容、发起时间、结束时间、总人数、现有人数。一般用户可以点击团约标题或者点击查看详情按钮进入团约页面，展示出该团约的所有信息，如团约标题、内容描述、发起时间、发起人头像、发起人名称、团约内其他用户的用户名、其他用户的用户头像、性别、QQ、微信、电话、加入时间、加入状态，一般用户可以在此页面发起申请，其他人的页面会及时的刷新并显示未读消息。管理员在成功授权登陆之后，在管理团约界面可以查看所有团约的详细信息，并可以在此页面进行删除团约操作。

当进行查询操作的时候，一般用户可以搜索框中搜索团约标题，该字段会通过服务器发送给数据库，在数据库进行模糊查询，将相似的标题全部返回到前端页面，点击标题或者查看详情进入当前目标团约的详细信息界面，包括团约标题、团约内容、发起时间、结束时间、总人数、现有人数。

团约管理模块的活动图（查询），如图5-12所示。

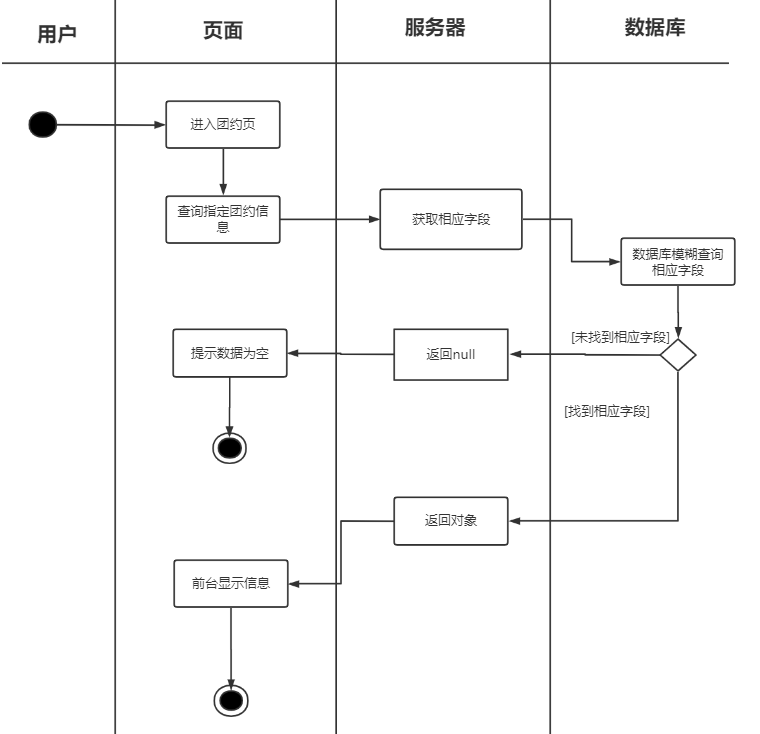


图5-12 团约管理活动图（查询）

当进行申请加入操作的时候，一般用户需要在进入某个团约的详细信息界面，在弹出的Drawer脚部模块中点击“我要申请”，系统弹出申请的Drawer，用户填写完自己的个人信息后，点击申请，信息会以对象（Object）的形式提交到服务器，系统会自动补充用户的部分信息，如团约ID、用户ID、用户名、用户头像、性别、QQ、微信、电话、加入时间、加入状态，此团约其他用户将会收到系统消息的提醒。

团约管理模块的活动图（申请），如图5-13所示

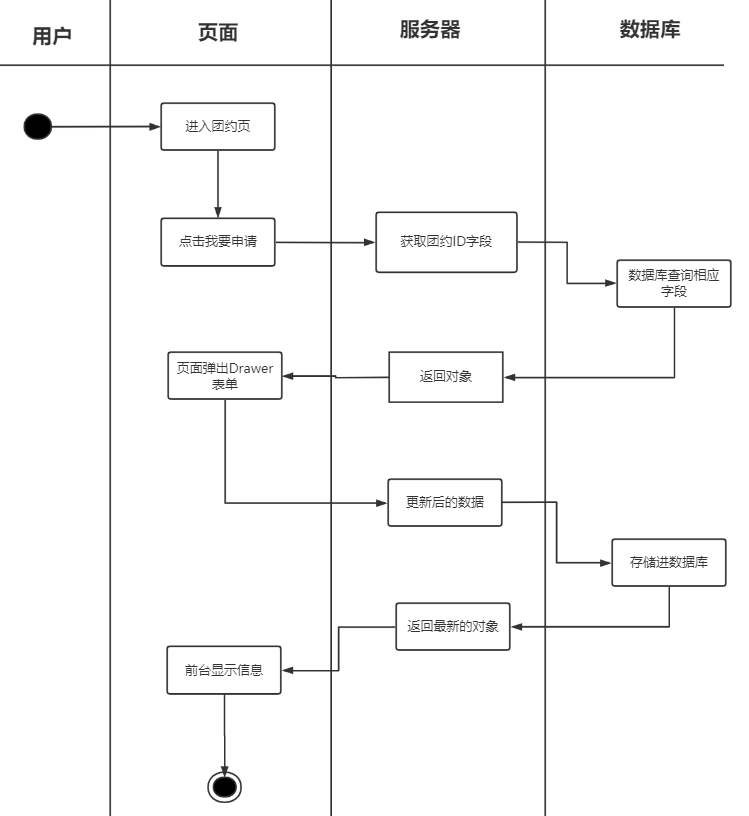


图5-12 团约管理活动图（查询）

当进行删除团约操作的时候，管理员需要在管理团约页面点击删除团约按钮，系统会弹出一个气泡选择框，如果管理员选择确认，系统成功将相关团约删除，并删除数据库的相关信息，包括此团约内的用户，他们个人团约记录中此团约的也将一并删除。

团约管理模块的活动图（删除），如图5-14所示

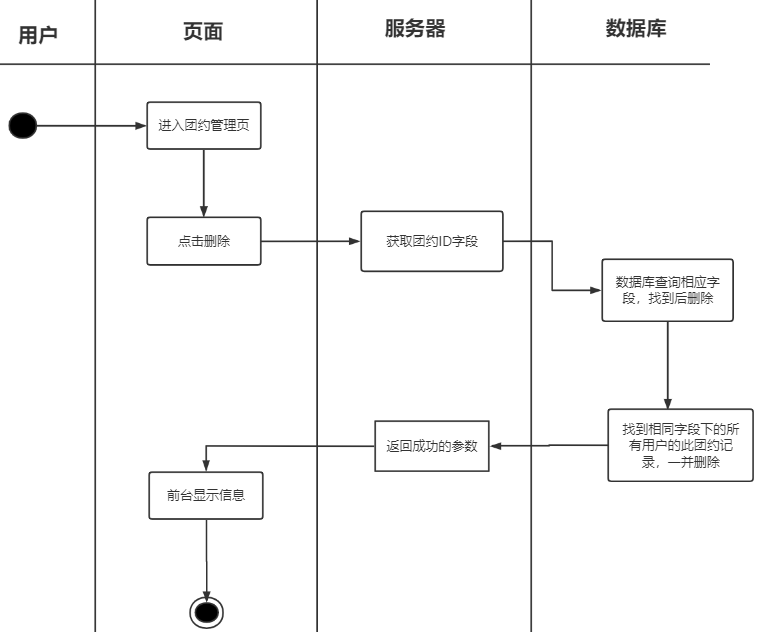


图5-12 团约管理活动图（删除）

## 5.2 系统的界面和接口说明

本系统采用MVVM架构设计模式，将整个系统的业务逻辑分为View层、Model层和ViewModel层。View对应于Activity和xml，负责View的绘制以及与用户交互，Model是实体模型，ViewModel负责完成View于Model间的交互,负责业务逻辑以“数据模型数据双向绑定”的思想作为核心，因此在View和Model之间没有联系，通过ViewModel进行交互，而且Model和ViewModel之间的交互是双向的，因此视图的数据的变化会同时修改数据源，而数据源数据的变化也会立即反应到View上。

本系统前端主要采用的是Axios异步请求传递数据，由于是组件化开发，前端页面功能被拆分成多个组件概括如下：

1. layouts 页面主体框架

（1）Index.tsx 页面主体框架组件

（2）components 主体页框架所使用的组件

（A）MainMenu.tsx 主体页导航栏组件

（B）FootMenu.tsx 主体页脚部栏组件

1. pages 每个路由标签页

（1）UserInfo 用户信息

（A）Index.tsx 用户个人信息主体组件

（B）components 用户信息组件可能使用的组件

（a）ChangeUserInfo.tsx 修改个人信息表单组件

（2）Message 我的消息

（A）Index.tsx 消息主体组件

（B）components 消息组件可能使用的组件

（a）CheckMessage.tsx 查看消息组件

（b）DealMessage.tsx 处理消息组件

（3）Contact 联系我们

（A）Index.tsx 联系我们主体组件

（4）Films 影视资源

（A）Index.tsx 影视资源主体组件

（B）components 影视资源组件可能使用的组件

（a）FilmList.tsx 影视信息列表组件

（b）FilmForm.tsx 影视信息管理组件

（5）HotFilms 热片速递

（A）Index.tsx 消息主体组件

（6）SpellGroups

（A）Index.tsx 团约主体组件

（B）components 团约组件可能使用的组件

（a）SpellGroupItem 团约个人组件

（b）SpellGroupInfo 团约详情组件

（c）SpellGroupList 团约列表组件

1. components 多个页面可能用到的公共组件

（1）Search.tsx 搜索框组件

（2）Comments.tsx 评论框组件

（3）FilmsView.tsx 影视资源框组件

本系统后端接口主要采用SpringBoot搭建，提供Http请求接口，用来接收参数，响应前端请求。具体介绍如下：

1. BaseAspectSupport AOP通用的方法类

2. LoginAuthorizationController 登录授权接口

3. UserController 用户管理接口

4. FilmController 影视资源管理接口

5. CommentController 评论管理接口

6. SpellGroupController 团约管理接口

7. DiscussionController 讨论帖管理接口

8. DiscussionItemController 讨论帖用户管理接口

9. SpellGroupsUserController 团约用户接口

本系统后端接口用来接收参数，将相关参数交由Service层进行处理。具体介

绍如下：

1. UserServiceImpl用户管理Service

2. FilmServiceImpl 影视资源挂你Service

3. CommentServiceImpl 评论管理Service

4. SpellGroupServiceImpl团约关联Service

5. DiscussionServiceImpl 讨论帖管理Service

6. DiscussionItemServiceImpl 讨论帖用户管理Service

7. SpellGroupsUserServiceImpl 团约用户管理Service

8. LoginAuthorizationServiceImpl 授权登录管理Service

## 5.3系统的具体实现

### 5.3.1 用户登录模块的实现

该模块是整个系统的入口，主要实现用户登录系统后台的功能。用户在登录界面填写用户名、密码和验证码，点击登录按钮，系统会判断输入的用户名、密码和验证码是否正确，如果正确则用户顺利登录该系统，进入主界面。用户登录图，如图5-16 所示。



图5-16 用户登录图

用户登录模块的实现流程如下：

1.用户访问网站首页，页面加载时会从接口获取验证码，用户填写用户名、密码和验证码点击“立即登录”，前端会校验用户名、密码和验证码的格式化是否正确，格式错误提交错误信息，格式正确提交到服务器端。

2.服务器端接收到用户的请求后，通过SpringMVC的@PostMapping("login")映射到login方法。由于登录接口安全性要求极高，为了防止暴力破解密码，登录接口声明注解@Limit(key = "login", period = 60, count = 20, name = "登录接口", prefix = "limit")，声明该注解的方法或者类会被LimitAspect做一层代理，LimitAspect通过lua限流脚本和Redis判断用户请求在60秒内是否超过20次，超过限制频率则会抛出LimitAccessException("接口访问超出频率限制")异常，未超过限制则进行校验验证码。

3.验证码校验，调用ValidateCodeService的check()方法传入SessionId和用户输入的验证码verifyCode参数。check()方法通过SessionId获取之前存入Redis的验证码，如果获取不到，返回验证码过期，获取到后与用户输入的验证码比对，比对错误返回验证码错误，比对成功进行用户名和密码校验

4.用户名和密码校验，使用用户名密码参数创建一个UsernamePasswordToken对象，然后使用shiro的subject调用login()方法，最终会回调ShiroRealm的doGetAuthenticationInfo()方法进行用户认证。如果根据用户名查询不到用户，抛出UnknownAccountException("账号未注册！")异常；如果查询到的用户的密码与用户输入的密码不一致，抛出IncorrectCredentialsException("用户名或密码错误！")异常；如果用户状态被锁定，抛出LockedAccountException("账号已被锁定,请联系管理员！")异常。都通过则说明用户名、密码无误，用户认证成，记录登录日志。

5.用户认证成功，服务器响应success给前端，前端跳转系统首页。认证失败则会显示各个阶段的异常信息。

### 5.3.2 用户管理模块的实现

在该模块中，主要实现对用户管理的功能。系统管理员登录成功以后，在用户管理下的用户列表界面，可以添加、删除系统中的用户，并可对已有的用户进行修改。这里重点介绍对用户的修改。当执行修改操作时，系统管理员可以对用户的基本信息、用户的角色和用户的状态进行修改。用户管理图，如图5-17所示。用户修改图，如图5-18所示。



图5-17 用户管理图

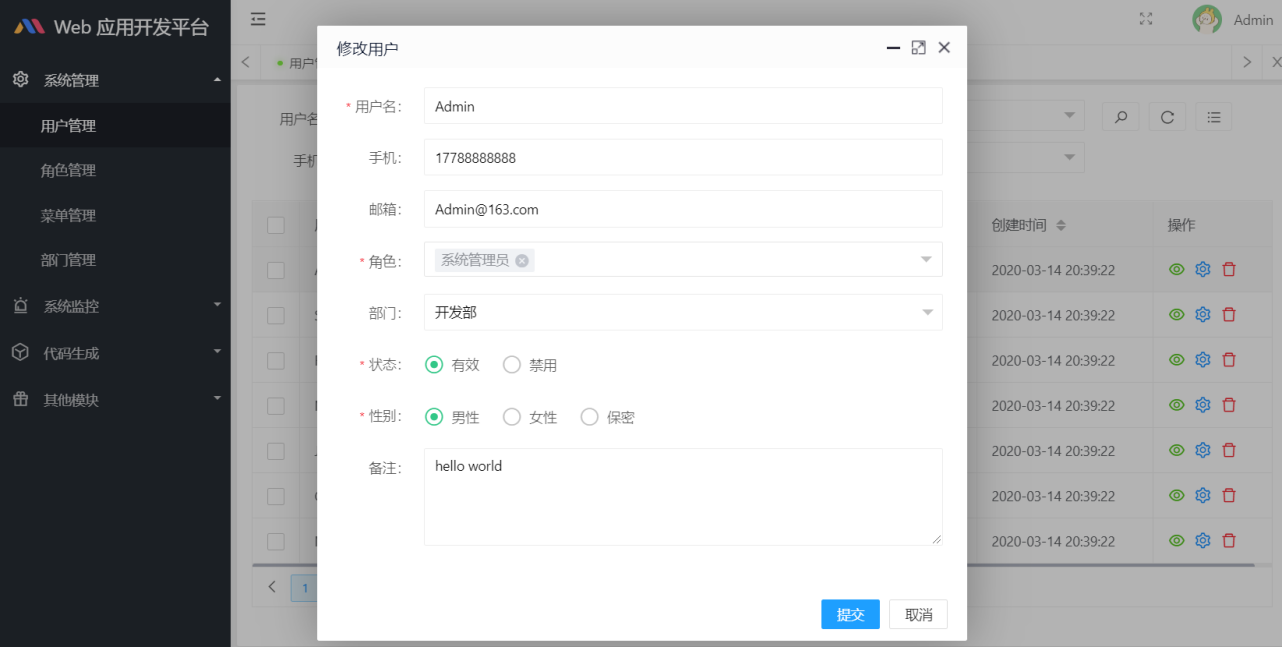


图5-18 用户修改图

用户管理模块中修改操作的实现流程如下：

1.用户访问用户管理页面，选择需要修改信息的用户，页面会弹出用户详细信息的模态窗口，并访问服务器获取用户的详细信息数据，将数据渲染到弹出窗。用户修改弹窗中的用户信息，并点击提交或者取消，点击提交时前端会校验修改后的数据与原数据是否相同，相同则提示“数据未作任何修改”，不相同则将修改后的用户信息数据提交到服务器端。

2.服务器端接收到用户的请求后，UserController通过SpringMVC的注解@PostMapping("update")映射到updateUser()方法。updateUser()方法被@RequiresPermissions("user:update")注解标识，在调用前会进行权限校验，拥有“user:update”权限则可以进行调用，无权限则返回权限不足的提示。权限校验通过后会对参数进行校验，参数中用户ID为空则抛出FwpException("用户ID为空")异常，参数不为空则继续。

3.权限、参数都校验通过后，首先将用户名和密码都置为null，将ModifyTime设置为当前时间new Date()，然后将修改的用户基本信息存储到数据库。

4.用户基本信息更新完后，需要更新用户关联的角色，先调用this.userRoleService.remove()将该用户的对应角色关联列表批量删除，然后调用userRoleService.saveBatch()将修改后最新的用户角色关联列表批量保存到数据库中。

5.修改成功后，服务器响应success给前端，前端提示用户“修改成功”。修改失败则会显示各个阶段的异常信息。

### 5.3.3 角色管理模块的实现

在该模块中，主要实现对角色管理的功能。系统管理员登录成功以后，在角色管理下的角色列表界面，可以添加新的角色、删除系统中的角色，并可对已有的角色进行基本信息的修改和角色对应的权限修改。这里重点介绍对角色的删除。当执行角色操作时，系统管理员可以点击确认或者取消。角色管理图，如图5-19所示。角色删除图，如图5-20所示。

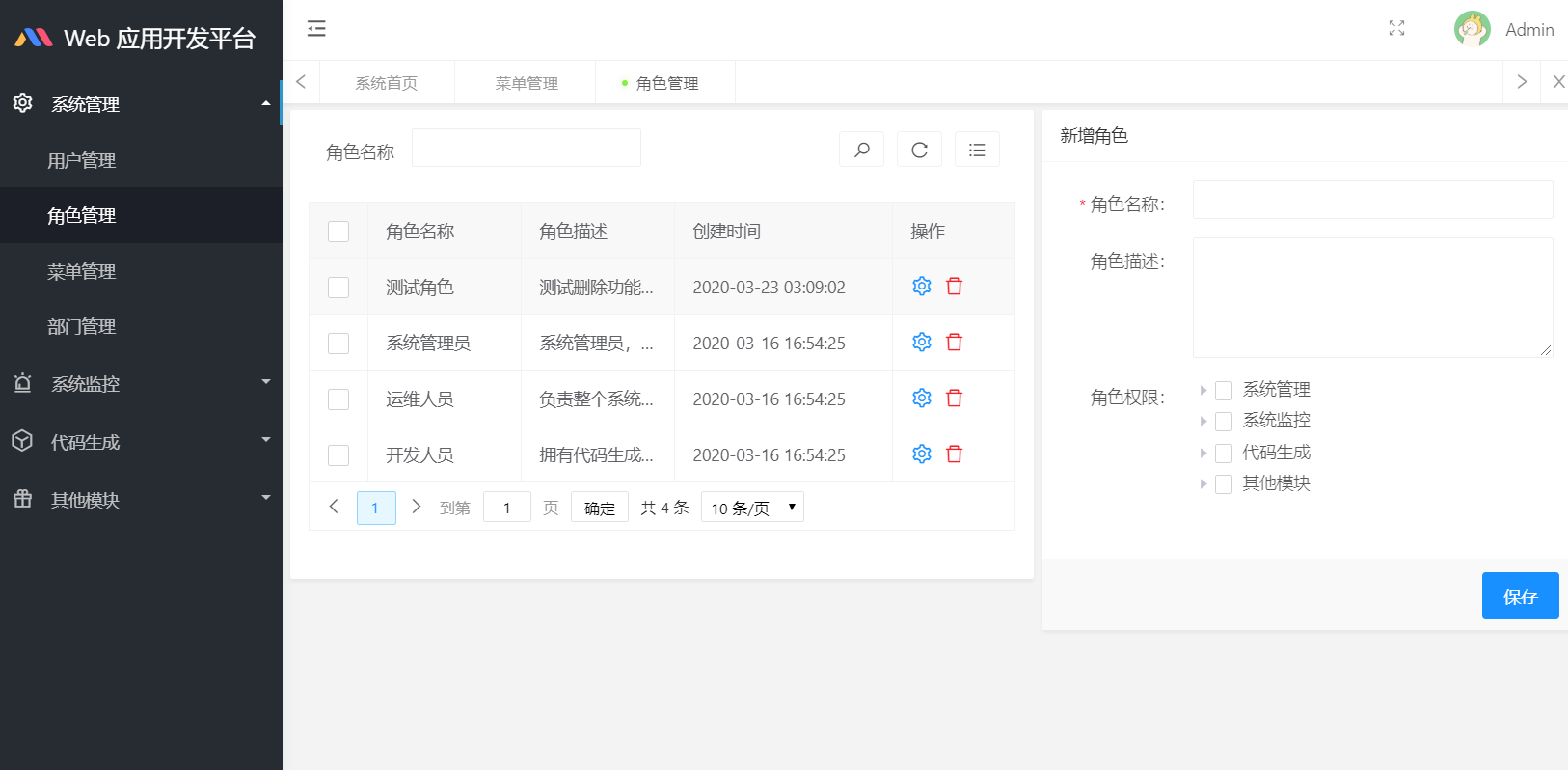


图5-19 角色管理图

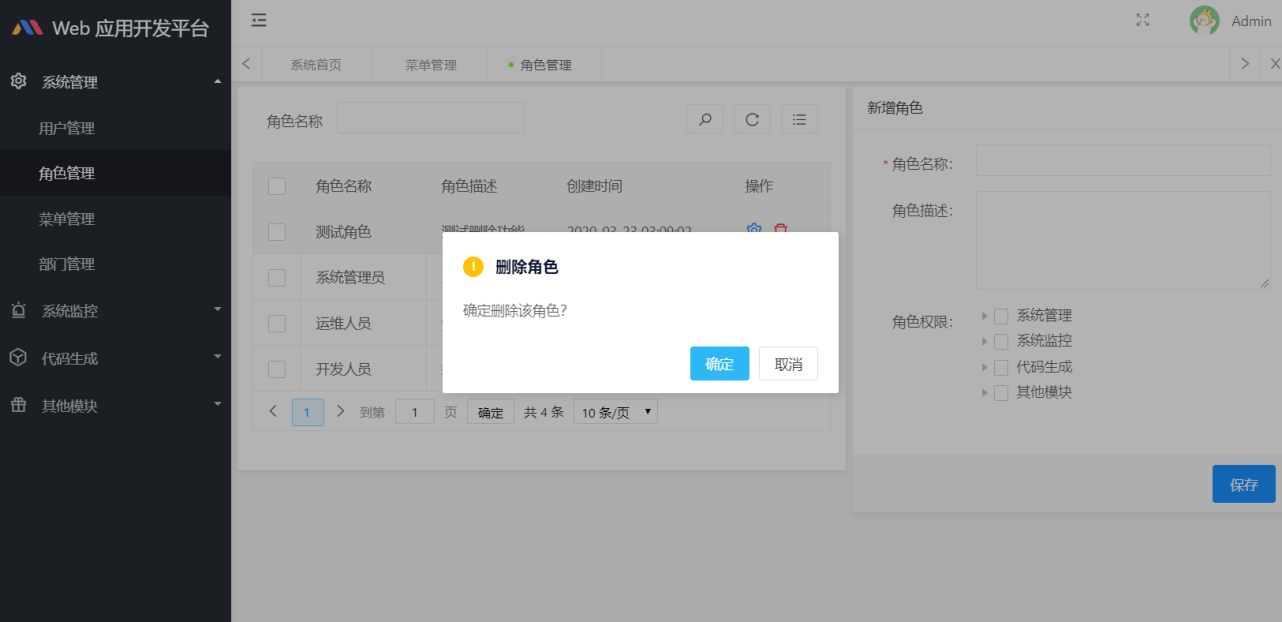


图5-20 角色删除修改图

角色管理模块中删除操作的实现流程如下：

1.用户访问角色管理页面，选择需要删除的角色，页面会弹出确定或者取消的模态窗口，用户点击取消可以进行其他操作，点击确定该角色的Id则会发送到服务器端。

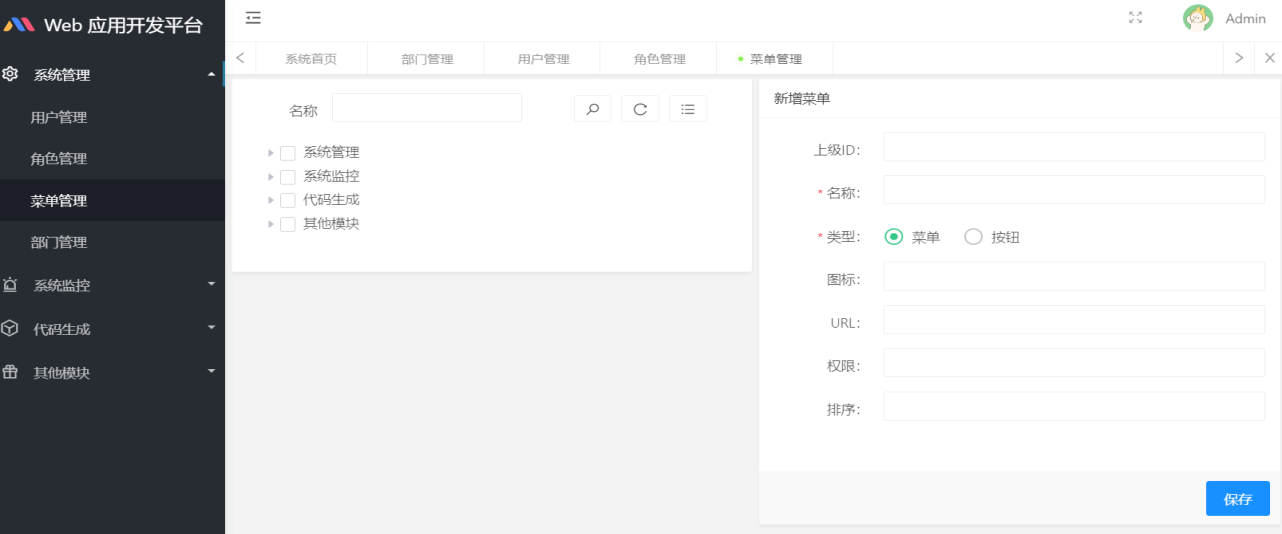
2.服务器端接收到用户的请求后，RoleController通过SpringMVC的注解@GetMapping("delete/{roleIds}")映射到deleteRoles ()方法。deleteRoles ()方法被@RequiresPermissions("role:delete")注解标识，在调用前会进行权限校验，拥有“role:delete”权限则可以进行调用，无权限则返回权限不足的提示。权限校验通过后会对参数进行校验，参数中角色ID为空则抛出FwpException("角色ID为空")异常，参数不为空则继续。

3.权限、参数都校验通过后，会先删除该角色的基本信息，然后删除该角色对应的用户关联，再删除该角色对应的菜单关联。

4.删除成功，服务器响应success给前端，前端提示用户“删除成功”。删除失败则会显示各个阶段的异常信息。

### 5.3.4 菜单管理模块的实现

在该模块中，主要实现对菜单管理的功能。系统管理员登录成功以后，在菜单管理下的菜单列表界面，可以添加、删除系统中的菜单，并可对已有的菜单进行修改。这里重点介绍对菜单的查询。菜单管理图，如图5-21所示。菜单查询图，如图5-22所示。

 图5-21 菜单查询图

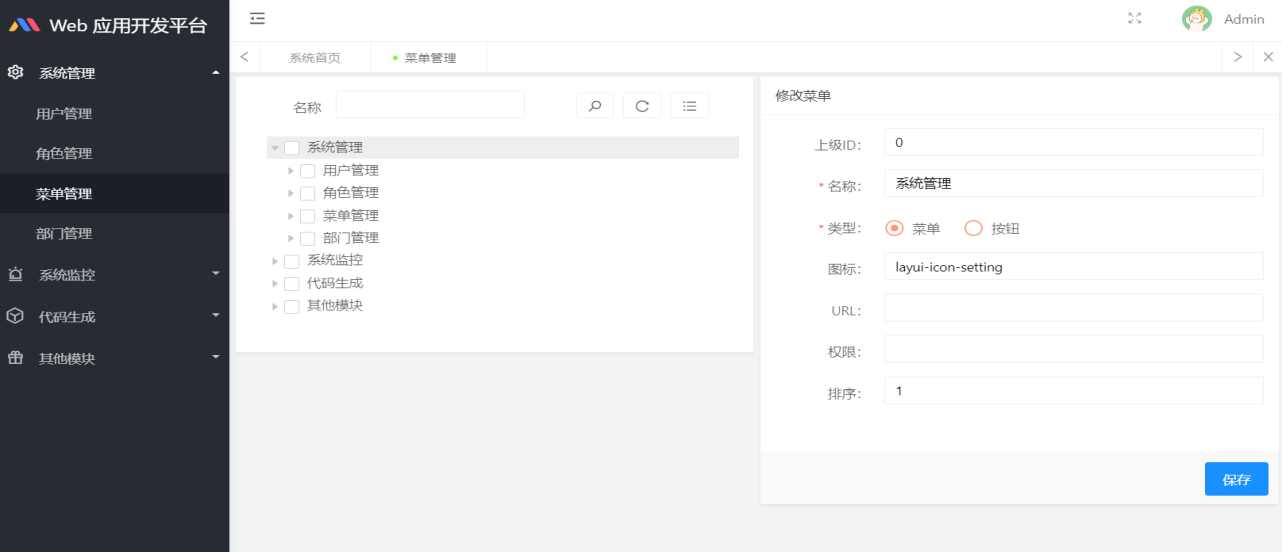


图5-22 菜单查询图

菜单管理模块中查询操作的实现流程如下：

1.用户访问菜单管理页面，页面加载时会请求system/menu。

2.服务器端接收到用户的请求后，首先ViewController的systemMenu ()方法转发到菜单列表页，systemMenu ()方法被@RequiresPermissions("menu:view")注解标识，在调用前会进行权限校验，拥有“menu:view”权限则可以进行调用，无权限则返回权限不足的提示。

3.权限校验通过后，MenuController通过SpringMVC的注解@GetMapping("tree")映射到getMenuTree ()方法。getMenuTree ()方法会调用MenuServiceImpl的findMenus()方法，findMenus()方法首先查询List<Menu>菜单列表，将菜单列表转换成前端需要的数据格式List<MenuTree<Menu>>菜单树列表，再根据父子Id关系将菜单树列表转成树形菜单MenuTree<Menu>，controller响应数据。

4.查询到结果集后，服务器响应success给前端，前端展示树形菜单。查询失败则会显示各个阶段的异常信息。

### 5.3.5 部门管理模块的实现

在该模块中，主要实现对部门管理的功能。系统管理员登录成功以后，在部门管理下的部门列表界面，可以添加、删除系统中的部门，并可对已有的部门进行修改。这里重点介绍对菜单的添加。部门管理图，如图5-23所示。部门添加图，如图5-24所示。

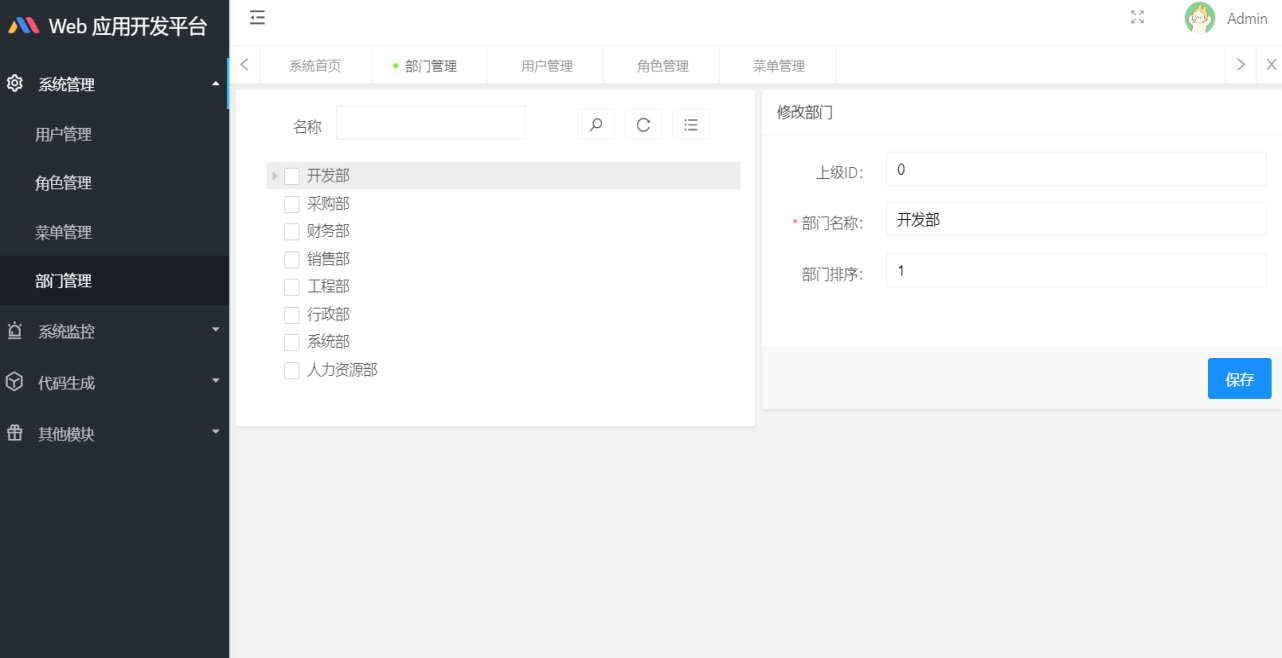


图5-23 部门管理图



图5-24 部门添加图

部门管理模块中添加操作的实现流程如下：

1.用户访问部门管理页面，选择需要添加部门的父部门，点击新增，页面会弹出部门添加的窗口，用户填写部门名称和部门排序后，点击确定该部门信息会发送到服务器端。

2.服务器端接收到用户的请求后，DeptController通过SpringMVC的注解@PostMapping映射到addDept ()方法。addDept ()方法被@RequiresPermissions("dept:add")注解标识，在调用前会进行权限校验，拥有“dept:add”权限则可以进行调用，无权限则返回权限不足的提示。权限校验通过后会对参数进行校验，参数中部门名称deptName为空则抛出FwpException("部门名称为空")异常，参数不为空则继续。

3.权限、参数校验通过后，调用DeptServiceImpl的createDept()方法进行保存部门信息。

4.添加成功后，服务器响应success给前端，前端展示最新的树形部门信息。添加失败则会显示各个阶段的异常信息。

### 5.3.6 在线用户模块的实现

在该模块中，主要实现对在线用户管理的功能。系统内的用户登录成功后，就会记录在线用户信息，以及在线用户数量。运维人员可以查询在线用户列表、对于一些非法在线用户进行下线操作。在线用户管理图，如图5-25所示。

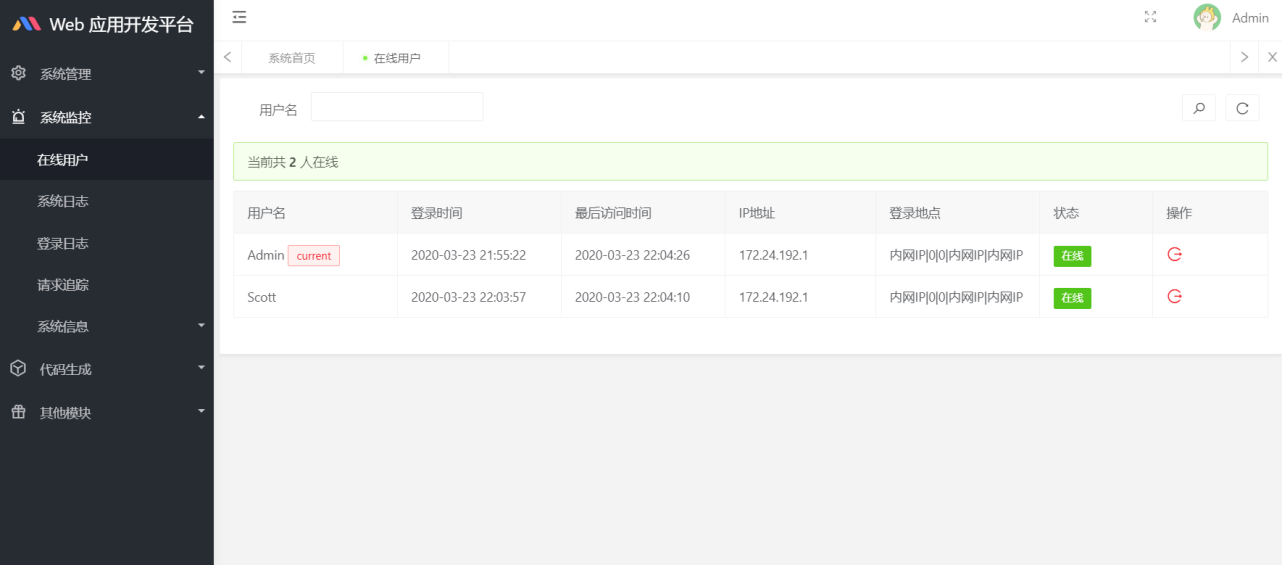


图5-25 在线用户管理图

在线用户管理模块中查询操作的实现流程如下：

1.系统内所有用户的登录均会被记录到shiro中。

2.用户登录系统，服务器端认证成功后，shiro会将用户认证成功的会话存入Redis缓存中，并将会话设置1800秒的过期时间，并且使用AtomicInteger记录在线用户的数量，AtomicInteger保证原子性和可见性，不会出现多线程脏数据的问题。

3.运维人员查看在线用户，服务器接收到用户的请求后，SessionController通过SpringMVC的注解@GetMapping("list")映射到list ()方法。list ()方法被@RequiresPermissions("online:view ")注解标识，在调用前会进行权限校验，拥有“online:view”权限则可以进行调用，无权限则返回权限不足的提示。

4.权限校验通过后，使用sessionDAO获取Redis中的所有会话未过期的在线用户，并将在线用户列表和在线用户总数返回到前端。

5.查询成功后，服务器响应success给前端，前端展示最新的用户在线信息。查询失败则会显示各个阶段的异常信息。

### 5.3.7 系统日志模块的实现

系统日志模块中，主要实现对系统日志的管理。系统内的用户登录成功后，进行相关的业务逻辑操作时，系统会获取到该请求，对于需要记录系统日志的操作方法通过@ControllerEndpoint注解进行标识，所有标识@ControllerEndpoint的方法会被ControllerEndpointAspect进行代理，获取操作人、操作描述、耗时、操作方法、方法参数、IP地址、操作地点、创建时间等信息，待此业务逻辑操作完成后，系统会将该记录信息保存到数据库中。运维人员可以在系统监控下的系统日志界面查看系统日志，并可通过操作人、操作时间、操作描述来查询特定的系统日志。系统日志管理图，如图5-26所示。

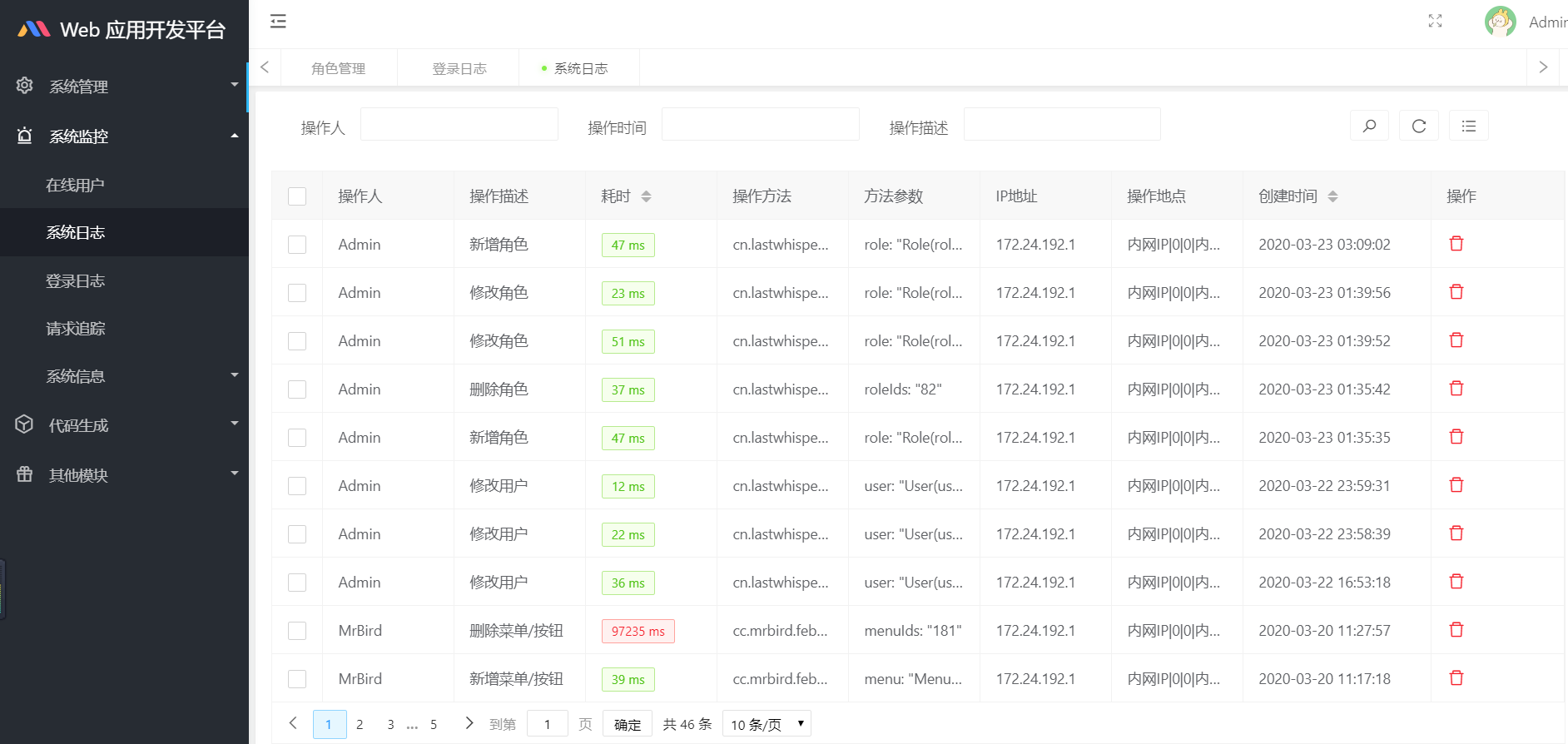


图5-26 系统日志管理图

系统日志管理模块中记录操作的实现流程如下：

1.系统内合法用户的所有操作均会被记录到系统日志，这里以系统管理员删除角色为例。

2.系统管理员在角色管理页面点击删除角色，服务器接收到删除的请求，执行RoleControllerde的deleteRoles()方法，由于该方法被注解@ControllerEndpoint(operation = "删除角色", exceptionMessage = "删除角色失败")标识，deleteRoles()方法执行的是被ControllerEndpointAspect代理过得代理方法，该代理方法会通过ProceedingJoinPoint获取目标方法的所有信息，如方法名、参数列表信息，以及HttpServletRequest中的IP地址等信息，在执行目标方法前使用System.currentTimeMillis()获取系统当前的时间戳，然后执行目标方法。目标方法执行完毕后，再获取当前时间戳，与目标方法调用前的时间戳相减获得方法执行耗时，将以上获取的信息封装成SystemLog持久化到数据库中。方法执行出现异常则将异常包装成FwpException系统异常，由统一的GlobalExceptionHandler进行处理。

### 5.3.8 代码生成模块的实现

在该模块中，主要实现代码生成的功能。开发人员点击代码生成，代码生成页面会加载数据库中的表信息，如表名、备注、表中的数据量、创建时间、修改时间信息，开发人员可以选择需要生成代码的表，点击下载即可生成生成实体类、Mapper接口、Mapper配置文件、Service接口、Service实现类，Controller类源代码。代码生成模块图，如图5-27所示。

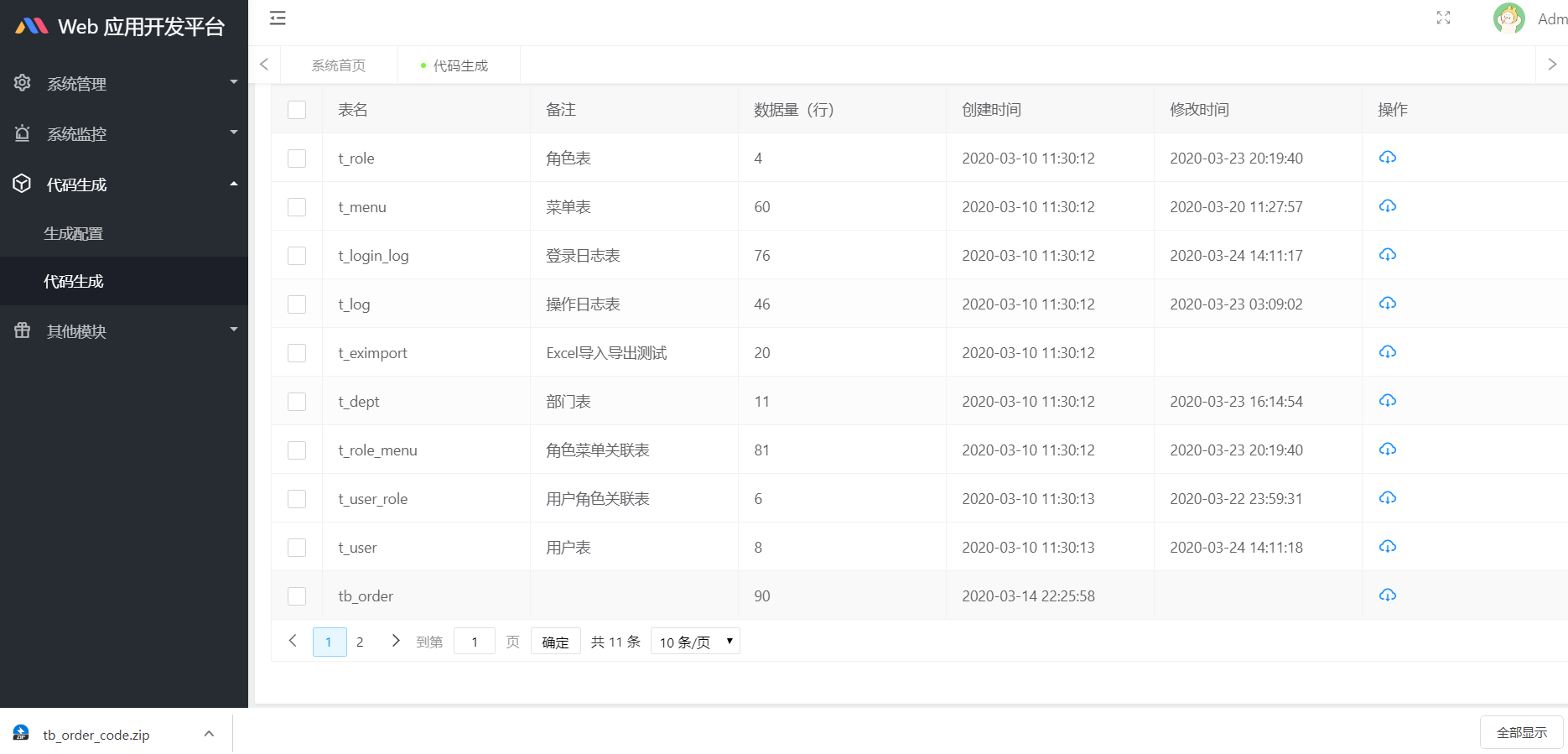


图5-27 代码生成模块图

在线用户管理模块中查询操作的实现流程如下：

1.开发人员点击下载按钮，服务器接收到用户的请求后，GeneratorController通过SpringMVC的注解@GetMapping映射到generate ()方法。generate ()方法被@RequiresPermissions ("generator:generate")注解标识，在调用前会进行权限校验，拥有“generator:generate”权限则可以进行调用，无权限则返回权限不足的提示。

2.权限校验通过后，generatorConfigService.findGeneratorConfig()获取用户填写的代码生成配置GeneratorConfig，里面包含了作者名称、基础包名、entity包名、service包名、serviceImpl包名、mapper包名、mapperXml包名、controller包名、是否去除表前缀、表前缀信息。

3.拿到生成配置后，通过generatorService.getColumns获取需要生成代码的表的列信息List<Column>，列信息包含名称、是否为主键、类型、注释、属性名称信息。

4.根据数据：生成配置、列信息，源代码模块生成对应的源代码，压缩成ZIP返回前端。

## 5.4 本章小结

本章节重点介绍系统的详细设计与实现，主要划分为系统的功能设计、系统的方法设计和系统的具体实现。在系统的功能设计中列举出8个模块对其功能设计进行详细描述，各个功能均采用活动图进行展示。系统的方法设计，根据项目传输图将前后台中核心的方法进行阐述。系统的具体实现首先简述了业务流程，然后分析各个功能模块在系统中是如何实现的，每个功能模块均配有系统截图。

6 测试

软件测试是检查实际结果与预期结果是否匹配并确保软件系统无缺陷的活动。软件测试还有助于识别产品与实际需求不符或是缺失项。测试活动既可以手动完成，也可以使用自动化工具完成。有些人更喜欢将软件测试称为白盒和黑盒测试。

软件测试的目标：

1. 在给定的产品中尽可能多地发现错误(或bug)。
2. 演示一个给定的软件产品与它的需求规格匹配。
3. 使用最小的成本和努力来验证软件的质量。
4. 生成高质量的测试用例，执行有效的测试，并发布正确和有用的问题报告。

## 6.1 测试的意义

软件测试可确保所提供的软件符合预期的功能要求,或者该软件的实际输出功能与预期的功能之间是否存在差异,并且该软件是否满足功能性,容错性和稳定性的要求。一般来说,意外的功能,隐藏的错误和软件缺陷是从用户的角度进行测试的。从开发人员的角度来看,我们希望开发的软件没有错误,并且可以处理异常,成为成熟,可靠,高质量的“艺术品”,使开发的软件可以满足客户要求,且易于使用,以提高用户使用软件的信心。

## 6.2 测试环境

硬件环境：

Windows操作系统，CentOS 7 操作系统

软件环境：

IntelliJ IDEA、MySQL、Chrome、VsCode

## 6.3 功能测试

功能测试也叫做行为测试，也就是对系统各个功能进行验证，根据测试用例，检查系统是否达到用户需求。功能测试也叫黑盒测试或数据驱动测试。测试人员不需要去了解系统的内部结构和内部编码，只需要测试系统的各项功能，按照需求编写相关的测试用例。用预期结果和实际结果进行对比，从而提高系统性能，提高用户体验，达到用户需求。

### 6.3.1 用户登录测试

用户进入到首页界面后，选择QQ授权登陆，在第三方平台获取用户信息后拿到Code，在服务器端进行判断，如果没有使用可以顺利登录该系统，否则不能正常登录。用户登录的测试用例如表6-1所示。

表6-1 用户登录测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例ID** | **QQ** | **是否授权** | **Code是否被使用** | **测试结果** |
| 1 | 1415865614 | 是 | 否 | 登录成功 |
| 2 | 1415865614 | 是 | 是 | Code被使用，登录失败 |
| 3 | 1415865614 | 否 | 否 | 拒绝授权，登录失败 |
| 4 |  |  |  | 拒绝授权，登录失败 |

步骤如下：

1.运行系统，用户切换到系统登录界面。

2.测试用例1，用户使用QQ登陆，确认授权，Code未被使用，登录成功。

3.测试用例2，用户使用QQ登陆，确认授权，Code被使用，Code被使用，登录失败。

4.测试用例3，用户使用QQ登陆，未授权，Code未被使用，拒绝授权，登录失败。

5.测试用例4，用户未使用QQ登陆，拒绝授权，登录失败。

### 6.3.2 修改用户信息测试

一般用户在个人信息管理界面下，点击修改个人信息，界面上会弹出一个包含Form表单的Drawer弹出层，显示出用户的用户名、性别、电话、QQ、微信、邮箱等信息，选择想要修改的属性，输入或选择填充内容，点击确定，系统提示操作成功。修改用户信息的测试用例如表6-2所示。

表6-2 修改用户信息测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **用例ID** | **是否填充信息** | **是否点击确认** | **操作结果** |
| 1 | 是 | 是 | 操作成功 |

续表6-2

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 未全部填充 | 否 | 操作成功，只修改填充的信息 |
| 3 | 否 | 是 | 操作失败 |
| 4 | 是 | 否 | 操作失败 |

1.运行系统，一般用户登录系统，进入用户列表界面。

2.测试用例1，用户填充了要修改的信息，点击确认，操作成功。

3.测试用例2，用户填充了部分要修改的信息，没有填充完整，点击确认，操作成功，只有填充过的信息被修改。

4.测试用例3，用户没有填充要修改的信息，点击确认，操作失败。

5.测试用例4，用户填充了要修改的信息，没有点击确认，操作失败。

### 6.3.3 上架影视资源信息测试

系统管理员在系统管理下的影视资源管理界面，可以看到目前所有的影视资源信息，点击上架，界面上会弹出一个包含Form表单的Drawer弹出层，显示出影视名称、影视封面、上映时间、时长、导演、演员、影视类型、详细描述等信息，输入或选择填充内容，点击确定，系统提示操作成功。上架影视资源信息的测试用例如表6-3所示。

表6-3删除角色信息测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **用例ID** | **是否填充信息** | **是否点击确定** | **测试结果** |
| 1 | 是 | 是 | 操作成功 |
| 2 | 是 | 否 | 操作失败 |
| 3 | 未填充完整 | 是 | 操作失败 |
| 4 | 否 | 是 | 操作失败 |

步骤如下：

1.运行系统，系统管理员登录系统，进入影视资源管理界面。

2.测试用例1，填充了要上架的影视资源的信息，点击确认，操作成功。

3.测试用例2，填充了要上架的影视资源的信息，未点击确认，操作失败。

4.测试用例3，填充了要上架的影视资源的部分信息信息，未填充完整，点击确认，操作失败。

5.测试用例4，未填充要上架的影视资源的信息，点击确认，操作失败。

### 6.3.4 查询影视资源信息测试

系统管理员在影视资源管理界面下，可以根据影视资源名称进行查询。在搜索框内填入部分或完整的影视资源名称，点击查询按钮，界面上会显示出所有符合要求的影视资源信息。查询影视资源信息的测试用例如表6-4所示。

表6-4 查询影视资源信息测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **用例ID** | **输入** | **是否点击查询** | **测试结果** |
| 1 | 战狼 | 是 | 操作成功，显示《战狼》的信息 |
| 2 | 战 | 是 | 操作成功，显示包含“战”的影视信息 |
| 3 | 战狼 | 否 | 操作失败 |
| 4 | 战 | 否 | 操作失败 |
| 5 |  | 是 | 操作失败 |

步骤如下：

1.运行系统，系统管理员登录系统，进入影视资源界面。

2.测试用例1，输入了影视资源名称，点击查询，该影视资源的信息查询成功。

3.测试用例2，输入了部分影视资源名称，点击查询，显示系统内所有影视资源名称中含有该字符的影视资源，查询成功。

4.测试用例3，输入了影视资源名称，没有点击查询，操作失败。

5.测试用例4，输入了部分影视资源名称，没有点击查询，操作失败。

6.测试用例5，没有输入，点击查询，操作失败。

### 6.3.5 更新影视资源信息测试

系统管理员在影视资源管理界面下，点击影视资源标题或者查看详情按钮，进入影视资源详情界面，点击修改按钮，界面上会弹出一个包含Form表单的Drawer弹出层，显示出该影视资源信息的影视名称、影视封面、上映时间、时长、导演、演员、影视类型、详细描述。选择想要修改的属性，输入或选择填充内容，点击确定，系统提示操作成功。更新影视资源信息的测试用例如表6-5所示。

表6-5 更新影视资源信息测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **用例ID** | **是否填充信息** | **是否点击确认** | **操作结果** |
| 1 | 是 | 是 | 操作成功 |

续表6-5

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 是 | 否 | 操作失败 |
| 3 | 未填充完整 | 是 | 操作失败 |
| 4 | 否 | 是 | 操作失败 |

步骤如下：

1.运行系统，管理员登录系统，进入影视资源更新界面。

2.测试用例1，填充了要修改的信息，点击确认，操作成功。

3.测试用例2，填充了要修改的信息，没有点击确认，操作失败。

4.测试用例3，用户填充了部分要修改的信息，没有填充完整，点击确认，操作失败。

5.测试用例4，没有填充要修改的信息，点击确认，操作失败，

### 6.3.6 下架影视资源测试

用户在各个模块中进行的新增、删除、修改等业务逻辑操作，系统会自动记录该操作的日志信息，并保存到数据库中。例如，系统管理员想要删除系统中的用户，在用户列表界面，选择想要删除的用户，点击删除，系统提示操作成功。

操作成功后，系统日志中会自动新增一条删除用户的记录。新增系统日志的测试用例（以删除用户为例）如表6-6所示。

表6-6 下架影视资源测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例ID** | **是否选中用户** | **是否点击删除** | **执行结果** | **测试结果** |
| 1 | 是 | 是 | 执行成功 | 新增记录 |
| 2 | 是 | 否 | 执行失败 | 无新增记录 |
| 3 | 否 | — | 执行失败 | 无新增记录 |

步骤如下：

1.运行系统，系统管理员登录系统。

2.测试用例1，进入用户列表界面，选中用户，确认删除，执行成功。进入系统日志界面，新增了一条删除用户的记录。

3.测试用例 2，进入用户列表界面，选中用户，没有确认删除，执行失败。进入系统日志界面，无新增记录。

4.测试用例3，进入用户列表界面，没有选中用户，执行失败。进入系统日志界面，无新增记录。

### 6.3.7 查询系统日志测试

运维人员在系统日志界面，可以根据日志记录的用户名或者用户操作进行查询。在文本框中填入用户名或者用户操作的关键字，点击查询按钮，界面上会显示出所有符合要求的系统日志记录。查询系统日志记录的测试用例如表6-7所示。

表6-7 查询系统日志记录测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **用例ID** | **输入** | **是否点击查询** | **测试结果** |
| 1 | Admin | 是 | 操作成功，显示用户“admin”的所有操作记录 |
| 2 | A | 是 | 操作成功，显示包含“A”/“a”的用户的所有操作记录 |
| 3 | 菜单 | 是 | 操作成功，显示所有对“菜单”的操作记录 |
| 4 | 删除 | 是 | 操作成功，显示所有关于“删除”的操作记录 |
| 5 | A | 否 | 操作失败 |
| 6 |  | — | 操作失败 |

步骤如下：

1.运行系统，运维人员登录系统，进入系统日志界面。

2.测试用例1，输入了用户名Admin，点击查询，操作成功，显示用户“Admin”的所有操作记录

3.测试用例2，输入了字符A，点击查询，操作成功，显示包含“A”/“a”的用户的所有操作记录

4.测试用例3，输入了可操作的对象名，点击查询，操作成功，显示所有关于该对象的操作记录。

5.测试用例4，输入了系统可进行的操作名，点击查询，操作成功，显示所有关于该操作的记录。

6.测试用例5，输入了字符，没有点击查询，操作失败。

7.测试用例6，没有输入，操作失败。

### 6.3.8 代码生成测试

开发人员在代码生成界面，可以根据表结构和模板进行代码生成。在文本框中填入表名，点击查询按钮，界面上会显示出所有符合要求的表信息。代码生成的测试用例如表6-8所示。

表6-8 代码生成测试用例表

|  |  |  |
| --- | --- | --- |
| **用例ID** | **是否点击生成** | **测试结果** |
| 1 | 是 | 操作成功，下载源代码ZIP |
| 2 | 否 | 操作失败 |

步骤如下：

1.运行系统，开发人员登录系统，进入代码生成界面。

2.测试用例1，点击生成按钮，操作成功，下载源代码ZIP。

3.测试用例2，没有点击生成按钮，操作失败。

## 6.4 本章小结

本章重点讲述了软件测试环节，我们先介绍了测试需要的环境，然后通过功能测试、WeTest页面测试和服务器响应速度测试对软件测试进行阐述，功能测试主要是列举系统中的几个操作案例，通过测试用例表来核查某个程序路径是否满足某个特定的需求。WeTest页面测试主要是测试用户实际浏览体验，找到页面单个瓶颈资源，提高用户体验。服务器响应速度测试主要是借助17CE监测平台实现区域分析，快速找到网站在哪些地方速度慢。从而进一步优化系统。

7 结束语

## 7.1 总结

本文主要论述了基于SSM的影视推荐系统的设计与实现的工作。本系统的主要目的是希望让广大用户能再现如今网络中爆发式增长的影视资源中迅速的找到自己想看的电视剧、电影或者综艺节目，节省大家再“淘片”过程中浪费太多的时间。在系统的设计过程中，我们按照软件工程设计过程中的各个阶段，主要从需求分析、总体设计、详细设计与实现、测试等这几个步骤进行并详细描述过程。在系统的开发过程中按照统一的代码风格进行设计与编码，同时力求界面大方简约又不失美感，希望给用户一个良好的体验。

通过本论文的设计与实现，为系统管理员和一般用户提供了一个简单方便的影视资源推荐系统。本系统要做的工作主要有：

1. 开发系统前需要调查和分析豆瓣、爱奇艺、腾讯视频等各类影视播放平台都会注意的情况，分析他们的优势与不足，取其精华去其糟粕。
2. 分析当下流行的主流技术与框架，根据他们各自的优势于局限性，选择最合适本系统的开发技术。
3. 对系统进行需求分析，总体设计，并确定系统详细的功能结构图，让读者对我们系统又一个初步的了解。
4. 进行系统的功能和数据库设计。
5. 完成影视推荐平台的开发工作，实现一般用户在网页浏览各类影视资源的详细信息以及与其他用户的交流。管理员在后台也能清晰迅速的对数据进行管理。
6. 进行功能测试，并对测试结果进行分析，找出本系统存在的不足之处并针对性的进一步优化系统。

软件最终完成，基本实现最初设计的功能，能够供各类用户在本系统能够迅速快捷的找到自己需要的信息。但是系统还尚不足之处，计划在以后的的生活和工作中能够通过学习逐步完善该系统。

## 7.2 展望

经过数月的不懈努力，本次毕业设计基本已经完成，本系统功能大体已经实现但仍存在一些不足之处。

1. 前台采用React框架，他是一款企业级开发框架，对于小型项目来说，它自身带有的包以及工具类太多，导致页面首次加载会出现过慢的问题，计划后期采用懒加载以及将部分包和工具类从外部引入，借此来提高本系统的首次运行速度。
2. 系统中部分页面的组件逻辑不够全面，在获取后台数据时或者用户在进行操作的时候，页面会产生抖动，影响用户的使用体验。

自己虽然竭尽所能使用所学的知识以及技术进行创作，但是通过本次毕业设计再一次深深地认识到自身知识与经验上的不足，不能很好地将理论与实践进行结合，逻辑思维不够严密。论文在内容上和表述上存在一定疏漏，加之时间和精力有限，许多问题没有真正的消化和解决，有待进一步的思考和研究。望在以后的学习和工作中，以更加严格的标准要求自己。要善于发现问题、思考问题和解决问题。在以后的生活中，自己会更加努力学习专业知识，不断超越自我，迎接更加美好的未来。

# 附 录

配置服务器返回数据以及报错：

import { message } from 'antd';

import type { RequestConfig } from 'umi';

interface errorProps {

  response: Response;

  name: string;

  data: any;

}

type codeMessageProps = Record<number, string>;

// 服务器返回的错误代码

const codeMessage: codeMessageProps = {

  200: '服务器成功返回请求的数据。',

  201: '新建或修改数据成功。',

  202: '一个请求已经进入后台排队（异步任务）。',

  204: '删除数据成功。',

  400: '发出的请求有错误，服务器没有进行新建或修改数据的操作。',

  401: '用户没有权限（令牌、用户名、密码错误）。',

  403: '用户得到授权，但是访问是被禁止的。',

  404: '发出的请求针对的是不存在的记录，服务器没有进行操作。',

  406: '请求的格式不可得。',

  410: '请求的资源被永久删除，且不会再得到的。',

  422: '当创建一个对象时，发生一个验证错误。',

  500: '服务器发生错误，请检查服务器。',

  502: '网关错误。',

  503: '服务不可用，服务器暂时过载或维护。',

  504: '网关超时。',

};

// 异常处理

const errorHandler = (error: errorProps): Response => {

  const { response } = error;

  if (error.name === 'BizError') {

    // 这个是请求数据中的错误代码的处理方式，即请求成功, 且反回错误处理代码（adaptor中的success为false的时候）

    message.error(error.data.info, 2);

  } else if (response && response.status) {

    // 这是请求错误中的处理方式

    const errorText =

      codeMessage[response.status as number] || response.statusText;

    const { status } = response;

    message.error(`请求错误:${status},${errorText}`, 3);

  } else if (!response) {

    message.error(`请求错误:网络发生异常,无法连接服务器`, 3);

  }

  return response;

};

export const request: RequestConfig = {

  timeout: 2000,

  errorHandler,

  errorConfig: {

    adaptor: (resData) => {

      return {

        ...resData,

        success: true,

        errorMessage: resData.info || '请求失败',

      };

    },

  },

  headers: {

    'Content-Type': 'application/json',

  },

配置数据请求端口以及请求规范：

import { request } from 'umi';

import type { RequestOptionsInit } from 'umi-request';

interface ServiceProps {

  (url: string, params?: RequestParams): Promise<any>;

}

interface RequestParams extends RequestOptionsInit {

  method?: 'get' | 'post' | 'put' | 'delete';

  skipErrorHandler?: boolean | undefined;

}

export const baseURL: string = 'https://wenmeng.online/FilmFriend';

const services: ServiceProps = (url, params) => {

  return request(baseURL + url, {

    ...params,

    credentials: 'include',

  });

};

export default services;

配置TypeScript类型接口

type carFriendProps = {

  poolingcarid: number;

  poolinguserid: number; // 发起人id

  userids?: string; //  加入的id

  aboutpay?: number; // 大概要求

  totalnum: number; //  共需要人数

  getnum: number; //  需要人数

  starttime?: number; //  发起的时间

  endtime?: number; //  结束的时间

  readtime: number; //  最早的时间

  gotime: number; //  最晚出发的时间

  readyplace: string; //  出发地

  goplace: string; // 目的地

  poolinglimit?: number; // 限制

  poolingstatus?: number; //  状态

  leaveids?: string; // 退出，

};

参考文献

[1]李天赐.基于SpringBoot卫生备案系统的设计与实现[J].网络安全技术与应用,2018(08):13-28

[2]叶云鹏.基于SpringBoot的家政服务平台设计[J].科技广场,2017(03):32-37

[3]王丹, 孙晓宇, 杨路斌, et al. 基于SpringBoot的软件统计分析系统设计与实现[J]. 软件工程, 2019, 022(003):P.40-42.

[4]叶方超, 张思扬, 李传锴. 基于SpringBoot的旧物回收商城的设计与实现[J]. 智能计算机与应用, 2019(5).

[5]温晓丽.基于SpringBoot微服务架构的城市一卡通手机充值支撑系统研究[J].电子产品世界,2017(10):35-46

[6]李沛.基于SpringBoot的职来职往小程序开发[J].中国科技信息,2018(11):14-25

[7]郭海智, 冯玉松. 基于Spring+SpringMVC+Mybatis的业余足球俱乐部管理系统[J]. 电脑知识与技术, 2019, 15(09):77-78+82.

[8]夏禹. 基于FreeMarker、ECharts以及SSM的数据可视化平台[J]. 电脑编程技巧与维护, 2019(7).

[9]孙佩刚. 基于FreeMarker的测试文档生成方法[J]. 电脑编程技巧与维护, 2018, 000(006):17-19,26.

[10]王杉文. 基于SpringBoot+Shiro的权限管理实现[J]. 电脑编程技巧与维护, 2019(9).

[11]倪晓锋. 基于SSM和Shiro的火电SIS系统的设计与实现[J]. 工业控制计算机, 2019(9).

[12]曹灿, 刘志刚. 基于SSH和Layui的工程科学前沿与实践系统[J]. 工业控制计算机, 2019, 32(02):94-95+99.

[13]李建华, 夏汛, 罗明全. 基于ThinkPHP和Redis的高并发微信公众号开发的研究与实现[J]. 计算机应用与软件, 2019, 36(02):114-118.

[14]葛宝龙, 邓欣, 秦嘉罗. SpringBoot集成Redis缓存在电信运营系统中的应用研究[J]. 中国新通信, 2017(23):106-108.

[15]朱少民.软件测试方法和技术(第3版)[M].北京：清华大学出版社，2014.

**致 谢**

终南幽幽，雁塔相伴，是我美丽的校园。转眼间，我已经在美丽的xx度过了四个年头。四年，这是我人生中十分重要的四年，我有幸能够接触到这些不仅仅传授我知识、学问，而且从更高层次指导我的人生与价值追求的良师。他们使我坚定了人生的方向，获得了追求的动力，留下了大学生活的完美回忆。在此，我真诚地向我尊敬的老师们和母校表达我深深的谢意!

首先感谢我的指导老师张国平老师。从论文的选题到结构安排，从资料到文字润饰，都凝聚了他超多的心血。在这篇论文的写作过程中，张老师不辞辛劳，多次与我就论文中许多核心问题作深入细致地探讨，给我提出切实可行的指导性推荐，并细心全面地修改了我的论文。张国平老师这种一丝不苟的负责精神，使我深受感动。更重要的是张老师在指导我的论文的过程中，始终践行着“授人以鱼，不如授之以渔”的原则。他常教导我要志存高远，严格遵守学术道德和学术规范，为以后的继续深造打好坚实的基础。在此，请允许我向尊敬的张国平老师表示真挚的谢意!

非常感谢大家在我的毕业设计中，给予我极大的帮助，使我对整个毕业设计的思路有了总体的把握，并耐心的帮我解决了许多实际问题，使我有了很大的收获。感谢这些年来传授我知识的老师们，更要感谢那些对我学习上支持和鼓励的人。

由于水平有限，设计作品有很多不足之处，请老师和同学们多多批评、指正。谢谢！

在此，我表示衷心的感谢和崇高的敬意！