石家庄铁道大学四方学院毕业设计

四方出版社管理系统的设计与实现

**Si Fang Design and Implementation of Publishing House Management System**

**2019 届 计算机 系**

**专 业 计算机科学与技术**

**学 号**

**学生姓名 韩朝琪**

**指导教师 安凌鹏**

**完成日期 2023年 月 日**

**毕业设计成绩单**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学生姓名 | 韩朝琪 | | 学号 | 20218302 | 班级 | 方21.5Z-1 | | 专业 | | 计算机科学与技术 | |
| 毕业设计题目 | | | 四方出版社管理系统的设计与实现 | | | | | | | | |
| 指导教师姓名 | | | 安凌鹏 | | | |  | | | |  |
| 指导教师职称 | | | 讲　师 | | | |  | | | |  |
| 评 定 成 绩 | | | | | | | | | | | |
| 指导教师 | |  | | | | | 得分 | |  | | |
| 评阅人 | |  | | | | | 得分 | |  | | |
| 答辩小组组长 | |  | | | | | 得分 | |  | | |
| 成绩：  综合评定成绩为：  系主任签字：  年 月 日 | | | | | | | | | | | |

毕业设计任务书

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题　目 | 基于nodejs+vue的出版社管理系统 | | | | | | | |
| 学生姓名 | 韩朝琪 | 学号 | 20218302 | 班级 | 方2105z-1 | 专业 | 计算机科学与技术 | |
| 承担指导任务单位 | | 石家庄铁道大学  四方学院 | | | 导师姓名 | 安凌鹏 | 导师职称 | 讲师 |
| 1. 研究内容   基于nodejs+vue的出版社管理系统是一个功能性的用户级系统，此应用会提供以下模块：约稿模块、出版模块、库存模块、系统管理、作者管理。  在本系统中使用vue-router管理各个页面间的访问，使用node.js搭建后端的API接口，并完成对密码的加密，使用axios异步与后端交换数据。旨在用于探索前后端分离式项目在开发中的各种优劣以及模块化的开发思路在实战中的应用。   1. 基本要求 2. 熟练掌握JavaScript开发语言与node.js编译器； 3. 熟悉整个项目的结构及其各结构所能实现的功能； 4. 熟悉node.js与vue代码的交互原理； 5. 熟悉前后端分离的开发模式； 6. 主要技术指标   操作系统：Windows 10  开发工具：node.js、postman与VScode  环境配置：node.js自带虚拟器或实机运行方式   1. 应收集的资料及参考文献   由于采用前后端分离的开发模式，所以应收集前端路由的解决方案及状态管理的解决方案。由于node没有提供连接数据库、对密码的加密的方式，所以应收集所对应的解决方案。  本项目使用模块化的开发模式，需收集模块化的开发思想，及模块发开发所需要的技术储备。  对于vue的一些难题与理论知识，应多去参阅vue官方开发者网站所提供的技术文档，更加了解vue系统以及控件与系统的原理。   1. 进度安排   第1周：任务分配和开题报告  第2周：需求和可行性调研  第3周：概要设计  第4周—第13周：详细设计和代码设计  第14周—第16周：写论文和文献翻译  第17周：成绩综合评定 | | | | | | | | |

毕业设计开题报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题　目 | 基于nodejs+vue的出版社管理系统 | | | | | | | |
| 学生姓名 | | 韩朝琪 | 学号 | 20218302 | 班级 | 方2105z-1 | 专业 | 计算机科学与技术 |
| 一、研究背景  当前的社会,是信息化和网络化的时代。就国内出版行业而言,国内多数的出版社都已经将信息化工作作为企业重要的核心竞争力之一。希望利用较为先进的管理模式和技术手段,结束与计算机和网络技术,实现出版行业的办公自动化、数据数字化、管理自动化、商务网络化,同时实现数据的高校共享,以及基于行业数据的数据分析和挖掘。我的图书出版管理系统是一个基于人员管理与发行管理的信息化系统,该系统针对出版设约稿、人员管理、出版、库存等环节实现信息化支持。系统建设过程中提出了“以财务为核心,以图书出版的生产、发行业务为主线”的设计思路,并将这一思路贯穿在整个系统开发过程中，采用了软件工程和项目管理的基本思想。本项目主要是对图书约稿管理、图书发行管理、出版统计查询管理、系统维护管理等相关的子系统进行了分析与设计,工作重点针对约稿模块、出版模块、库存模块、系统管理、作者管理、人事管理等子系统进行了开发实现。  二、研究内容   1. 基础信息模块   用户管理:这里可以看到作家姓名手机号等信息，并对作家进行管理  书籍信息管理:可以看到书籍作者、当前进度等信息，并对书籍管理  作品管理:作家对自己的书稿进行管理  编辑管理:对编辑的职位基本信息进行管理  题材信息管理:对书籍的题材分类进行管理   1. 业务管理模块   投稿审核:根据作家的投稿以及出版社的需求进行审核  编辑意见:对已通过审核的书稿给出建议  书稿校对:对以完成修改的书稿进行文字校对，为最后的出版进行最后的准备  约稿发布:对所有作家发布出版社需要的约稿  **图1 系统功能图**  三、预期达到的目标  1.设计一款集出版社管理等功能为一体的平台为用户带来了应对人工方式繁琐易出错的解决方案。  2.实现用户访问安全性。  3.用户权限管理、会话管理。  4.用户的界面友好性，操作友好性。  5.设计数据库、开发时的规范。  6.实现对功能的测试。  四、研究方案  存储方案：采用mySql数据库的方式进行存储.。  开发方案： 使用express+vue全家桶以及其他第三方库实现，使用VS Code软件完成代码的编写  界面跳转方案： 使用vue-router管理页面间的跳转。  运行方案： 使用node.js 18.13.0版本运行。  五、参考文献  [1] [基于Vue.js的表单可视化构建系统的设计与实现](https://www.zhangqiaokeyan.com/academic-degree-domestic_mphd_thesis/020312272095.html) [A] [. 牛仁腾](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E7%89%9B%E4%BB%81%E8%85%BE&option=202) . 2017  [2] [基于Node.js中间层Web开发的研究与实现——以微信图书借阅平台为例](https://www.zhangqiaokeyan.com/academic-journal-cn_hebei-journal-industrial-science-technology_thesis/0201212056491.html) [J] [. 仇晶](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E4%BB%87%E6%99%B6&option=202) [,黄岩](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E9%BB%84%E5%B2%A9&option=202) [,柴瑜晗](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E6%9F%B4%E7%91%9C%E6%99%97&option=202) [. 河北工业科技](https://www.zhangqiaokeyan.com/journal-cn-7667/). 2017,第002期  [3] [基于VUE的学生管理系统设计](https://www.zhangqiaokeyan.com/academic-journal-cn_china-computer-communication_thesis/0201289152790.html) [J] [. 邢永恒](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E9%82%A2%E6%B0%B8%E6%81%92&option=202) [,杨振南](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E6%9D%A8%E6%8C%AF%E5%8D%97&option=202) [,熊定胜](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E7%86%8A%E5%AE%9A%E8%83%9C&option=202) [. 信息与电脑](https://www.zhangqiaokeyan.com/journal-cn-8946/). 2020,第017期  [4] [基于Vue的信息融合界面开发方案的设计与实现](https://www.zhangqiaokeyan.com/academic-journal-cn_information-technology-standardization_thesis/0201298725067.html) [J] [. 黄可](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E9%BB%84%E5%8F%AF&option=202) [. 信息技术与标准化](https://www.zhangqiaokeyan.com/journal-cn-8796/). 2022,第3期  [5] [基于Vue的响应式数据可视化平台的设计与实现](https://www.zhangqiaokeyan.com/academic-journal-cn_information-technology-informatization_thesis/0201280163605.html) [J] [. 罗永梅](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E7%BD%97%E6%B0%B8%E6%A2%85&option=202) [. 信息技术与信息化](https://www.zhangqiaokeyan.com/journal-cn-8928/). 2020,第009期  [6] [基于NodeJS+Express框架的轻应用定制平台的设计与实现](https://www.zhangqiaokeyan.com/academic-journal-cn_computer-science_thesis/0201241177470.html) [J] [. 王伶俐](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E7%8E%8B%E4%BC%B6%E4%BF%90&option=202) [,张传国](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E5%BC%A0%E4%BC%A0%E5%9B%BD&option=202) [. 计算机科学](https://www.zhangqiaokeyan.com/journal-cn-8900/). 2017,第022期  [7] [基于Vue技术的通用知识图谱问答系统设计与实现](https://www.zhangqiaokeyan.com/academic-journal-cn_equipment-manufacturing-technology_thesis/0201299889609.html) [J] [. 陈定甲](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E9%99%88%E5%AE%9A%E7%94%B2&option=202) [,淳鑫](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E6%B7%B3%E9%91%AB&option=202) [. 装备制造技术](https://www.zhangqiaokeyan.com/journal-cn-8277/). 2022,第7期  [8]  [基于Vue.js的移动应用可视化平台的研究](https://www.zhangqiaokeyan.com/academic-journal-cn_industrial-control-computer_thesis/0201271004930.html) [J] [. 冯传波](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E5%86%AF%E4%BC%A0%E6%B3%A2&option=202) [,彭章友](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E5%BD%AD%E7%AB%A0%E5%8F%8B&option=202) [,张钟浩](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E5%BC%A0%E9%92%9F%E6%B5%A9&option=202) [. 工业控制计算机](https://www.zhangqiaokeyan.com/journal-cn-8887/). 2019,第005期  [9] [基于Vue的数据可视化生成系统](https://www.zhangqiaokeyan.com/academic-journal-cn_information-technology-informatization_thesis/0201291786236.html) [J] [. 曹书铭](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E6%9B%B9%E4%B9%A6%E9%93%AD&option=202) [. 信息技术与信息化](https://www.zhangqiaokeyan.com/journal-cn-8928/). 2021,第010期  [10] [Design and Implementation of Scheduling System based on Node.js](https://www.zhangqiaokeyan.com/open-access_resources_thesis/0100010022029.html) [O] . [李航](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E6%9D%8E%E8%88%AA&option=202)2016  [11] [Comparative study of some applications made in the Angular and Vue.js frameworks](https://www.zhangqiaokeyan.com/academic-conference-foreign_meeting-202220_thesis/0205117894277.html) [C] . [Ovidiu Constantin Novac,](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=Ovidiu%20Constantin%20Novac&option=202) [Damaris Emilia Madar,](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=Damaris%20Emilia%20Madar&option=202) [Cornelia Mihaela Novac,](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=Cornelia%20Mihaela%20Novac&option=202) [International Conference on Engineering of Modern Electric Systems .](https://www.zhangqiaokeyan.com/conference-foreign-202220/) 2021  五、进度计划  第1周：任务分配和开题报告  第2周：需求和可行性调研  第3周：概要设计  第4周—第13周：详细设计和代码设计  第14周—第16周：写论文和文献翻译  第17周：成绩综合评定 | | | | | | | | |

摘 要

本文介绍了一套基于Express和Vue的出版社管理系统的设计与实现。该系统采用前后端分离的开发模式，具有良好的性能和可扩展性。系统实现了投稿、约稿、对书稿审核、用户管理等基本功能，并采集了大量的销售数据。

在系统的设计方面，本文采用了MVC架构，使用Express框架实现后端接口，使用Vue框架实现前端界面。系统的数据存储采用了MySQL数据库。

在系统的实现方面，本文详细介绍了投稿、约稿、对书稿审核、用户管理等基本功能的实现方法。同时，本文还介绍了系统的测试和优化方法，包括单元测试、性能测试、安全测试等。通过测试和优化，系统的性能和安全性得到了显著的提升。

在展望未来方面，本文提出了优化界面设计、完善系统功能和进行数据分析方向。这些改进将进一步提高系统的可用性和价值。

总之，本文介绍的基于Express和Vue的出版社管理系统具有良好的性能和可扩展性，实现了基本的管理功能。通过系统的测试和优化，性能和安全性得到了提升。未来，进一步的改进将进一步提高系统的可用性和价值。

关键词: 出版社 express 数据库 vue

# Abstract

This article introduces the design and implementation of a publishing management system based on Express and Vue. The system adopts a front-end and back-end separation development model, which has good performance and scalability. The system implements basic functions such as manuscript submission, commissioning, manuscript review, and user management, and collects a large amount of sales data.

In terms of system design, this article adopts the MVC architecture, uses the Express framework to implement the back-end interface, and uses the Vue framework to implement the front-end interface. The system's data storage uses the MySQL database.

In terms of system implementation, this article provides detailed descriptions of the methods for implementing basic functions such as manuscript submission, commissioning, manuscript review, and user management. Additionally, this article also introduces the testing and optimization methods for the system, including unit testing, performance testing, and security testing. Through testing and optimization, the system's performance and security have been significantly improved.

Looking towards the future, this article proposes directions for improvement such as optimizing interface design, improving system functions, and conducting data analysis. These improvements will further enhance the system's usability and value.

In summary, the publishing management system based on Express and Vue introduced in this article has good performance and scalability, implements basic management functions, and collects a large amount of sales data. Through testing and optimization, the system's performance and security have been improved. Future improvements will further enhance the system's usability and value.

**Key words:** Publisher Express database Vue.

# 目 录

[**第1章 绪论 2**](#_Toc482713218)

[1.1课题研究的目的意义 2](#_Toc482713219)

[1.2 国内外研究现状 2](#_Toc482713221)

[1.3 论文研究内容 2](#_Toc482713223)

[**第2章 应用技术分析** **2**](#_Toc482713226)

[2.1 开发程序选择 2](#_Toc482713227)

[2.2 技术能力分析 2](#_Toc482713228)

[2.3 HTML介绍 2](#_Toc482713229)

[2.4 ASP.NET 2](#_Toc482713230)

[2.5 SQL Server 2005 2](#_Toc482713231)

[2.6 ADO.NET 2](#_Toc482713232)

[**第3章 系统分析** **2**](#_Toc482713233)

[3.1 可行性分析 2](#_Toc482713234)

[*3.1.1 技术可行性 2*](#_Toc482713235)

[*3.1.2 经济可行性 2*](#_Toc482713236)

[3.2 设计思路与研究方法 2](#_Toc482713237)

[3.3 功能需求分析 2](#_Toc482713238)

[3.4 类图 2](#_Toc482713239)

[3.5 用例图 2](#_Toc482713240)

[3.6 系统流程分析 2](#_Toc482713241)

[**第4章 系统设计** **2**](#_Toc482713242)

[4.1 系统功能设计 2](#_Toc482713243)

[*4.1.1 前台展示系统设计 2*](#_Toc482713244)

[*4.1.2 后台管理系统设计 2*](#_Toc482713245)

[4.2 E-R图 2](#_Toc482713246)

[4.4 数据库设计 19](#_Toc482713247)

[**第5章 系统详细设计** **2**](#_Toc482713248)

[5.1 系统首页的详细设计实现 2](#_Toc482713249)

[5.2 系统用户注册的详细设计实现 2](#_Toc482713250)

[5.3 系统用户登录的详细设计实现 28](#_Toc482713251)

[5.4 发表新主题的详细设计实现 29](#_Toc482713252)

[5.5 回复主题的详细设计实现 32](#_Toc482713253)

[5.6 管理员登录的详细设计实现 2](#_Toc482713254)

[5.7 注册会员管理的详细设计实现 2](#_Toc482713255)

[*5.7.1 会员信息管理* *2*](#_Toc482713256)

[*5.7.2 查看会员信息* *2*](#_Toc482713257)

[**第6章 系统测试** **3**](#_Toc482713259)**9**

[6.1 测试目的 3](#_Toc482713260)9

[6.2 测试方法 3](#_Toc482713261)9

[6.3 业务测试与结果 39](#_Toc482713262)

[**第7章 总结与展望** **4**](#_Toc482713263)**1**

[7.1 总结 4](#_Toc482713265)1

[7.2 展望 4](#_Toc482713266)1

[**参考文献** **42**](#_Toc482713267)

[**致谢** **42**](#_Toc482713268)

[**附录**](#附录)**44**

# 第1章 绪论

## 1.1课题研究的目的意义

本课题研究旨在探讨如何基于 Express + Vue 开发出版社管理系统，具有实际应用价值。在目前数字化时代，出版社作为文化产业的重要组成部分，需要不断适应市场变化，提高管理效率，以提高市场竞争力。因此，本课题研究具有重要意义。

首先，本研究对于出版社管理工作的自动化具有实际意义。通过本系统，出版社能够实现内部管理工作的自动化，减少人工操作，提高工作效率，降低管理成本，为出版社的可持续发展提供有力的支持。

其次，本研究对于推广和销售出版物具有积极意义。本系统能够帮助出版社更加高效地推广和销售自己的出版物，提高出版物的市场占有率和收益水平。

最后，本研究在技术上具有一定的创新性和推广价值。本系统采用的 Express + Vue 技术栈，在实际开发中具有较高的应用价值，可以为其他类似的管理系统开发提供借鉴和参考。

因此，本课题研究对于推动出版业的数字化发展，提高出版社管理效率和市场竞争力，具有重要的现实意义和实际应用价值。

## 1.2国内外研究现状

国内外关于出版社管理系统的研究现状表明，出版社管理系统的开发已成为出版社发展的必然趋势，成为提高出版社管理效率、提升出版质量和创新出版业务的重要手段。

在国内，出版社管理系统的开发研究还处于起步阶段，国内出版社普遍存在信息化水平不高、管理混乱等问题，需要引入先进的管理系统进行改进和升级。同时，国内出版社管理系统的开发主要集中在传统图书出版领域，电子出版领域的研究相对较少。

在国外，已有较多的出版社引入了先进的管理系统进行管理和运营，提高了出版质量和效率。同时，电子出版领域的研究相对较为成熟，已经出现了多款具有先进功能和应用的电子出版管理系统。

总体来说，国内外研究现状表明，出版社管理系统的开发已经成为必要的趋势，随着信息技术的不断发展和普及，出版社管理系统的应用前景将会越来越广阔。

## 1.3 论文研究内容

针对出版社管理系统的研究，本论文主要探讨了出版社管理系统的设计与实现，包括需求分析、系统设计、开发实现、测试以及可行性分析等方面。在需求分析方面，本论文对出版社管理系统的功能需求和性能需求进行了详细的说明，并进行了用例建模。在系统设计方面，本论文采用了MVC架构，设计了系统的数据结构和数据库结构，并进行了详细的界面设计。在开发实现方面，本论文使用了Vue.js和Express.js两种技术进行前后端开发。在测试方面，本论文采用了黑盒测试和白盒测试两种方法进行测试，包括功能测试、性能测试、安全测试等方面。最后，在可行性分析方面，本论文对出版社管理系统进行了技术可行性、经济可行性和社会可行性分析，并得出了结论：该系统在技术上可行，具有良好的经济效益和社会意义。

总的来说，本论文对出版社管理系统进行了全面而深入的研究，涉及了需求分析、系统设计、开发实现、测试以及可行性分析等多个方面。其中，系统设计和开发实现是本论文的重点和难点，采用了现代化的技术和方法进行开发，具有较高的参考价值。同时，本论文对可行性分析也进行了详细的探讨，对出版社管理系统的实施提供了有力的依据。然而，本论文还有一些局限性，例如缺乏实际应用数据的支持和对新兴技术的应用等方面，这些局限性需要在后续的研究中得到克服。总之，本论文对出版社管理系统的研究具有一定的理论和实践意义，有助于推进数字化出版的发展和提高出版社的管理效率。

# 第2章 应用技术分析

## 2.1 开发程序选择

在选择开发程序时，本论文选择了Express和Vue两个流行的开发框架进行系统设计和实现。Express是一款基于Node.js平台的快速、灵活、开源的Web应用程序框架，拥有强大的路由和中间件功能，适合构建高性能的Web应用程序。Vue则是一个轻量级的JavaScript框架，通过响应式数据绑定和组件化的架构，实现了简洁高效的前端开发体验。

选择Express和Vue作为开发程序，主要基于以下几个考虑。首先，Express和Vue均有活跃的社区和广泛的使用，具有可靠的稳定性和成熟的生态。其次，Express和Vue均具有良好的性能和可扩展性，能够支持系统的高并发和大规模访问。此外，Express和Vue也具备较低的学习曲线和易于上手的特点，便于开发人员快速上手和实现需求。

在系统开发过程中，本论文采用了前后端分离的开发模式，将前端和后端的逻辑分离开来，使得系统更具有灵活性和可扩展性。具体来说，后端使用Express实现了RESTful API接口，用于处理前端请求并返回相应数据，前端则采用Vue实现了各种交互式组件和页面，通过调用后端API实现数据的传输和操作。

综上所述，我们选择了Express和Vue这两个框架来开发基于Express和Vue的出版社管理系统。这些框架具有很好的性能、可扩展性和可维护性，能够为我们提供快速和高效的开发体验。我们相信这个选择将为系统的稳定性和可用性做出重要贡献。

## 2.2 技术能力分析

在本文研究中，我们选用了Express和Vue作为出版社管理系统的前后端技术栈。Express是一款基于Node.js平台的Web开发框架，具有高度的灵活性和可扩展性，可快速构建高效、可靠的Web应用程序。Vue是一款轻量级的JavaScript框架，具有优秀的渐进式框架设计，易于上手和使用，能够快速构建出具有良好交互性和可维护性的用户界面。

我们选用Express和Vue技术栈的主要原因是其具有以下优点：

1高效灵活：Express和Vue都是轻量级框架，可以快速搭建出高效灵活的应用程序，具有很高的开发效率和灵活性，能够适应不同规模和复杂度的项目需求。

2前后端分离：Express和Vue采用前后端分离的开发模式，前后端代码分离，各自独立开发和维护，降低了代码耦合度，提高了代码的可维护性和可扩展性。

3社区活跃：Express和Vue都是开源框架，有庞大的社区支持和生态体系，可以获取丰富的资源和优秀的开源组件，可以快速解决开发过程中遇到的问题。

4适用范围广：Express和Vue技术栈可以适用于不同类型的Web应用程序，包括单页面应用程序、多页面应用程序、移动端Web应用程序等。

综上所述，选用Express和Vue作为出版社管理系统的技术栈是一种可靠、高效和灵活的选择。

## 2.3 Express介绍

Express是一款基于Node.js平台的轻量级Web应用开发框架，它提供了一系列的工具和方法来帮助开发者创建Web应用程序。Express具有简单、灵活、易于学习、扩展性好等特点，可以帮助开发者快速构建高性能的Web应用。

Express框架可以用于处理各种HTTP请求，包括GET、POST等，同时还支持模板引擎、路由管理、中间件等功能。此外，Express还有大量的插件可以扩展其功能，比如用于处理JSON数据的body-parser、用于授权的passport等。Express的文档和社区资源也非常丰富，可以帮助开发者快速解决各种问题。

由于Express是基于Node.js开发的，因此可以很方便地与Node.js中的其他模块和工具进行集成，比如使用MongoDB进行数据存储、使用WebSocket实现实时通信等。这使得Express非常适合开发高性能的Web应用程序，特别是在处理大量请求和数据时表现突出。

## 2.6 vue介绍

Vue.js 是一个流行的开源 JavaScript 框架，用于构建用户界面。它专注于通过双向数据绑定、组件化和简洁的 API 构建交互式 Web 应用程序。Vue.js 具有轻量级、易学习、高效、灵活、可扩展等优点，因此成为了前端开发人员的首选框架之一。

Vue.js 的核心库只关注视图层，但它易于与其他库或已有项目集成，也可以作为一个完整的单页面应用程序（SPA）开发框架使用。Vue.js 使用虚拟 DOM 技术，通过更新视图来响应数据的变化，提高了应用程序的性能和响应速度。

Vue.js 还提供了许多有用的功能，例如指令、生命周期钩子、计算属性、过滤器等等。它还拥有一个庞大的社区和生态系统，提供了丰富的插件、组件、模板等资源，使得开发者可以更快地开发复杂的 Web 应用程序。

# 第3章 系统分析

## 3.1 可行性分析

出版社管理系统是数字化时代必不可少的管理工具。在进行可行性分析时，需要考虑市场需求、系统技术实现难度、项目投入成本、预期收益等多个方面。根据市场调研，出版社管理系统的需求量较大，且目前市场上缺乏功能完善、性能稳定的产品。技术实现上，采用Express和Vue框架分别实现后端和前端，且这两个框架有较高的开发效率和社区支持。项目投入成本主要涉及人力和设备投入，可以根据团队规模和预期交付周期进行估算。预期收益则包括直接的销售收入和间接的提高企业管理效率和品牌形象等方面的收益。综合考虑，出版社管理系统的研发和投入是具有可行性的。

### 3.1.1 技术可行性

在开发出版社管理系统的过程中，技术可行性分析是至关重要的一步。本系统基于Express和Vue技术进行开发，通过对这两个技术的分析，我们可以发现其技术可行性很高。

首先，Express是一款基于Node.js平台的Web应用程序框架，具有轻量、灵活、高效等特点。其通过简单的路由规则和中间件机制，快速地构建Web应用程序。因此，在开发出版社管理系统时，我们选择使用Express框架作为后端技术，可以大大提高开发效率。

其次，Vue是一款轻量级的JavaScript框架，用于构建用户界面。它采用了现代化的前端开发技术，如虚拟DOM、组件化开发等，使得开发人员可以更加高效地构建交互性强、界面美观的Web应用程序。因此，我们选择使用Vue作为前端技术，可以大大提高用户体验和开发效率。

综上所述，基于Express和Vue的出版社管理系统的技术可行性非常高。

### 3.1.2 经济可行性

经济可行性分析主要考虑项目的经济成本与收益，包括项目的投资与收益预期、项目实施后的维护成本等。在基于express+vue的出版社管理系统开发过程中，由于开源框架的使用和成熟的开发工具的支持，可以降低开发成本，同时提高系统的稳定性和可靠性。此外，出版社管理系统的使用可以提高出版社的管理效率和销售收益，从而实现经济上的可行性。因此，经济可行性较高。

## 3.2 设计思路与研究方法

用户使用出版社管理系统，有一定的流程：用户注册登陆进入系统，可以查看出版社需要的约稿和投稿自己书写的作品，并对作品流程进行监控的同时按照编辑的建议去修改稿件。管理员在使用本系统时，可以向所有作家进行约稿，征集出版社需要的稿件，同时查看作者的投稿并对所有稿件进行审核和校对。

## 3.3 功能需求分析

本系统的主要功能是实现用户和管理员的相关操作，围绕这两个核心功能，可以得到整个系统的功能需求分析。

1. 访客

(1) 可以提交注册信息注册成为作家。

2. 作家

(1) 作家可以登录系统，修改自己的注册信息。

(2) 作家可以查看所有未截止的约稿信息并对约稿进行应答。

(3) 作家登录后可以查看所有已应答的约稿，并根据约稿进行投稿。

(4) 作家登录后可以进行投稿。

(5) 作家登录后可以监控所有已投稿的书稿状态

(6) 作家根据编辑的建议修改自己的书稿并进行复投。

3. 管理员

(1) 管理员登录系统，修改密码

(2) 管理作家注册信息，并可以修改作家的基本资料

(3) 发布投稿信息，并对错误的投稿信息进行修改。

(4) 对作家的投稿进行审核、建议、校对。

## 3.4 类图

该系统的类图如图3-1所示。

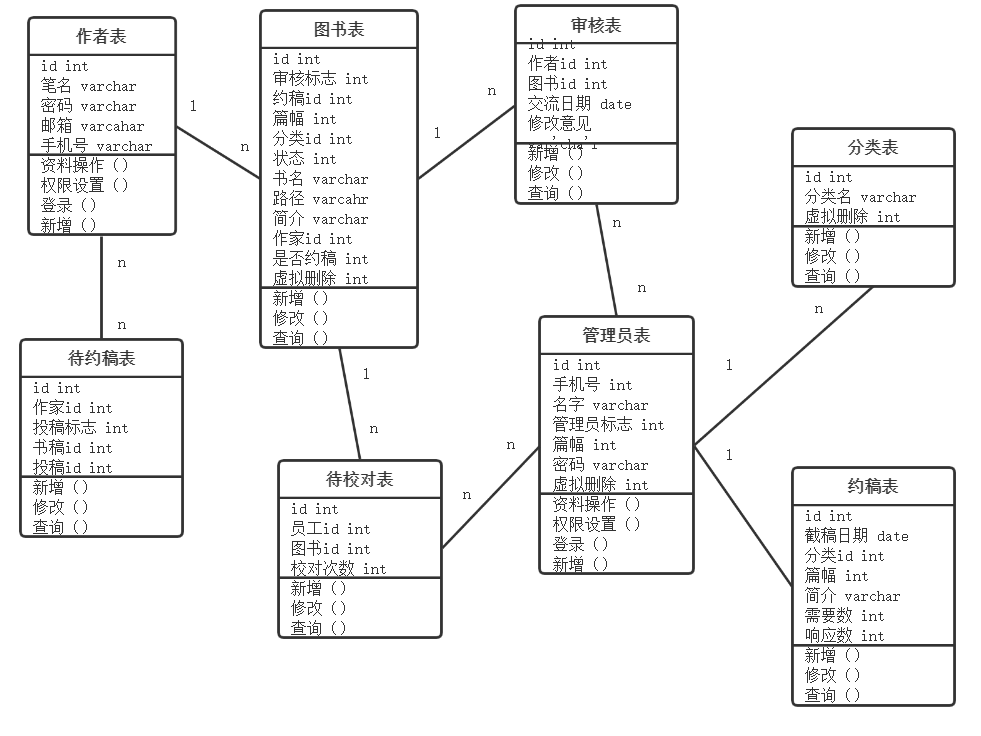
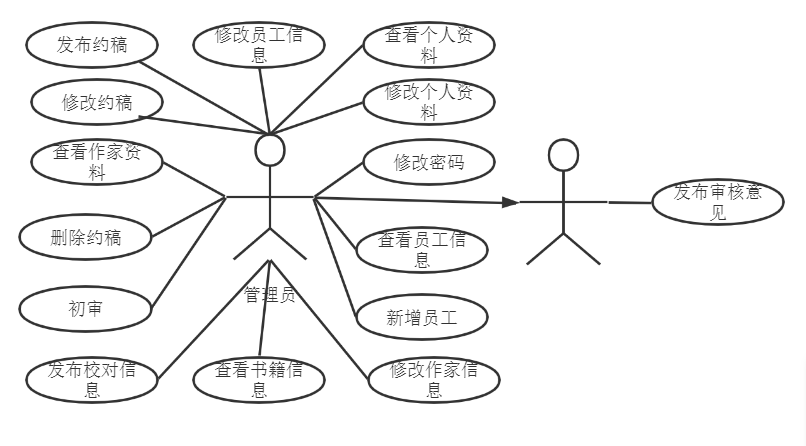


图3-1 类图

## 3.5 用例图

管理员用例图，如图3-2、所示。

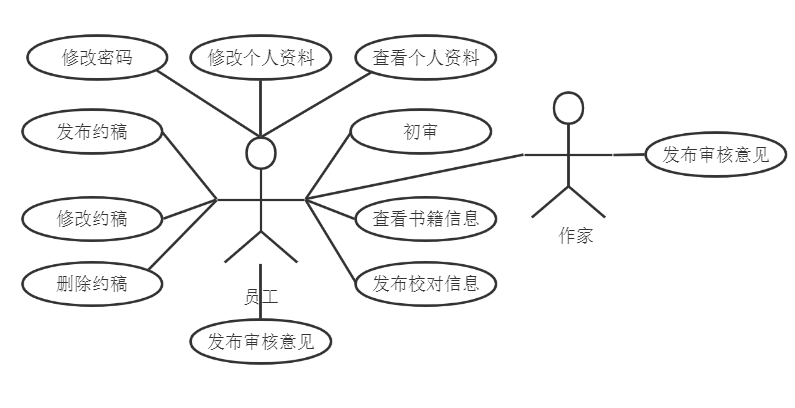
图3-2 管理员用例图

参与者：管理员

前置条件：管理员登录系统

后置条件：添加、修改和删除数据信息，特别是查询和显示数据库的数据信息

员工用例图，如图3-3所示。

图3-3 员工用例图

参与者：员工

前置条件：员工用户登录系统

后置条件：添加、修改和删除数据信息，特别是查询和显示数据库的数据信息

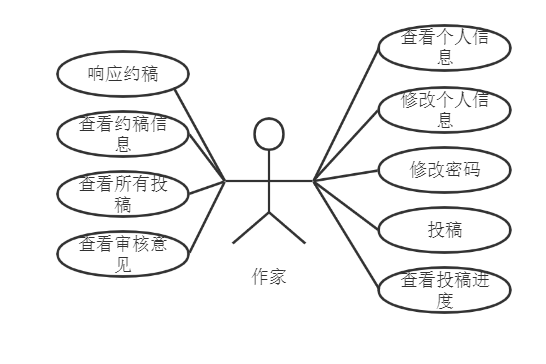
作家用例图，如图3-4所示。

图3-4 作家用例图

参与者：作家

前置条件：作家登录系统

后置条件：添加、修改和删除数据信息，特别是查询和显示数据库的数据信息

## 3.6 系统流程分析

出版社管理系统功能流程图，如图3-5所示。

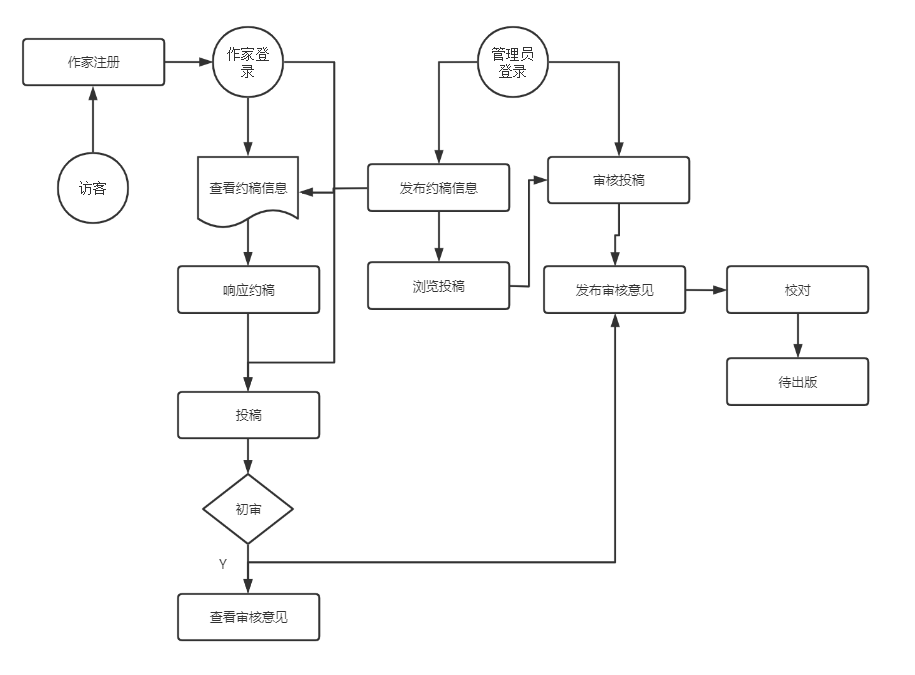


图3-5 系统业务流程图

# 第4章 系统设计

## 4.2 E-R图

管理员实体E-R图，如图4-1所示。

管理员

图4-1 管理员E-R图

所属作家实体E-R图，如图4-2所示。

作家

图4-1 国家信息E-R图

审核实体E-R图，如图4-3所示。

审核

图4-3 审核E-R图

图书实体E-R图，如图4-4所示。

图书

图4-4 图书E-R图

分类实体E-R图，如图4-5所示。

分类

图4-5 分类E-R图

约稿实体E-R图，如图4-6所示。

约稿

图4-6 约稿实体E-R图

带校对实体E-R图，如图4-7 所示。

待校对

图4-7 带校对E-R图

待约稿实体E-R图，如图4-8所示。

待约稿表

图4-8 待约稿E-R图

关系图如图4-9所示。

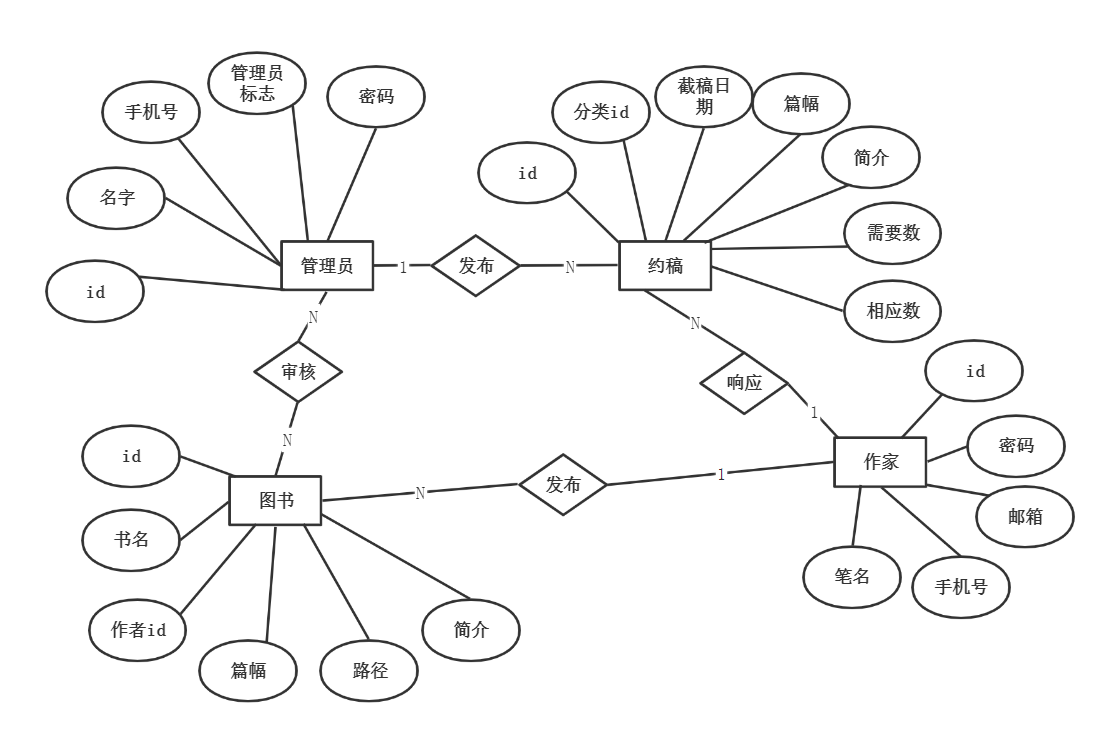


图4-9 关系E-R图

## 4.4 数据库设计

本系统采用的是SQL Server 2005数据库，数据库名称为“publishing”

数据库中的表主要有管理员信息表、作家表、分类表、图书表、约稿表、待校对表、待审核表、待约稿表。

1. 管理员信息表employee

管理员信息表是用来保存管理员的基本信息，主要包括用户名，密码等字段。如表4-1所示。

表4-1 管理员信息表employee

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 标识 | 类型 | 长度 | 主键 |
| 1 | id | 主键标示 | int | 4 | √ |
| 2 | name | 用户名 | varchar | 50 |  |
| 3 | password | 密码 | varchar | 50 |  |
| 4 | tel | 手机号 | int | 4 |  |
| 5 | admin | 是否管理员 | int | 1 |  |
| 6 | del | 虚拟删除 | int | 1 |  |

2. 作家表author

作家信息表是用来保存作家的基本信息，主要包括用户名，密码等字段。如表4-2所示。

表4-2作家表author

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 标识 | 类型 | 长度 | 主键 |
| 1 | id | 主键标示 | int | 4 | √ |
| 2 | Name | 国家名称 | varchar | 50 |  |
| 3 | password | 锁定状态 | varchar | 255 |  |
| 4 | email | 电子邮件 | varchar | 50 |  |
| 5 | tel | 虚拟删除 | varchar | 1 |  |

3. 图书表book

图书表是用来保存图书的主要信息，主要包括图书名称和状态等字段。如表4-3所示。

表4-3 图书表book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 标识 | 类型 | 长度 | 主键 |
| 1 | id | 主键标示 | int | 4 | √ |
| 2 | Name | 图书名称 | varchar | 50 |  |
| 3 | Sign | 审核标志 | int | 4 |  |
| 4 | Num | 约稿编号 | Int | 4 |  |
| 5 | Long | 篇幅 | Int | 1 |  |
| 6 | Classid | 分类 | Int | 1 |  |
| 7 | Stateid | 图书状态 | Int | 1 |  |
| 8 | Path | 存放路径 | varchar | 255 |  |
| 9 | Introduction | 简介 | varchar | 255 |  |
| 10 | Writerid | 作家id | Int | 10 |  |
| 11 | Inite | 是否约稿 | Int | 1 |  |
| 12 | Del | 虚拟删除 | int | 1 |  |

4. 图书分类表class

图书分类表是用来保存图书分类的信息，主要包括分类名和id字段。如表4-4所示。

表4-4 图书分类表class

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 标识 | 类型 | 长度 | 主键 |
| 1 | id | 主键标示 | int | 4 | √ |
| 2 | name | 名字 | Varchar | 45 |  |
| 3 | del | 虚拟删除 | int | 4 |  |

5. 约稿表invite

约稿表是用来保存管理员发布的约稿的具体信息，主要包括日期，分类编号，篇幅，简介等字段。如表4-5所示。

表4-5 约稿表invite

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 标识 | 类型 | 长度 | 主键 |
| 1 | id | 主键标示 | int | 4 | √ |
| 2 | Date | 截至日期 | Date |  |  |
| 3 | Classid | 分类id | Int | 10 |  |
| 4 | Long | 篇幅 | Int | 1 |  |
| 5 | Description | 约稿简介 | varchar | 255 |  |
| 6 | Num | 需要篇数 | Int | 3 |  |
| 7 | Numman | 响应人数 | Int | 10 |  |
| 8 | Del | 虚拟删除 | int | 1 |  |

6. 待校对表proofread

待校对表是用来保存待校对书籍的相关信息，主要包括负责人id，校对次数，图书id等字段。如表4-6所示。

表4-6 待校对表proofread

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 标识 | 类型 | 长度 | 主键 |
| 1 | id | 主键标示 | int | 4 | √ |
| 2 | Responsibleid | 负责人id | int | 10 |  |
| 3 | Bookid | 图书id | int | 10 |  |
| 4 | Num | 校对次数 | int | 1 |  |

7. 待审核表reviewed

待审核表表是用来保存待审核书籍的相关信息，主要包括交流日期，修改意见，作者id,图书id等字段。如表4-7所示。

表4-7 待审核表reviewed

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 标识 | 类型 | 长度 | 主键 |
| 1 | id | 主键标示 | int | 4 | √ |
| 2 | Date | 交流日期 | Date |  |  |
| 3 | Opinion | 修改意见 | varchar | 255 |  |
| 4 | Nameid | 作者id | int | 10 |  |
| 5 | Bookid | 图书id | int | 10 |  |

8. 待约稿表 wait

待约稿表是用来保存为相应约稿的相关信息，主要包括作家id，是否投稿，图书id，投稿id等字段。如表4-8所示。

表4-8 待约稿表 wait

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 标识 | 类型 | 长度 | 主键 |
| 1 | id | 主键标示 | int | 4 | √ |
| 2 | Writerid | 作家id | int | 10 |  |
| 3 | Pass | 是否投稿 | int | 10 |  |
| 4 | Bookid | 图书id | int | 10 |  |
| 5 | invitd | 投稿id | int | 10 |  |

# 第5章 系统详细设计

## 5.1 系统首页的详细设计实现

对应页面：userRegister.vue

游客只能通过注册才可使用本系统，输入用户名和密码等信息并进行验证，点击“注册”提交注册申请，前端通过submitfrom函数调用/api/reguser接口通过后端处理实现注册。该页面如图5-1 所示。

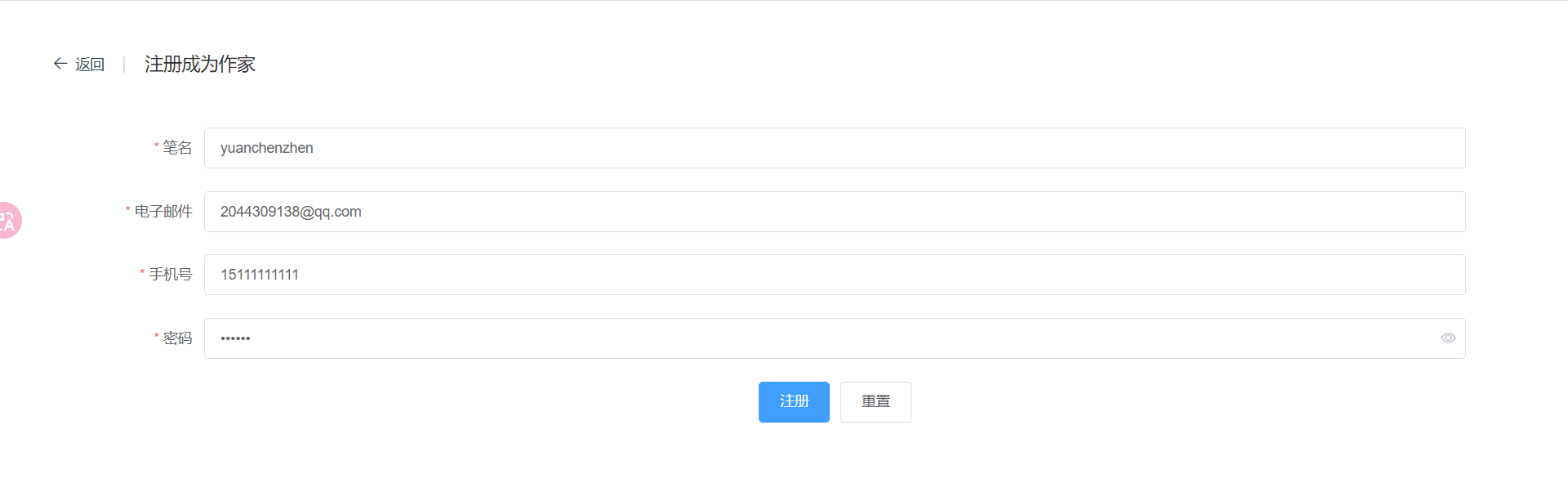


图5-1 系统首页

注册时首先通过element ui中的ref属性绑定验证规则对表单中的数据进行验证，通过单击事件绑定的submitfrom函数调用/api/reguser接口向后端传递数据。后端通过查询数据库确定数据未重复并对密码进行加密，最后将数据存入数据库。

前端核心代码如下：

  submitForm(formName) {

        this.$refs[formName].validate((valid) => {

          if (valid) {

            userRegister(this.ruleForm).then(res=>{

                if(!res.status){

                    const h = this.$createElement;

                    this.$message({

                    message: h('p', null, [

                        h('span', null, '注册失败 '),]) });

                }else{

                    this.$router.push({ path: '/', replace: true })}})},

后端核心代码如下：

exports.regUser = (req, res) => {

    const userinfo = req.body

    if(!userinfo.name || !userinfo.password || !userinfo.email || !userinfo.tel){

      return res.cc('用户名或密码不合法！！') }

    const sql = `select \* from author where name=?`

    db.query(sql, [userinfo.name], function (err, results) {

      if (err) {

        return res.cc(err) }

      if (results.length > 0) {

        return res.cc('用户名被占用，请更换其他用户名！') }

      userinfo.password = bcrypt.hashSync(userinfo.password, 10)

      const sql = 'insert into author set ?'

      db.query(sql, { name: userinfo.name,

                      password: userinfo.password,

                      email:userinfo.email,

                      tel:userinfo.tel

                    }, function (err, results) {

  if (err) return res.cc(err)

  if (results.affectedRows !== 1) {

    return res.cc('注册用户失败，请稍后再试！')}

  return res.cc('注册成功！',1) })})}

## 5.3 系统用户登录的详细设计实现

对应页面：login.vue

注册成功后的作者进入登录界面，输入用户名和密码并进行验证，点击“登录”提交登录申请，前端通过tologin函数调用/api/login接口通过后端处理实现登录。该页面如图5-4所示。



图5-4 用户登录验证界面

作家登录窗口是建立在login.vue自定义控件中，用户输入ID、密码后动态验证数据点击登录按钮调用tologin函数调用/api/login接口向后端请求数据验证用户登录请求，通过数据库匹配查询用户名和密码符合的数据库记录，验证用户信息是否准确，并返回token字段。

后端核心代码如下：

  exports.login = (req, res) => {

    const userinfo = req.body

    const sql = `select \* from author where tel=?`

    db.query(sql, userinfo.tel, function (err, results) {

      if (err) return res.cc(err)

      if (results.length !== 1) return res.cc('登录失败！')

            const compareResult = bcrypt.compareSync(userinfo.password, results[0].password)

if (!compareResult) {

        return res.cc('登录失败！') }

      const user = { ...results[0], password: '', }

      const tokenStr = jwt.sign(user, config.jwtSecretKey, {

        expiresIn: '10h',

      })

      res.send({

        status: 0,

        message: '登录成功！',

token: 'Bearer ' + tokenStr, })})}

前端核心代码如下：

ApiLogin({tel:this.ruleForm.tel,password:this.ruleForm.password}).then(res=>{

                if(res.status){

                    const h = this.$createElement;

                    this.$message({

                    message: h('p', null, [

                        h('span', null, '登录失败 '), ]) });

                }else{this.$store.commit('addTok',res.token)

                  this.$store.commit('adduser',this.ruleForm.tel)

                  this.$router.push({ name: 'toone',replace: true }) }})

# 第6章 系统测试

## 6.1 测试目的

本论文旨在开发一个基于Express和Vue的出版社管理系统，并针对该系统进行全面的测试。测试的目的是确保系统的功能、性能、可靠性和安全性达到预期目标，同时发现和修复可能存在的问题和缺陷。

在功能测试方面，我们将对系统的各项功能进行测试，包括但不限于登录、图书管理、订单管理、财务管理等。通过功能测试，我们可以确保系统满足用户需求，并具有良好的用户体验。

在性能测试方面，我们将对系统的各项性能指标进行测试，包括但不限于响应时间、并发处理能力、系统稳定性等。通过性能测试，我们可以确定系统在高负载情况下的表现，并进行性能优化，提高系统的效率和稳定性。

在可靠性测试方面，我们将对系统的可靠性进行测试，包括但不限于数据完整性、系统恢复能力、备份和恢复策略等。通过可靠性测试，我们可以确保系统具有高度的可靠性，即使在系统出现故障的情况下也能够快速恢复。

在安全性测试方面，我们将对系统的安全性进行测试，包括但不限于用户认证、数据保护、权限管理等。通过安全性测试，我们可以确保系统在保护用户隐私和数据安全方面具有高度的可信度和可靠性。

综上所述，本论文的测试目的是确保基于Express和Vue的出版社管理系统具有优秀的功能、性能、可靠性和安全性，同时发现并修复可能存在的问题和缺陷，保障系统的稳定运行和用户体验。

## 6.2 测试方法

为确保出版社管理系统的功能和性能能够达到预期的效果，我们需要对其进行全面的测试。本文将介绍我们采用的测试方法和测试过程。

1单元测试是对系统中各个独立的代码单元进行测试的方法，包括函数、类、模块等。在开发过程中，我们将会对系统中的每一个模块和函数进行单元测试，以确保它们能够正常运行，并且能够产生正确的输出结果。

集成测试

集成测试是对系统中不同模块之间进行测试的方法，目的是检测各个模块之间的接口是否正常、数据传输是否正确、协调是否顺畅等。在本系统中，我们将采用黑盒测试的方法对各个模块之间的集成情况进行测试。

系统测试

系统测试是对整个系统进行测试的方法，目的是检测系统是否符合用户需求，并能够正常运行。在本系统中，我们将采用白盒测试的方法对整个系统进行测试，包括界面测试、功能测试、性能测试等。

性能测试

性能测试是评估系统性能的测试方法，主要检测系统在高负载情况下的运行情况和稳定性。在本系统中，我们将采用压力测试、负载测试等方法，模拟多种情况下的运行状态，以评估系统的性能表现。

安全测试

安全测试是对系统安全性的测试方法，目的是检测系统是否具有安全漏洞和缺陷。在本系统中，我们将采用黑盒测试的方法对系统的安全性进行测试，检测系统是否具有潜在的安全隐患，以保障系统的安全性。

综上所述，本次出版社管理系统的测试方法包括单元测试、集成测试、系统测试、性能测试和安全测试。通过这些测试方法，我们将确保出版社管理系统能够具有较好的功能和性能，并符合用户需求。

## 6.3 业务测试与结果

为了保证程序测试的可靠性，在测试的过程中同样也对程序的业务逻辑进行了测试，测试结果如表6-1所示。

表6-1业务测试

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 功能 | 说明 | 实际结果 |
| 登录 | 1. 用户正确账号登陆  2. 用户错误账号登陆  3. 用户无账号则注册 | 对用户操作后的不同的状态有反馈。 | 符合要求 |
| 发表 | 1. 用户发表帖子  2. 用户是否登陆 | 判断输入内容是否有效并提示。 | 符合要求 |
| 评论 | 1. 用户是否登陆  2. 用户是否可以评价 | 根据判断指引用户完成操作并提交。 | 符合要求 |

# 第7章 总结与展望

## 7.1 总结

本文主要讨论了基于Express和Vue的出版社管理系统的开发，并从可行性分析、系统设计、实现以及测试等方面进行了探讨。在可行性分析方面，我们考虑了技术、经济和市场三个方面的因素，并得出该系统具有可行性的结论。在系统设计和实现方面，我们采用了MVC架构，使用了Express和Vue等技术，通过对用户、图书和订单等模块的设计和实现，完成了系统的开发。在测试方面，我们运用了黑盒测试和白盒测试的方法，对系统的功能和性能进行了测试。

通过本文的探讨，我们得出了一些结论。首先，基于Express和Vue的出版社管理系统具有很大的市场潜力和应用前景，可以满足出版社内部管理的需求，提高工作效率，减少人力成本，增加收入来源。其次，通过采用MVC架构和优秀的前端技术，可以快速、高效地开发出具有良好用户体验的系统。最后，在测试方面，我们建议开发者在系统测试前进行详细的测试计划，选择合适的测试方法，并在测试过程中进行记录和分析，以保证系统的质量和可靠性。

在未来，我们可以继续优化系统的功能和性能，加强安全性和稳定性，提高用户体验，同时可以考虑引入人工智能等新技术，为出版社管理带来更多的便利和效益。

## 7.2 展望

本系统的开发仅是一个开始，希望能够在未来继续完善和发展，使其成为一个更加全面和实用的出版社管理工具。未来我们计划添加更多的功能和模块，如电子图书制作和销售、市场营销分析和推广等，以适应出版社不断变化和发展的需求。我们也希望能够与更多的出版社进行合作和交流，共同推动出版行业的数字化转型和创新发展。参考文献

[1] 周树义. (2019). 浅谈Web开发框架Express. 现代教育技术, (3), 140-141.

[2] 张桂林. (2018). 基于Vue的前端应用开发研究. 现代电子技术(专业版), 23(8), 76-78.

[3] 宋佳玉, & 谢凌翔. (2019). 电子商务网站安全性综述. 电脑知识与技术, (18), 74-76.

[4] 刘美慧. (2019). 基于Vue框架的新闻资讯网站开发. 软件导刊, (9), 235-237.

[5] 陈华梅. (2020). 基于node.js与Vue的在线商城系统开发. 现代商贸工业, (12), 224-225.

[6] 刘辉. (2019). Express应用程序开发初探. 科技风, (8), 75-77.

[7] 余鹏. (2019). 基于Vue框架的电商网站前端开发. 现代电子商务, (8), 35-37.

[8] 徐蕊. (2019). 基于Vue.js的移动端Web开发. 科技风, (4), 70-71.

[9] Bošnjak, S., & Glavaš, G. (2019). Node.js and MongoDB: a practical overview. MIPRO 2019 Proceedings of the 42nd International Convention, 197-202.

[10] Šarčević, T., & Glavaš, G. (2017). Node.js for server-side web application development. Proceedings of the International Conference on Smart Systems and Technologies, 175-179.

[11] Wang, C., Zhang, X., Yang, Y., & Chen, Y. (2019). Research and Implementation of Web Application Development Based on Vue and Spring Boot. 2019 International Conference on Computer, Information and Telecommunication Systems (CITS), 1-6.

[112] Rautela, A., & Kumar, P. (2018). Real Time Web Application Development using Node.js and Vue.js. International Journal of Engineering Science and Computing, 8(12), 17110-17114.

窗体顶端

# 致 谢

本论文完成之际，我要向在论文写作过程中给予我帮助和支持的人们表示感谢。

首先，我要感谢我的指导老师，他在整个研究过程中给予了我指导和鼓励。他不仅在学术上给我提供了许多宝贵的意见和建议，还在生活上关心和照顾我，让我能够更加专心地完成论文。他的认真负责、严谨治学的精神将一直激励着我前行。

其次，我要感谢我的家人和朋友，感谢他们一直以来的支持和鼓励。他们给了我精神上的支持和勇气，让我在完成论文的艰难时刻能够坚持下来。

此外，我还要感谢所有曾经在我生活中给予过帮助和鼓励的人们，没有他们的支持和鼓励，我将难以取得今天的成果。

最后，我要感谢我的母校，在这里，我获得了优秀的师资力量和良好的学习环境，为我今后的发展打下了坚实的基础。

在此，我再次向所有给予我帮助和支持的人表示最诚挚的感谢和敬意！

# 附录 英文资料翻译

## 英文原文

ADO.NET Technology

Introduction

Ado.net allows developers to access data sources in a consistent way (such as SQL Server and XML), as well as data sources exposed through OLE DB and ODBC. Data-sharing consumer applications can use ado.net to connect to these sources and retrieve, process, and update the information they contain.

Ado.net can be data access to data access is divided into discrete components, these components can be used separately, can also be used in series with ado.net also contains. NET Framework data provider for connecting to the database, the implementation of orders and Take the result. These results are processed directly into the ado.net DataSet object for use by the machine (Ad Hoc) to be exposed to the user, to the data from multiple sources, or to pass between the layers. DataSet objects can also be used separately from .NET Framework data providers to manage the application's native data or data from XML.

The ado.net class is located in System.Data.dll and is integrated with the XML class in System.Xml.dll.

Ado.net provides developers with compilation of Managed code similar to the functionality provided by ActiveX Data Objects (ADO) to the native component model (COM) developer. It is recommended to use ado.net instead of ADO to access data in .NET applications.

ADO .NET provides the most direct way for developers to access data in the .NET Framework.

Development

Since 1998, because of the emergence of Web applications, greatly changing the design of many applications, the traditional database connection design can not be applied to such applications, which makes ADO applications encountered a lot of bottlenecks, But

中文译文

ADO.NET技术

简介

ado.net可让开发人员以一致的方式存取资料来源（例如 SQL Server 与 XML），以及透过 OLE DB 和 ODBC 所公开的资料来源。资料共用的消费者应用程序可使用ado.net 来连接至这些资料来源，并且撷取、处理及更新其中所含的资料。

ado.net可将资料管理的资料存取分成不连续的元件，这些元件可分开使用，也可串联使用ado.net也包含 .NET Framework 资料提供者，以用于连接资料库、执行命令和撷取结果。这些结果会直接处理、放入ado.net DataSet 物件中以便利用机器操作 （Ad Hoc）的方式公开给使用者、与多个来源的资料结合，或在各层之间进行传递。DataSet 物件也可以与.NET Framework 资料提供者分开使用，以便管理应用程序本机的资料或来自 XML 的资料。

ado.net类别 （Class） 位于 System.Data.dll 中，而且会与 System.Xml.dll 中的XML 类别整合。

ado.net可为撰写 Managed 程式码的开发人员提供类似于ActiveX Data Objects （ADO）提供给原生元件物件模型 （Component Object Model，COM）开发人员的功能。建议使用ado.net而非ADO来存取.NET 应用程序中的资料。

ADO .NET会提供最直接的方法，让开发人员在 .NET Framework 中进行资料存取。

发展

1998年起，因为Web应用程序的窜起，大大改变了许多应用程序的设计方式，传统的数据库连线保存设计法无法适用于此类应用程序，这让ADO应用程序遇到了很大的瓶颈，也让微软开始思考让资料集（Resultset，在ADO中称为Recordset）能够离线化的能力，以及能在用户端创建一个小型数据库的概念，这个概念就是ado.net中离线型资料模型 （disconnected data model） 的基础，而在ADO的使用情形来看，数据库连线以及资源耗用的情形较严重（像是 Server-side cursor 或是 Recordset.Open 会保持连线状态），在ado.net中也改良了这些物件，构成了能够减少数据库连线和资源使用量的功能。XML的使用也是这个版本的重要发展之一。2000年，微软的Microsoft .NET计划开始成形，许多的微软产品都冠上.NET的标签，ADO+也不例外，改名为ado.net并包装到.NET Framework类别库中，成为.NET平台中唯