

**晋城职业技术学院**

**专科毕业论文**

**论文题目： 图书管理系统的设计与实现**

**系  别： 信息工程系**

**专  业： 软件与信息服务**

**学  号： 1801293123**

**姓  名： 卫婷婷**

**指导老师： 陈志坚**

**中国 山西 晋城**

2020年6月

摘 要

图书管理系统的建设主要是为了使用户方便的借阅图书馆图书，使管理员快速、简单的对图书馆图书及用户信息进行管理，实现在线的借阅与图书、用户管理，充分利用网络的方便快捷、时时互动的特点，打破传统的手工操作模式，提出新颖的模式，使用户在网络模式下利用网络资源进行方便、快捷的图书借阅、利用虚拟情境与管理员进行交流。系统主要实现的功能有图书信息查询、添加管理，图书借阅记录、添加管理，用户列表、资料管理，数据统计等，其中借阅管理为重点，包括图书借阅记录管理，借阅记录查询等。

根据本系统的研究现状和发展趋势，系统从需求分析、结构设计、数据库设计，在到系统实现，分别为前端实现和后端实现。论文内容从系统描述、系统分析、系统设计、系统实现、系统测试来阐述系统的开发过程。本系统力求结合实际找出一种切实可行的开发方案，经过反复研究和学习，借助JS编程语言、HTML/CSS前端技术、VUE框架，

Element-UI框，和后端Mysql数据库和nodejs后端语言来完成系统的所有功能，最后进行系统测试，来检测系统的权限和漏洞，从而将系统完善，达到符合标准。

**关键字：** 图书管理;js编程语言；nodejs后端编程语言；Mysql数据库

目 录

[第1章 绪论 1](#_Toc16847)

[第2章 系统分析 2](#_Toc23608)

[2.1可行性分析 2](#_Toc31676)

[2.1.1技术可行性 2](#_Toc8675)

[2.1.2经济可行性 2](#_Toc9140)

[2.1.3操作可行性 2](#_Toc27019)

[2.1功能需求分析 2](#_Toc1012)

[2.2业务流程分析 3](#_Toc23840)

[2.3数据流程分析 4](#_Toc21460)

[2.4本章小结 4](#_Toc14017)

[第3章 系统设计 5](#_Toc4046)

[3.1系统设计思想 5](#_Toc14937)

[系统总体设计 5](#_Toc10261)

[3.3系统功能模块设计 5](#_Toc11128)

[3.4系统安全设计 6](#_Toc15737)

[3.5表单字段校验处理设计 6](#_Toc27947)

[3.6系统维护设计 7](#_Toc15973)

[3.7数据库设计 7](#_Toc13688)

[3.7.1数据库设计概述 7](#_Toc11496)

[3.7.2概念模型设计 7](#_Toc6608)

[3.7.3数据库表设计 8](#_Toc24035)

[3.7.4数据库连接计 9](#_Toc9203)

[3.8本章小结 10](#_Toc17112)

[第4章 系统的实现 11](#_Toc23324)

[4.1主页面的实现 11](#_Toc6887)

[4.3登录模块的实现 11](#_Toc19315)

[4.4借阅模块的实现 15](#_Toc3323)

[4.5数据统计模块的实现 17](#_Toc9742)

[4.7图书管理模块的实现 17](#_Toc31810)

[4.8本章小结 19](#_Toc9744)

[第5章 系统测试 20](#_Toc12630)

[5.1测试目的 20](#_Toc2981)

[5.2界面测试 20](#_Toc1479)

[5.3功能测试 20](#_Toc17382)

[5.4测试结果 21](#_Toc27879)

[5.5本章小结 21](#_Toc19700)

[第6章 总结 22](#_Toc19466)

[致 谢 23](#_Toc8134)

[参考文献 24](#_Toc25996)

# **第1章 绪论**

当今时代是飞速发展的信息时代。在各行各业中离不开信息处理这正是计算机被广泛应用于信息管理系统的环境。计算机的最大好处在于利用它能够进行信息管理。使用计算机进行信息控制，不仅提高了工作效率，而且大大提高了其安全性。尤其对于复杂的信息管理，计算机能够充分发挥它的优越性。计算机进行信息管理与信息管理系统的开发密切相关，系统的开发是系统管理的前提。本系统就是为了管理好图书馆而设计的。

图书馆管理需要的人员复杂，信息量复杂庞大，如果还使用手工操作处理图书借阅问题，工作将非常繁琐，需要大量的人力、物理、财力，极大的浪费了资源，对于图书馆管理人员来说，图书馆管理包括图书信息管理、读者管理、借阅信息管理、归还管理、管理员信息管理等等。而这些项目在过去靠手工操作，需要手工记录这些事情，不但麻烦，还经常出错，给广大用户带来很多不便，因此，开发这样一个在线网页图书管理系统。让管理员方便的管理图书及用户信息，方便用户查找图书。

# **第2章 系统分析**

## **2.1可行性分析**

本次毕业设计基于windows系统下，运用JS、Nodejs后端技术采用的是Mysql数据库和Visualstudio实现，总体的可行性共分为以下三个方面。

2.1.1技术可行性

### 本系统针对图书馆的工作特点,设计了用户管理、图书管理、借阅管理、数据统计等4个子系统。应用本系统可以在计算机上灵活、方便地管理图书,从而大大的提高了处理效率,使管理更加现代化。本系统是根据实际情况和具体内容,按一定的要求、科学、合理进行系统分析,设计,具体包括菜单设计、数据输入、查询、删除、修改、打印等各种设计。

### 2.1.2经济可行性

### 图书管理系统采用的插件都是开源免费的，开发工具，visualstudio，以及数据库是Mysql皆是开源免费的，后端采用基于一个保持最小规模的灵活的Node.js Web 应用程序开发框架Express，为 Web 和移动应用程序提供一组强大的功能，因为都是开源免费的，所以在开发前期，开发时用于项目的经费将会大大降低，不会让开发该软件在项目启动期受到经费的影响，所以经济上还是可行的。尽量用最少的花费去满足用户的需求。省下经费用于人工费，以及设备费用。将在无纸化，高效率的道路上越走越远。

### 2.1.3操作可行性

本系统实现功能的操作很简单，普通电脑的常见配置就可以运行本软件，并且只要粗通电脑使用的基本常识就可以流畅的使用本软件。电脑具备连接互联网的能力，并且可以正常访问系统，并不需要操作者有什么高超的能力，只需了解业务流程，并且按照专业知识进行正确操作即可，所以图书管理系统具备操作可行性。

## **2.1功能需求分析**

根据调查得知用户的需求，从而对系统的功能进行分析，系统应该包括了用户列表管理、用户资料管理、图书信息查询、图书信息添加、借阅记录管理、借阅记录添加、用户数据统计等功能模块。

表3-1 功能需求列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 功能  名称 | 功能描述 | 输入内容 | 输出内容 |
|  | 用户  登录 | 保证用户通过身份验证进入系统进行操作 | 用户名、密码 | 用户登录是否成功和用户登录状态 |
|  | 注册  账户 | 新用户根据个人信息注册个人账户 | 姓名、密码、班级、身份 | 注册是否成功的状态 |
|  | 修改个人信息 | 用户可以根据自己当前的情况修改个人的信息 | 要修改的信息 | 提示修改的结果 |
|  | 添加借阅记录 | 用户添加借阅记录 | 借阅记录的相关信息 | 借阅记录列表 |
|  | 编辑借阅记录 | 用户修改借阅记录信息 | 借阅记录相关信息 | 是否编辑成功 |
|  | 删除借阅记录 | 用户删除借阅记录 | 借阅记录信息 | 借阅记录删除成功或失败 |
|  | 添加还书记录 | 用户添加还书记录 | 还书记录的相关信息 | 还书记录列表 |
|  | 编辑还书记录 | 用户修改还书记录信息 | 还书记录相关信息 | 是否编辑成功 |
|  | 删除还书记录 | 用户删除还书记录 | 还书记录信息 | 还书记录删除成功或失败 |
|  | 图书信息查询 | 查询馆内图书列表以及相关信息 | 图书ISBN/图书名称 | 图书列表以及相关信息 |
|  | 图书信息添加 | 书籍相关信息/作者信息 | 被添加图书的相关信息 | 是否添加成功 |
|  | 删除图书信息 | 管理员删除图书信息 | 图书信息 | 图书信息删除成功或失败 |

## **2.2业务流程分析**

总体业务流程：以用户的身份在登录页面输入账号和密码，经过数据库身份验证，验证成功后登录系统主页，可以使用读者资料管理、图书信息添加、查询、图书借阅排行查询、图书借阅、还书添加查询等功能操作，以管理员的身份在登录页面输入账号和密码，经过数据库身份验证，验证成功后登录系统主页，可以使用系统用户管理、图书管理、借阅管理、数据统计管理等功能操作。

## **2.3数据流程分析**

本系统根据上节所设计的各个业务流程图，采用逐层细化的方法，画的每一部分各层的数据流图如下：

数据流图

普通用户和管理员都可以通过登录系统的界面来进入系统，如下图所示：



图3-4 0层数据流图

系统根据登录的数据，判断是该用户是哪种角色，仍然后跳转至功能页面，系统用户在系统内进行数据操作，此时数据流的有效数据流流向数据库中心执行相应的数据SQL语句，反馈结果到显示页面上。

管理员操作后台数据流图，管理员可以分别通过添加、修改和删除来对系统进行管理。数据流实体主要是管理员，数据流程包括图书类别管理和图书管理；数据流包括添加信息、删除信息、修改信息、查询信息、浏览信息、提示信息；数据表包括用户表、图书表、借阅表。

## **2.4本章小结**

本章主要论述了对用户的需求调研，系统业务功能，用例分析，系统业务流程分析、数据流程分析，其中数据的组成包括前端和后端的详细数据。

# **第3章 系统设计**

## **3.1系统设计思想**

本课题的目的是设计一款基于web的图书管理系统以实现图书的借阅和管理。本系统不仅可以避免用户必须安装专业软件才能开发系统或者访问系统的局限性，而且更加便利。该系统在不改变和影响用户操作习惯的前提下主要完成对有用信息的记录。

整个软件采用Vue.js，Element-ui，Express框架，分别对以下三块进行详细介绍：

（1）Vue.js：

Vue (读音 /vjuː/，类似于 view) 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](https://cn.vuejs.org/v2/guide/single-file-components.html)以及各种[支持类库](https://github.com/vuejs/awesome-vue" \l "libraries--plugins" \t "https://cn.vuejs.org/v2/guide/_blank)结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

（2）Element-ui（Ui框架）：

Element UI是一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库，包含了一套设计原则、组件和组件，还提供如Axure组件、Sketch模板的设计资源。

（3）Express：

Express 是一个保持最小规模的灵活的 Node.js Web 应用程序开发框架，为 Web 和移动应用程序提供一组强大的功能。 API使用您所选择的各种 HTTP 实用工具和中间件，快速方便地创建强大的 API。 性能Express 提供精简的基本 Web 应用程序功能，而不会隐藏您了解和青睐的 Node.js 功能。框架许多 [流行的开发框架](https://www.expressjs.com.cn/resources/frameworks.html) 都基于 Express 构建。

## **3.2系统总体设计**

根据前面的各项设计分析，按照系统开发的基本理念对系统进行分解。

## **3.3系统功能模块设计**

1. 系统登录：系统登录是用户访问系统的路口，设计了系统登录界面，包括用户名、密码和班级，然后对登录进来的用户判断身份信息，判断是管理员用户还是普通用户。
2. 用户管理：不管是超级管理员还是普通管理员都需要管理系统用户，包括普通管理员的添加、删除、修改、查询，修改管理员的登录密码，新添加的管理员用户可以登录系统。
3. 修改密码：系统所有用户（管理员和普通用户）应该都要能修改自己的登录密码，修改后需要重新登录。
4. 个人资料管理：由普通用户使用，普通用户登录系统后，可以修改个人原始信息。
5. 在线借阅：普通用户登录个人账号后，可以选择某个图书，进行借阅，查看别人的借阅信息。
6. 借阅管理：管理员对图书的借阅管理，查询所有用户的借阅记录删除、修改。
7. 图书信息管理：管理员管理图书信息，包括图书信息增加，图书信息删除，图书信息更新，对应数据库的图书信息表，增加图书信息，要对图书编号，图书名称，图书分类，添加时间等属性进行录入。

## **3.4系统安全设计**

图书管理系统才为了防止被暴力登录，设置了登录加密功能。除此之外，为了保证数据的安全性，用户密码被加密后存入系统数据库中，此外数据库采用的是Mysql相对稳定，数据通过人工备份的方式进行定期备份。系统整体上数据业务逻辑前台页面做到了很好的分离，这也是能够相对保证安全。

## **3.5表单字段校验处理设计**

1.出错信息类型

出错信息类型包括：

A 未输入必填项；

B 应输入合法字符或数字却输入非法字符或数字；

C 对空数据表进行更改或删除操作；

D 向要求唯一值的关键字段添加重复值。

2.出错处理对策

对于错误A，系统要求用户输入非空值。

对于错误B，系统要求用户输入合法字符。

对于错误C，系统提示数据表无记录可更改或删除。

对于错误D，系统要求用户输入非重复值。

## **3.6系统维护设计**

图书管理系统是个现代化的高度集成的综合信息系统，系统主要服务对象为用户，最大化在页面中提供多的信息，因此在系统维护上，数据量较大，整体的维护设计如下：

（1）代码的维护：部署在云服务器上或本地服务器，通过SVN保持版本更新迭代。

（2）功能增加：功能升级在必要的前提下。

（3）数据维护：根据安全性等需求，定期对数据库进行人工备份。

### **3.7数据库设计**

### 3.7.1数据库设计概述

数据库的设计在整个系统开发的过程中扮演者关键的角色，前期考虑不全面修改数据库有可能整个系统都要重新开发，所以数据库的设计不容忽视。数据库具有完整性和一致性，并且数据库的响应速度非常的快。在图书管理系统中，设计数据库表的各个字段，确定表和表之间的关联关系，在后期的开发中尽量减少改动数据库，提高开发效率。

### 3.7.2概念模型设计

数据库的前期设计最关键的步骤是概念结构的设计，是数据模型的概念信息。将需求分析抽象为信息世界的结构，才能更好更准确的使用数据库管理系统。其设计具有以下两点优势：

（1）能够体现事物之间的关联关，满足用户对数据的处理要求。

（2）易于理解、更改，用户的积极参与是数据库设计成功的关键。

（3）描述概念模型最有力的工具就是E-R模型，使用E-R图可以清楚的指导各实体之间的联系，它有三个关键的元素，即实体、属性和联系。

建立数据库之前，要对系统的数据进行概念模型设计，设计实体包含哪些属性，实体和实体直接的关系是怎么样的，根据概念设计，得到下图的系统总体ER图。



图4-2系统总体ER图

### 3.7.3数据库表设计

在服务器上建立名为mybook的数据库，其中包括以下表：

User表:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段类型 | 大小 | 允许为空 | 最大长度 | 备注 |
| 1 | ID | Int | 4 | 自增编号 | 10 | 用户ID |
| 2 | Name | VarChar | 50 | 是 | 255 | 用户姓名 |
| 3 | pwd | VarChar | 50 | 是 | 255 | 用户密码 |
| 4 | Admin | VarChar | 50 | 是 | 255 | 用户身份 |
| 5 | Status | Int | 4 | 是 | 10 | 用户状态 |
| 5 | Last\_login\_time | DateTime | 8 | 是 | 23 | 最后一次登录时间 |

Book\_info表:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段类型 | 大小 | 允许为空 | 最大长度 | 备注 |
| 1 | ID | Int | 4 | 自增编号 | 10 | 图书编号 |
| 2 | Name | VarChar | 50 | 是 | 255 | 图书名称 |
| 3 | Author | VarChar | 50 | 是 | 255 | 图书作者 |
| 4 | Press | VarChar | 50 | 是 | 255 | 出版社 |
| 5 | Press\_time | VarChar | 50 | 是 | 255 | 出版时间 |
| 6 | Price | VarChar | 50 | 是 | 255 | 价格 |
| 7 | ISBN | VarChar | 50 | 是 | 255 | 图书编号 |
| 8 | Desc | VarChar | 50 | 是 | 255 | 图书描述 |

Borrow\_list表:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段类型 | 大小 | 允许为空 | 最大长度 | 备注 |
| 1 | Book\_id | Int | 4 | 自增编号 | 10 | 图书编号 |
| 2 | User\_id | Int | 4 | 自增编号 | 10 | 用户编号 |
| 3 | Borrow\_date | VarChar | 50 | 是 | 255 | 借阅时间 |
| 4 | Back\_date | VarChar | 50 | 是 | 255 | 归还时间 |

### 3.7.4数据库连接计

本节在Mysql中利用SQL语句和图形界面相结合的方法建立各个表并设置其主外键联系，并在项目中对数据库的连接进行详细设计。

建立一个数据库助手类来执行基本的数据库增删改查工作，在源文件建立数据库连接的主要代码是：

var app = express();

var http = require('http');

var server = http.createServer(app);

const mysql = require('mysql');

module.exports = {

  config: {

    host: 'localhost',

    port: '3306',

    user: 'root',

    password: '123',

    database: 'mybook'

  },

  sqlConnect: function (sql, sqlArr, callback) {

    var pool = mysql.createPool(this.config);

    pool.getConnection((err, conn) => {

      if (err) {

        console.log('---数据库连接失败---')

        return;

      }

      conn.query(sql, sqlArr, callback);

      //释放

      conn.release();

    })

  },

}

每次执行数据库的基本操作时询问数据库的连接状态，执行完数据库的基本增删改查工作后需要断开连接：conn.release()。

## 3.8本章小结

本章主要论述了开发本系统时对系统进行的总体设计，然后对系统的普通用户模块和管理员模块进行功能的设计，最后对系统的各个模块进行划分，详细介绍如何设计。

# **第4章 系统的实现**

### **4.1主页面的实现**

经过用户登录后，页面跳转至后台首页，首页是由2个页面组成的，包括header、aside和container等页面，通过router路由引入，这里也是浏览者操作系统功能的入口。

其中载入页面的主要代码如下：

Vue.use(VueRouter)

const routes = [{

    path: '/',

    redirect: 'login',

  },

  {

    path: '/login', //登录界面

    name: 'login',

    component: () => import('../components/Login.vue')

  }, {

    path: '/regist',//注册界面

    name: 'regist',

    component: () => import('../components/Regist.vue')

  }, {

    path: '/home',//主页界面

    component: () => import('../views/Home.vue'),

    redirect: '/userlist', //用户列表界面

    children: [{

      path: '/userlist',

      name: 'userlist',

      component: () => import('../views/users/UserList.vue')

}

...

}

### **4.3登录模块的实现**

系统的登录是使用该系统的第一关卡，用户在登录界面输入账号和密码后，系统判断该账号密码是否能在数据库中存在并且匹配，返回登录成功或登录失败的信息，登录成功后才能使用系统的功能。

系统的登录窗口是用户的入口，用户只有在登录成功后才可以进入访问。通过在登录提交表单，后台处理判断是否为合法用户，进行页面跳转，进入系统中去。

登录合法性判断过程：用户输入账号和密码后，系统首先确定输入输入数据合法性，其不能为空，然后调用数据库查找userlist表中数据，进行匹配查看此账号是否存在和密码是否正确。

在主页中切入登录窗口页面，调用设计好的CSS、JS和HTML等文件对页面进行整体设计，创建表单，在login.vue页面发送登录请求，调用方法来验证。当系统判定账号密码都正确，系统会跳转到系统的首页。

用户登录模块的IPO如下所示：

输入：用户名和密码。

处理：

1）检验用户是否满足输入的要求，即检验用户名和密码文本框是否为空，若为空，则提示用户输入用户名和密码。

2）检验是否存在用户输入的用户名，并且密码是否正确。

3）根据用户名，将其显示在系统首页上。

4）关闭窗口。

输出：是否成功的信息。

登录流程图如下图所示。



图5-1登录流程图

用户登录界面如下图所示。

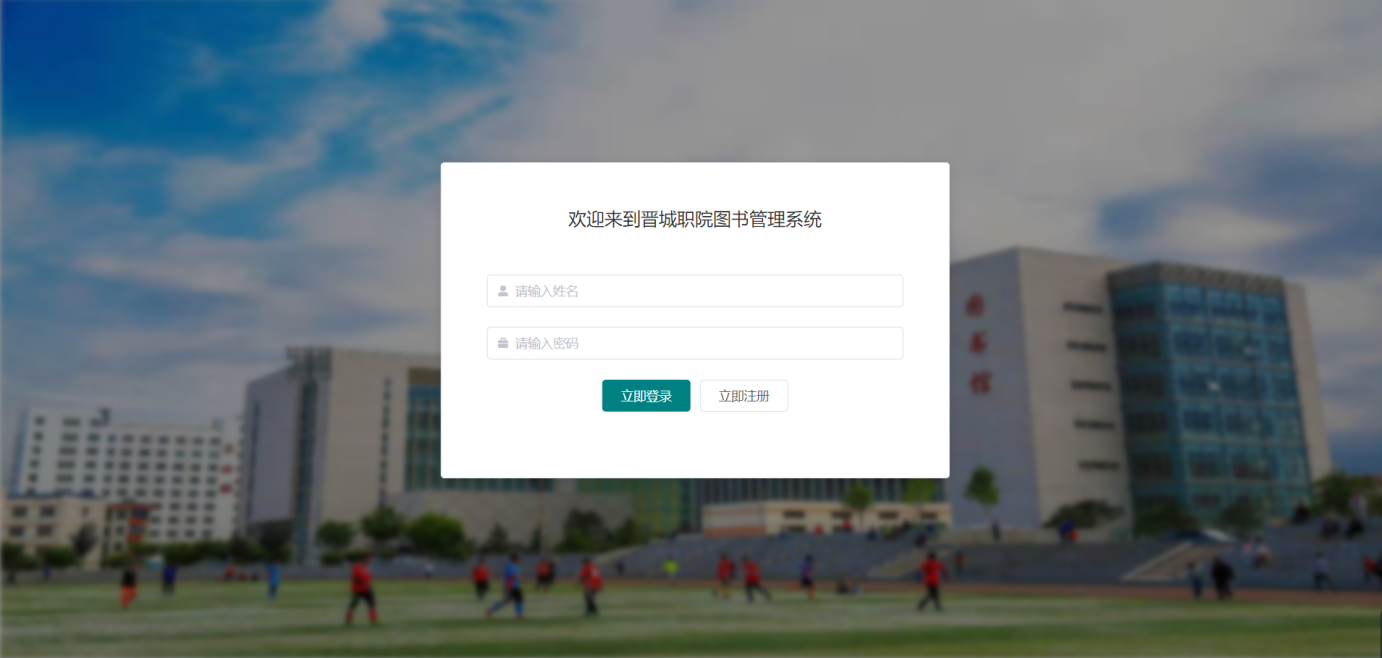


图5-2用户登录界面



图5-3用户注册界面

其中部分登录注册页面的主要代码如下：

submitForm() {

      this.$refs.loginForm.validate(async valid => {

        if (!valid) return this.$message.error("请输入有效的用户名和密码！");

        const data = await getLogin(this.loginForm);

        if (data.code != 200) {

          this.$message.error(data.msg);

          return;

        }

        window.sessionStorage.setItem("user", JSON.stringify(data.data));

        this.$message.success(data.msg);

        this.$router.push("/home");

      });

}

 submitForm() {

      this.$refs.registForm.validate(async valid => {

        if (!valid) return this.$message.error("请完善有效的注册信息！");

        const data = await getRegist(this.registForm);

        if (data.code != 200) return this.$message.error(data.msg);

        this.$message.success(data.msg);

        this.login();

      });

    }

随着系统规模的壮大，系统发布更新的信息及对普通用户量会越来越多，超级管理员有权限可以增加管理员的模块。管理员角色不同对应权限亦不相同。

系统用户管理界面如下图所示。

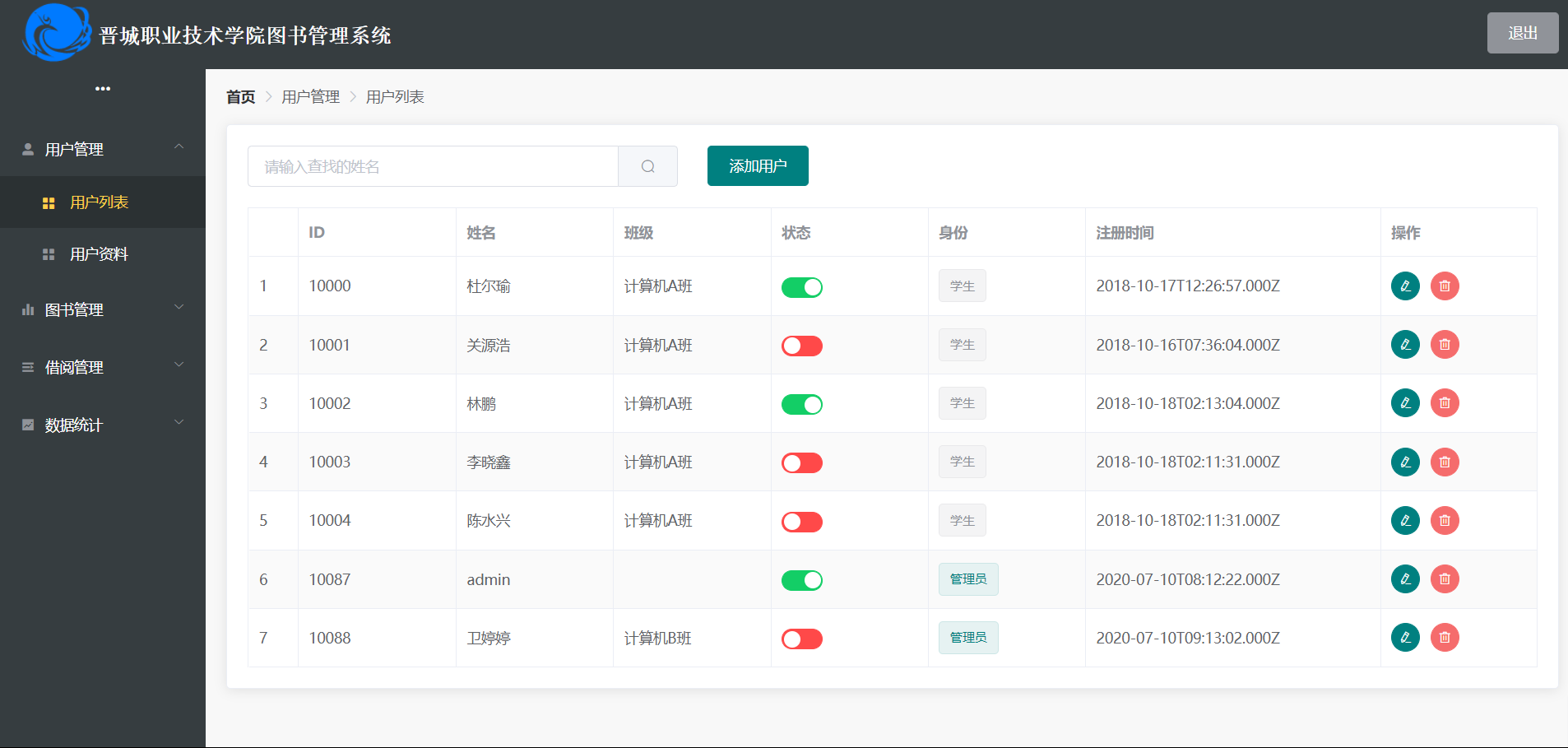


图5-5系统用户管理界面

其中部分用户管理页面的主要代码如下：

async getUserlist() { //获取用户基本信息

      const data = await getUserAll();

      if (data.code != 200) return this.$message.error(data.msg);

      this.userData = data.data;

    },

    async switchChange(obj) {//修改用户状态

      console.log(obj, obj.status);

      obj.status = obj.status ? 0 : 1;

      const data = await updateUserstat({ id: obj.id, status: obj.status });

      if (data.code != 200) return this.$message.error(data.msg);

      console.log(data);

      this.$message.success(data.msg);

    },

    async findUserClick() {//查找用户

      const data = await findUsername({ name: this.searchName });

      if (data.code != 200) return this.$message.error(data.msg);

      this.userData = data.data;

    }

### **4.4借阅模块的实现**

借阅者对状态为还书或者没有状态的图书进行借书操作，借阅者选择借图书信息，两者建立关联，生成借书记录，借书成功后，如果没有还书操作，系统会在应还日期到期时自己的进行还书操作，借阅者可以通过借阅者编号、借阅者姓名、书籍编号、图书类别等关键字搜索借阅记录。添加图书借阅是在“查询图书信息 “的前提下操作的。当页面显示查询的图书信息后，可以点击“借阅记录添加”，既可跳转到borrowadd.vue页添加借阅。管理员在borrowlist.vue通过查询数据库的借阅表列出所有借阅信息。

添加借阅界面设计效果如下图所示。



图5-12添加借阅界面

其中用户记录管理页面的主要代码如下：

searchClick() {//搜索

      this.getBookData();

    },

    async getBookData() {//搜索书籍相关信息

      if (this.selectType == 1) {

        this.getBookList();

      } else {

        let data;

        if (this.selectType == 2) {

          data = await findBookAuthor({

            author: this.bookParams.bookname

          });

        } else if (this.selectType == 3) {

          data = await findBookIsbn({

            isbn: this.bookParams.bookname

          });

        } else if (this.selectType == 4) {

          data = await findBookId({

            id: this.bookParams.bookname

          });

        }

        if (data.code != 200) return this.$message.error(data.msg);

        this.$message.success(data.msg);

        this.total = data.data.length;

        this.bookList = data.data;

      }

    },

    async getBookList() {//馆内书籍列表

      const data = await findBook(this.bookParams);

      if (data.code != 200) return this.$message.error(data.msg);

      this.$message.success(data.msg);

      this.total = data.data.total;

      this.bookList = data.data.result;

}

...

借阅管理页面效果如下图所示。

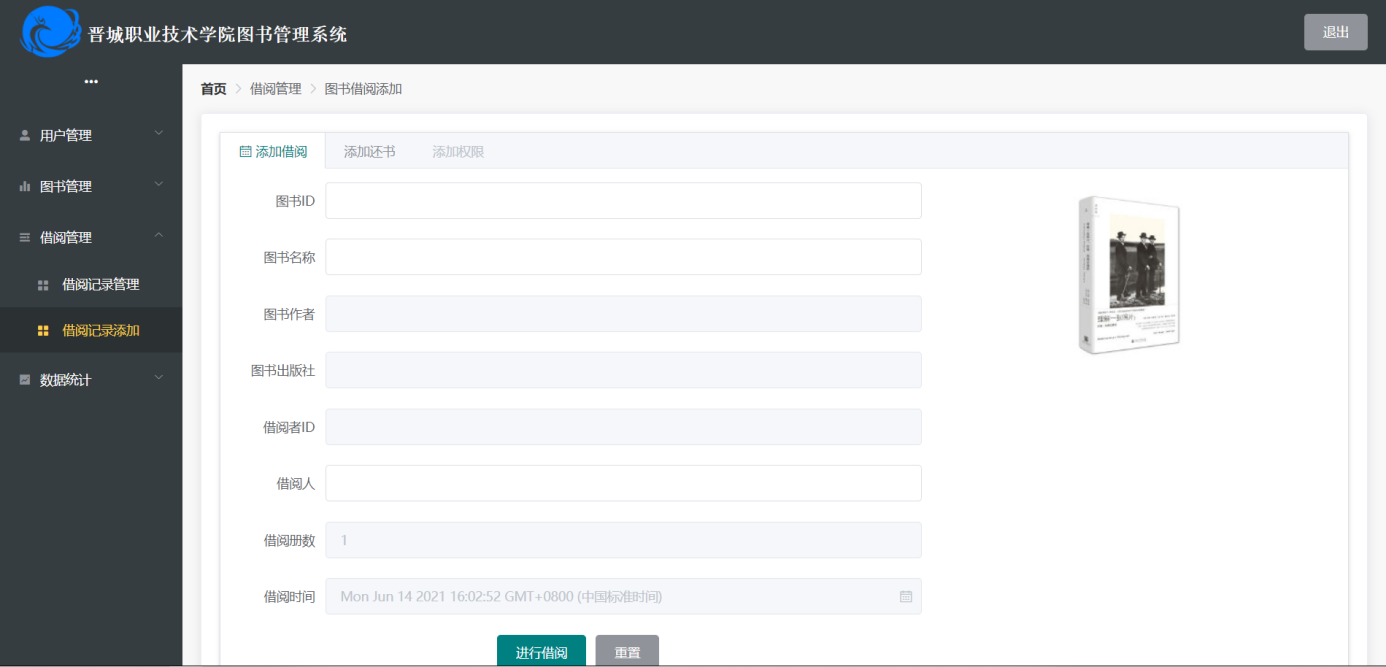


图5-12借阅管理界面

其中用户记录添加页面的主要代码如下：

   addBookClick() {//进行借阅响应的事件

      this.$refs.addbookForm.validate(async valid => {

        if (!valid) return this.$message.error("请完善书籍表单的内容");

        const data = await addBook(this.addForm);

        if (data.code != 200) return this.$message.error(data.msg);

        this.$message.success(data.msg);

        this.activeTab = "1";

        this.$refs.addbookForm.resetFields();

      });

    }

  }

### **4.5数据统计模块的实现**

对所有注册相关用户借阅情况数据的统计

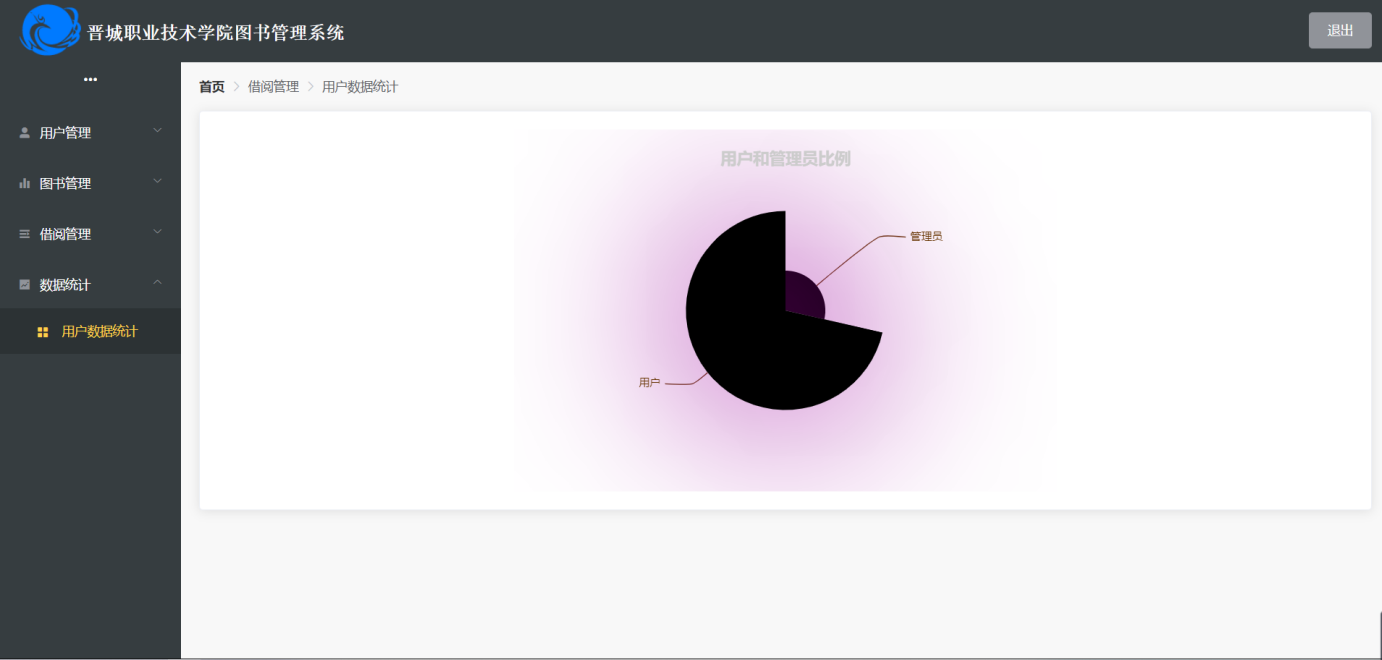


图5-9还书操作界面

其中用户数据统计页面的主要代码如下：

async request() {

      let data = await userDataCount();

      if (data.code != 200) return this.$message.error(data.msg);

      this.$message.success(data.msg);

      // 基于准备好的dom，初始化echarts实例

      var myChart = echarts.init(document.getElementById("main"));

var option = {

        title: {

          text: "用户和管理员比例",

          left: "center",

          top: 20,

          textStyle: {

            color: "#ccc"

          }

        }

...

}

### **4.7图书管理模块的实现**

管理员添加图书信息是在点击添加按钮的前提下操作的，当页面跳转至bookinfo.vue，添加成功后，管理员在booklist进行图书管理，request请求通过查询数据库的图书表列出所有图书信息，每条图书对应一个删除按钮和修改按钮，当管理员点击删除按钮，直接在数据库删除图书信息，并重定向当前页面，当管理员选择点击修改，则弹出书籍相关信息对话框，进行图书信息的修改。

图书管理流程图如下图所示。



图5-11 图书管理流程图

图书添加页面设计效果如下图所示。

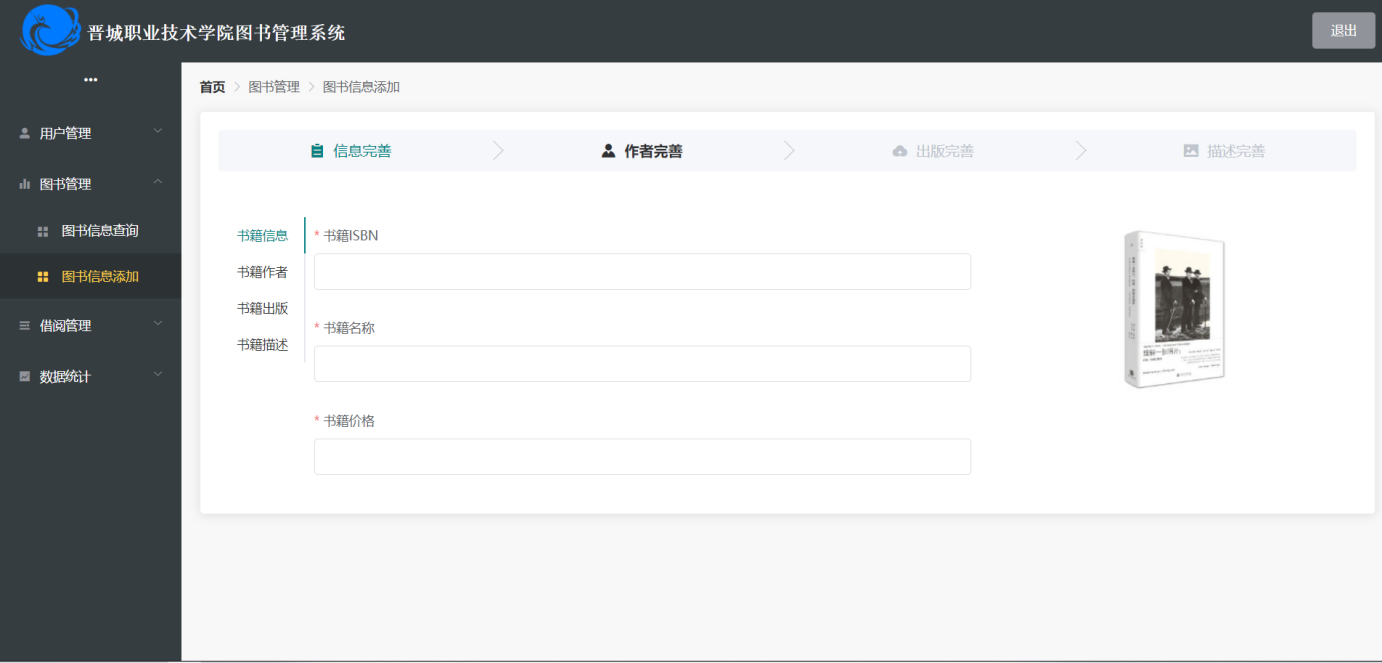


图5-12图书添加界面

图书管理页面效果如下图所示。

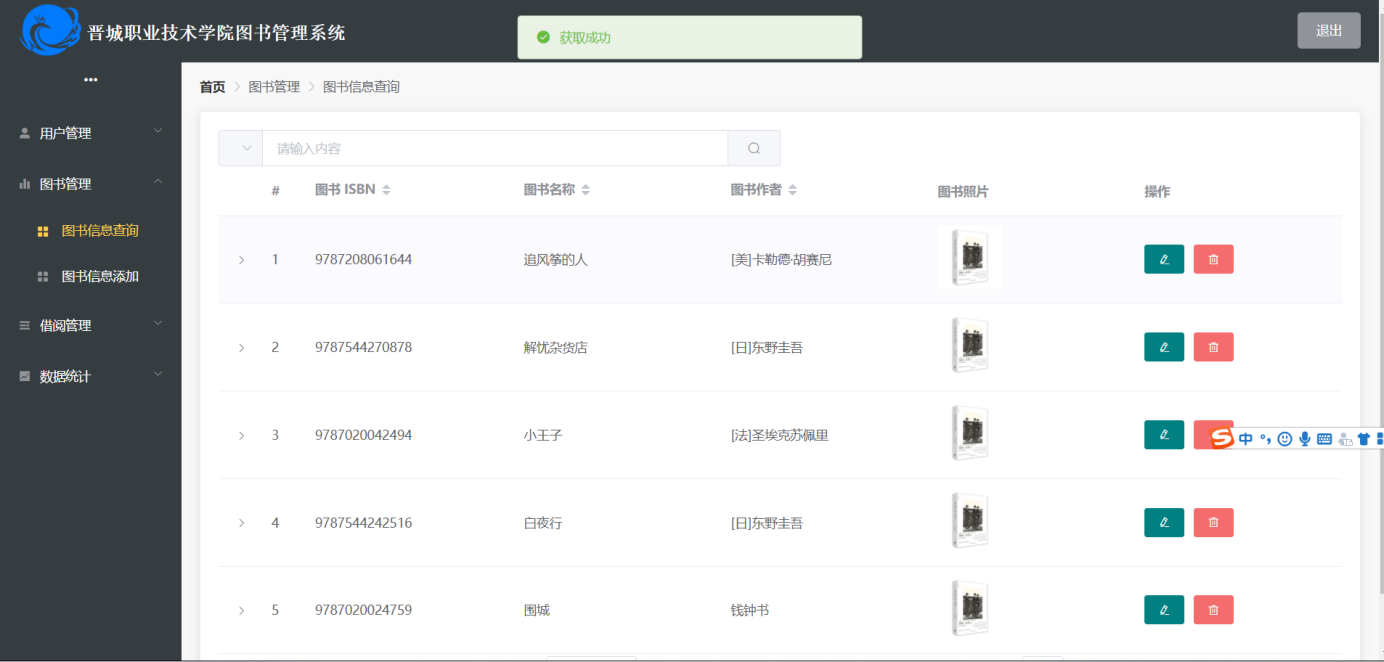


图5-12图书管理界面

### **4.8本章小结**

本章节主要论述了登录模块用户管理、图书管理、借阅管理等功能模块的设计与代码的编写，以及最终实现的步骤。

# **第5章 系统测试**

## **5.1测试目的**

软件测试能够识别项目风险，为开发人员和程序经理提供软件测试的反馈结果，为风险评估提供必要的信息。再者，软件测试确保在上线日前达到上线标准。包括持续追踪项目进度和严格把控各个开发阶段的产品质量。

本系统主要测试客户端的使用和后台服务器的使用。客户端主要是测试用户注册、用户登录、用户完善信息、页面是否进行跳转、读者资料管理、图书借阅排行查询、图书借阅、还书添加等功能是否实现。后端主要是测试系统用户管理、图书信息管理、读者信息管理、图书借阅管理、图书归还管理、系统管理的功能是否实现。

## **5.2界面测试**

测试本系统的界面，测试界面是否正常、可用。

用户界面测试检查表如下表。

表6-1用户界面测试表

|  |  |  |
| --- | --- | --- |
| 测试项 | 测试人 | 测试结果 |
| 窗口切换、移动、改变大小时正常吗？ | 本人 | 正常 |
| 各种界面元素的文字正确吗？（如标题、提示等） | 本人 | 正常 |
| 各种界面元素的状态正确吗？（如有效、无效、选中等状态） | 本人 | 正常 |
| 各种界面元素支持键盘操作吗？ | 本人 | 正常 |
| 各种界面元素的布局合理吗？ | 本人 | 正常 |

## **5.3功能测试**

1.用户登录测试

当用户以“admin”身份登录，密码为空或不是**“456**”时，提示框会提示“密码错误”或“请输入有效的用户名和密码”

图6-1用户登录测试界面

当用户以“admin”身份登录，密码为 “123”时，提示框会提示“登陆成功”



图6-2用户登录测试界面

2.主要功能测试

对系统做了相关测试，主要功能测试用例如下表所示。

表6-3主要功能测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 用例描述 | 预期结果 | 备注 |
| 密码修改 | 在用户管理中修改密码，原始密码若输入不正确 | 提示“原始密码错误” | 测试原始密码是否正确 |
| 图书管理 | 必填文本框中不输入任何数据，如标题和类别 | 提示“请填写完整的信息” | 测试标题和类别是否为空 |
| 在封面栏输入非图片格式数据 | 提示“封面栏必须为图片格式 | 测试封面栏是否能为非图片格式 |
| 输入完整的图书信息” | 提示“图书添加成功” | 测试图书录入是否可用 |
| 图书借阅 | 选择未借阅的图书，输入借阅时间，点击提交 | 系统提示借阅成功 | 测试借阅录入是否可用 |
| 未输入借阅时间，点击提交 | 系统提示添加失败 | 测试借阅时间是否可以为空 |

## **5.4测试结果**

图书管理系统经测试和调试后能够按照需求正常运行，基本没有错误，能够满足开发者和用户的需求。在系统整体测试过程中，系统功能相对来说比较简单，数据源的配置，需进一步改善。

## **5.5本章小结**

本章主要论述了系统开发结束后，要对系统进行各方面的测试，开头介绍了系统的运行环境，需要哪些工具，由于本系统属于毕业设计，是小型的系统，所以本人只对系统进行采用一些单独的测试数据对系统进行输入输出的结果判断，最后对测试结果进行总结。

# **第6章 总结**

在老师的悉心指导下，在自己不断拼搏下的努力我的毕业设计终于接近了尾声，这段日子里有过心酸有过坎坷，感觉自己收获了许多的知识。此次毕业设计的要求是要创建一个图书管理系统，使用的开发软件是visualstudio编辑器以及Mysql数据库。为了更好的开发和设计图书管理系统，本人尽可能多的去了解并掌握有关JS语言和数据库的专业知识，通过查找资料和教学视频来自主学习。对书中的一些经典案例，不断的揣摩他的编程技巧，尽可能的弄明白每行代码的意思，老师也不厌其烦为我解答困惑。积累了大量基本知识之后，开始构思毕业设计。根据老师及任务书的要求，将整个系统分为若干个模块进行设计以满足程序所需要的功能。由于实战经验的不足，在设计过程的当中走了许多弯路，及时停止毕业设计的开发，通过不断的充电和学习来调整自己的心态，老师也为我传授了许多心得与经验，让我学会了一些巧妙的方法来改进程序。

完成之后，立马对程序进行了调试，由于前期充分的准备和积累大量的经验，调试过程当中相对比较顺利。虽然开发的过程当中无比艰辛但我也从里面学到了十分昂贵的经验，这是一生用之不竭的财富。

这些日子以来，让我明白了一个道理：不要害怕困难，不要恐惧，坚持才是胜利。这次的毕业设计不仅是让我对这些年所学知识有了更好的综合性整理，同样也让我学会了遇事不慌、沉着冷静的解决问题的方式，明白了团队协作的不可或缺，打开了视野，增长了知识，为我以后进一步走向社会打下了最坚实的基础。

# 

# **致 谢**

大学的学习生活总是感觉短暂的，总感到知识还没有学够，但大学毕业就在我们面前，即毕业设计即将的结束，意味着我们的大学生涯即将结束。毕业设计是我们在大学所学知识的一次总结，更是对我们过去所学知识的提炼和升华，通过毕业设计，我们认识到大学学习过程中还有那些知识没有掌握，那些知识需要我们进入深入研究。这次毕业设计让我学到了很多新的知识，研究了一些新的以前自己不熟悉的领域，使我处理问题的能力得到了一定的提升，同时也为今后的继续深造或踏入社会大门提供了很好的锻炼机会。

我也曾经沮丧过，失落过，想放弃过，可是因为你们这不厌其烦的鼓励和支持才使得我坚持了下来，这里我要对我的老师和同学们说句“谢谢你们”。因为我们的知识是有限的，所以本文将不可避免的会出现一些错误和不足之处，还请老师指正。

**参考文献**

[1]刘博文 深入潜入Vue.js[M].人民邮电出版社，2019.3.

[2]刘新锋.图书管理系统的问题及应对措施[J].法制社会，2013，4（3）：223-230.

[3]丁惠成，刘国灿.关于图书管理系统的思考[M].山东教育出版社，2013，20-100.

[4]韩志超.国外的系统建设与管理.中国电脑期刊[J]，2014 ，5（1）：200-300.

[5]张玉起.国内外图书管理系统现状及其发展[J] .今日科苑， 2015，7（1）：1-200.

[9]朱越，图书管理系统的设计与实现[D].东北师范大学，2011年.

[10]周翊鹏.图书管理系统的设计与实现[D].电子科技大学，2012年.

[11]张本成.图书管理系统的设计与实现[J] .渝西学院学报(自然科学版)，2014，2（2）：30-40.

[15]宋健，陈士俊.图书管理系统的变革因素与趋势分析[J].未来与发展，2012，3(08)：40-50.