

**本科毕业设计(论文)**

**FINAL PROJECT/THESIS OF UNDERGRADUATE**

**(2020届)**

**图书馆借阅管理系统的设计与实现**

**The Design And Implementation Of The Management System Of Library Borrowing**

|  |  |
| --- | --- |
| **学　　院** | 求真学院理工分院 |
| **专　　业** | 计算机科学与技术 |
| **学生姓名** | 郑兴樑 |
| **学　　号** | 2018284129 |
| **指导教师** | 申情 |
| **完成日期** | 2020年04月 |

# 承诺书

本人郑重承诺：所提交的毕业论文“图书馆借阅管理系统的设计与实现”是在导师的指导下，严格按照学校和学院的有关规定由本人独立完成。文中所引用的观点和参考资料均已标注并加以注释。论文研究过程中不存在抄袭他人研究成果和伪造相关数据等行为。如若出现任何侵犯他人的知识产权等问题，本人愿意承担相关法律责任。

承诺人(签名)： 郑兴樑

日期：2020 年 04月15日

# 摘 要

图书馆是汇聚书籍的海洋，人们在生活闲暇之时不免会和图书进行接触。在这21世纪信息化技术飞速发展的时代[2]，图书的种类和读者的需求量也越来越大，给图书馆的管理加大了难度。利用计算机对信息的高度敏感，开发一款图书馆自动管理系统来改善管理是很有必要的，同时也可以保证了读者们的阅读需求，减轻了图书馆管理员的管理压力[3]。

目前来说，图书馆借阅管理系统是国内一款典型的信息化管理系统，本系统主要是以我校的图书馆的管理现状和一些国内图书馆管理情况作为调研背景，对他们管理系统中的业务处理流程和使用情况作了充分的调查，针对其中出现的一些问题作分析，对图书馆借阅管理系统的总体结构作出相应的规划。

本系统采用前后端分离的开发方式，后端使用Java、SpringBoot框架和Mybatis来完成图书管理的功能,而且以上的技术栈，都是现在国内十分热门的java后端开发技术，在开发技术上用了最了科学的方案。使用的数据库是MySql,该数据库与SqlServe相比，性能更加优越、服务更加稳定，不容易被侵略，容易维护。前端页面使用HTML5、CSS3、JS、Vue等技术栈来完成页面的渲染、动画特效等，增添了图书馆管理员和读者的体验使用感。

**关键词**：信息管理，前后端分离，MySql

# ABSTRACT

Library is a sea of books, people will inevitably contact with books in their leisure time. In this era of rapid development of information technology in the 21st century, the types of books and the demand of readers are also growing, which makes the management of the library more difficult. It is necessary to develop a library automatic management system to improve the management by making use of the high sensitivity of the computer to the information. At the same time, it can guarantee the readers' reading demand and reduce the management pressure of the librarian.

The library lending system is a typical information management system. This system is mainly based on the management situation of our library and the library management system of some domestic colleges and universities as the research background. It makes a full investigation of its working process and use, some unreasonable problems, and then designs the overall structure and regulations of the library lending system Plan to manage the information of books, readers and borrowing in a unified way.

The system adopts the development mode of front-end and back-end separation. The back-end uses Java, springboot and mybatis to complete the query, deletion and modification of the database. Moreover, the above technologies are very popular back-end development technologies in China. The database used is mysql. Compared with sqlserve, this database has more excellent performance, more stable service, small volume and easy maintenance. The front-end page uses H5, C3, JS, Vue and other technology stacks to complete page rendering, animation effects, etc., giving readers and administrators a bright sense of use.

**KEY WORDS**: Information management, Front and rear end separation, MySql

# 目 录

**摘要**

**ABSTRACT**

[承诺书 i](#_Toc38305588)

[摘 要 i](#_Toc38305589)

[ABSTRACT ii](#_Toc38305590)

[目 录 iii](#_Toc38305591)

[第1章 绪论 1](#_Toc38305592)

[1.1课题背景与意义 1](#_Toc38305593)

[1.2系统功能简介 1](#_Toc38305594)

[1.3 系统存在的问题 2](#_Toc38305595)

[1.4 可行性分析 2](#_Toc38305596)

[1.4.1经济可行性 2](#_Toc38305597)

[1.4.2技术可行性 2](#_Toc38305598)

[第2章 开发环境 3](#_Toc38305599)

[2.1 Java开发环境 3](#_Toc38305600)

[2.2前端开发环境 3](#_Toc38305601)

[2.3 MySql简介 4](#_Toc38305602)

[第3章 系统分析 5](#_Toc38305603)

[3.1 业务分析 5](#_Toc38305604)

[3.2 图书馆借阅系统数据流程图 6](#_Toc38305605)

[3.3 数据库的设计 6](#_Toc38305606)

[3.3.1逻辑结构设计 6](#_Toc38305607)

[3.3.2文献信息表结构设计 7](#_Toc38305608)

[3.3.3 读者信息表结构设计 7](#_Toc38305609)

[3.3.4 管理员信息表结构设计 7](#_Toc38305610)

[3.3.5 预约借阅信息表结构设计 7](#_Toc38305611)

[3.3.6 预约排队信息表结构设计 8](#_Toc38305612)

[3.3.7 通知公告表结构设计 8](#_Toc38305613)

[3.3.8 留言信息表结构设计 8](#_Toc38305614)

[3.3.9 资源信息表结构设计 8](#_Toc38305615)

[3.3.10 骑手信息表结构设计 8](#_Toc38305616)

[第四章 系统总体设计 9](#_Toc38305617)

[4.1 系统功能模块图 9](#_Toc38305618)

[4.2学生登录模块设计 10](#_Toc38305619)

[4.3学生书目检索设计 12](#_Toc38305620)

[4.4图书预约模块设计 13](#_Toc38305621)

[4.5学生个人中心模块设计 13](#_Toc38305622)

[4.6管理员登录模块设计 14](#_Toc38305623)

[4.7 管理员预约管理模块设计 14](#_Toc38305624)

[4.8管理员个人中心模块设计 15](#_Toc38305625)

[4.9管理员图书管理模块设计 15](#_Toc38305626)

[4.10骑手端模块设计 16](#_Toc38305627)

[第五章 系统测试与维护 17](#_Toc38305628)

[5.1测试概述 17](#_Toc38305629)

[5.1.1软件测试的目的 17](#_Toc38305630)

[5.1.2软件测试的任务 17](#_Toc38305631)

[5.2单元测试 17](#_Toc38305632)

[5.2.1黑盒测试 17](#_Toc38305633)

[5.2.2白盒测试 19](#_Toc38305634)

[第六章 总结与展望 20](#_Toc38305635)

[6.1总结 20](#_Toc38305636)

[6.2展望 20](#_Toc38305637)

[参考文献 22](#_Toc38305638)

[附录1：图书借阅流程源代码 23](#_Toc38305639)

[前端源码 23](#_Toc38305640)

[后端源代码 28](#_Toc38305641)

[实体类 28](#_Toc38305642)

[Controller层 28](#_Toc38305643)

[Service层 29](#_Toc38305644)

[Mapper层 31](#_Toc38305645)

[Mapper映射层 31](#_Toc38305646)

[致 谢 33](#_Toc38305647)

# 绪论

在这个直奔5G的信息化时代[8]，目前很多图书馆的借阅方式还是采用人工借阅、或者说人工的半自动式管理方式，工作效率很低，而且不能满足读者们的需求。图书馆的浩浩书海和人来人往的读者，读者信息与书籍信息相互依赖产生预约信息、借阅、还书信息，手动管理避免不了信息遗漏、误报。计算机信息管理有着存储量大、速度快等优点，给我们提供了处理图书信息的快捷性。本课题是图书馆借阅管理系统的设计与实现，能实现图书的信息检索，借阅，预约，从而解决预约借阅信息的复杂性。

## 1.1课题背景与意义

如今的电子信息技术日渐成熟，软件开发和移动开发技术也随之得到了迅速的发展，其强大的功能已经为我们深刻认识[2]。如今，信息管理技术已经遍及我们生活中的每一个领域，计算机为我们的生活提供方便些、高效性、灵活性和安全性，尤其是大数据和WEB技术，这将让我们的生活在此提升到另一个层次。图书馆借阅信息管理系统的应用是十分广泛的，利用计算机的特点来进行信息数据管理。

图书馆借阅管理信息系统在我国很多的图书馆，书店都已经得到了应用，其管理效果的优越性已十分明显，让图书资源的利用率得到了很大的提高，管理的便捷程度也得到了很大的提升。另一方面其取得的效果是无形的，是无法估量的。不仅仅是图书馆的管理质量得到提高，同时为图书馆管理员在管理决策上提供了很大的帮助，给到读者们得服务也更加完善，让图书资源利用最大化，促进了管理现代化。在图书馆借阅管理系统的开发设计中，基于对图书馆自动化管理的调研，一切设计思想都是为了让读者在阅读图书中能更方便快捷，图书需求也得到了满足；管理员再管理上便于维护，便于统计，不存在借书证丢失，补办的传统问题发生。

## 1.2系统功能简介

图书馆借阅系统通俗的讲就是MIS(Manangent Infornation SysTem),主要强调的是管理，是一门新型学科。20世纪以来，随着全球经济和科技的蓬勃发展，经济学家们也参与到信息技术的发展当中，他们纷纷提出了新的管理理论。20世纪50年代，信息技术开始用于工作，数据一词从此开始浮现在人们的脑海里。

系统适用于各大图书馆或者是书店，这样促进图书馆建设的现代化发展。主要的功能模块：馆藏图书的添加、书目文献的检索、图书预约管理、续借、学生信息管理、读者留言管理，公告的发布。

馆藏图书的添加：把图书信息录入到系统中，这些信息会体现在图书馆主页端的馆藏图书中，读者可以检索到该书的信息进行预约。通过实验测试，添加一本图书的时间在25秒之内，相比起传统的手工录入效率有了极大的提高，减轻了图书馆管理员的工作量。

书目的检索和预约：学生登录图书馆主页端，不进行登录可以检索图书信息但是无法进行预约。之后可以在主页上搜索你想要的图书，检索内容可以通过书名、图书作者以及ISBN，如果在没有精确的书名，也可选择模糊查询的方式检索。本系统采用的是先预约后借阅排队的借阅形式，对你想要借阅的图书，可以点击“预约图书”来进行预约，预约周期为3天，3天之内前往指定地点借阅，预期则视为自动放弃预约。

图书的续借：本系统的图书借阅周期是30天。到期限，读者若对该书任有阅读需求，可以在期限最后3天进行续借，这样可以保证图书不会被一个读者包揽，图书续借可以保证读者的阅读需求得到满足，也为了防止图书的孤岛现象(图书被一个读者包揽的现象)。

读者可以发表自己对图书馆的看法，让学校能够能到同学们的心声，从而改进；同时在知识的海洋多一点与众不同的味道，大家可以分享有趣的图书内容。

## 1.3 系统存在的问题

由于是一款现代化信息管理系统，对于没有信息化概念的读者和管理员，在系统的使用上有操作难度。需要对管理员进行系统使用的培训，这样才能保证图书馆管理的流畅。

本系统没有连接硬件设备，既然是图书馆自动化借阅系统，那么加上扫条形码的装置(扫码枪)，这样在添加书籍时就更加方便快捷，只需要轻轻一扫就完成了添加图书的流程。在借还书时，不需要通过图书馆管理员来手动操作，通过扫码自动借还书仪器。

## 1.4 可行性分析

### 1.4.1经济可行性

软件经济可行性是指开发软件时投入的资本（财力、人力）和软件上线后能带来得经济效益的对比，同时还要看该系统是否满足广大用户的使用需求，从而推动吸粉量[3]。系统主要目的是为了减轻学校图书馆管理员的工作负担和充分满足读者的阅读需求，从而提高图书馆管理质量，比如：此系统可以随时查看图书的借阅情况，借阅数量等信息，对于管理者的管理决策上提供了很大的帮助，节省了时间，把时间充分利用在学生读者身上，给读者一个更完美的图书馆。时间就是金钱，从带来的经济效益来说，知识是无价的。

### 1.4.2技术可行性

本系统属于一个小型规模的信息管理系统，系统在使用的过程中还存在太多的不确定因素，整个项目的技术结构，还是属于一个简单的数据库应用系统[13]，大部分开发环节都基于数据库查询增删改，出现问题的概率不会很大。考虑到读者的借阅使用和管理员的管理上来说，本系统采用的前后端分离式开发，Java进行后端数据库操作，Vue来实现前端页面的渲染，开发工具使用的是vscode和idea，这两款开发工具目前来说是国内最为流行的软件开发编译器，使用起来十分快捷方便，可以减少代码在编译时发生的错误。后台使用的是MySql完成信息的存储。这些开发技术栈无论从安全性、可用性、可靠性和扩展性来比较都是一流的。

# 开发环境

系统以Java、JavaScript作为前后端开发环境，使用MySql作为关系数据库，将图书、读者等信息进行很好的存储，利用结构化查询语句来完成信息数据的访问、更新，较好的实现图书馆预约系统的功能需求。

## 2.1 Java开发环境

Java是一门面向对象的编程语言,在语法上没有太多的难点,大多数的人都可以很容易的学习上手使用。它丢弃了C++中难以理解模糊不清且存在一定风险的语法结构，比如：指针的概念、自动转换数据类型和多继承等等。我们在开发的过程中，即便Java自身带有垃圾回收机制，但我们也不能忽视内存的浪费，要严格我们的开发规范。

再该项目中，我使用了SpringBoot框架来进行后端项目的搭建。2003年Springboot框架被研发出世[4]，它在原来Spring框架的优秀基础上变的更加简洁明了，毕竟少了配置文件的存在，都是通过注解的方式将对象注入容器；在代码量上，像class类中的set/get方法都无需手动写，只需要一个简简单单的注解来完成，从而进一步简化了开发步骤，使用起来更加方便。该框架本身是基于maven插件,用对象模型的方式来集成第三方依赖包，创建JARs文件和WARS文件，与先前的Spring相比较，不需要在手动下载Jar包[9]。MyBatis支持SQL语句使用，将数据库种的字段完美地映射成Java实体类，而且在对数据库进行信息操作时，他也为我们提供了跟符合我们也无需求的Sql语句，减少了结构化查询语句的代码量。

## 2.2前端开发环境

超文本标记语言HTML5[5]，它并不是一种编程语言。HTML5简单易懂，没有语法的概念，只有属性的要求。在功能上却丝毫不落后，强大到如今的许多Android和ios的App开发中也使用H5来进行页面的过渡。具备绘图画板，添加多媒体文件，地理定位，数据存储等等功能，被认为是互联网的核心技术之一，广泛应用于网页的开发中。

CSS3是CSS的升级版本，他在原来的基础上增添了属性选择器、animation动画、伪类选择器等等，减少了JS脚本的使用量，单纯的样式文件就可以实现我们很多的炫酷功能。由于浏览器各有千秋，CSS3在兼容性上还有待提高，经常会出现IE浏览器样式文件失效，或者是谷歌浏览器布局错位等问题，比较考验开发者的使用水平。

JavaScript(简称“JS”)，我们称它为ECMAScript 5(ES5)[6]，也是一门面向对象编程语言。有的人可能会联想到JavaScript和Java之间必定有什么关联，其实它们根本就没有什么可比较性。JS通常以脚本得方式嵌入在HTML5中来进行使用，完成一些H5和C3无法完成功能。在语法上没有太多的限制，无需像C语言那样进行预编译，比较灵活，通常用于网页前端的开发中。

VUE.JS是基于原生JS的渐进式前端框架，一个vue文件包含了前3者的所有内容，统一了编程风格，使用中可以减少大量的JS DOM元素操作，让我们在开发时更加注重于功能逻辑层的操作，与其它大型框架不同的是，Vue 被设计为可以模块式开发，自底向上引用，减少了不必要地重复页面[12]。Vue 的核心关注视图层，方便与第三方库或插件整合使用。在我们使用中经常会和webpack一起结合使用，对其进行一下打包优化让页面在浏览器中渲染时更加兼容、速度更快，还可以配合其他的一些api，比如:百度地图、element等等。

微信开发者工具，可以在里面创建微信小程序的项目，小程序的开发也是基于MVC的开发模式和Vue类似。由于小程序一般都是小型开发项目，没有子路由的概念。微信小程序和普通的WEB网页最大的不同之处，小程序是运行在微信客户端，而WEB网页是运行在浏览器上，在运行机制上有着根本性的不同[7]。

## 2.3 MySql简介

MySql数据库的移植性较高，可以在诸多的编程语言中嵌入使用，有较大的数据存储量，一般可以存几千万条数据，现在国内的多数企业在项目开发时都以该数据作为首选存储工具。MySql历史悠久，20世纪70年代，Oracle才刚刚崛起，微软得SqlServer还没有出现开发者的视野中[13]。学校课程设计中使用的数据库是SqlServer，而我还本项目中却使用了MySql。原因大致有这么两点：其一是为了结合MyBatis来兼容MySql;其二是相比于SqlServer，MySql体积更加的小、速度更快、总体拥有的成本比较低，尤其是开源码。一般中小型项目开发大多使用该数据库作为网站数据库。

# 第3章 系统分析

## 3.1 业务分析

本系统开发目的是为了图书馆管理员进行图书馆信息的管理，增加工作效率，同时也为读者们提供了更加完美的图书资源需求。面向的使用者是各大图书馆或者是书店。根据实际调查情况，本系统的业务需求如下：

* 1. 图书馆管理员根据权限的高低分为:普通管理员和超级管理员。权限不同，管理员的职责也不同；超级管理员可以冻结或者删除其它的管理员账号，账号一旦被冻结或者删除将无法再登录；这是对于图书馆的人事来说，一旦有新的图书馆工作人员加入，可以为他添加管理员账号。普通管理员处理日常的图书借阅管理。
  2. 图书馆管理员可以发布的公告，读者在浏览主页时可以随时了解到本馆的最新情况或者是学校内的公众信息活动等。
  3. 学生登录，在学校中都会有学生信息的存在，在登录时采用IdCard后6位为默认密码的形式，登录成功后可在个人中心进行密码修改。
  4. 学生登录后可查看通知公告，留言信息，也可以发表自己的留言看法。但是留言内容必须有依据可靠，留言不符合规范，管理员可强行删除留言内容。
  5. 预约图书采用的是预约先来后到的形式，若该书馆藏都被预约了，您就得排队，同时排队的人数不能超过5人，否则该书就无法预约，同一本书在同一时间段内无法预约两次。预约的方式有图书自提和送书上门两种方式。成功预约了“图书自提”，在3天内前往指定地点借阅，预期则视为自动放弃预约。学生可在个人中心取消预约或者对已借阅的图书在借阅的最后3天期限(图书借阅期限为30天)内进行“续借”，前提该书没有预约排队的人；“送书上门”管理员将会指派骑手给你送到是你手上。
  6. 学生“图书自提”预约成功后，来图书馆借阅。图书馆管理员可根据学号来查询匹配信息，信息正确可借阅给学生。还可以根据学号进行还书操作。

## 3.2 图书馆借阅系统数据流程图

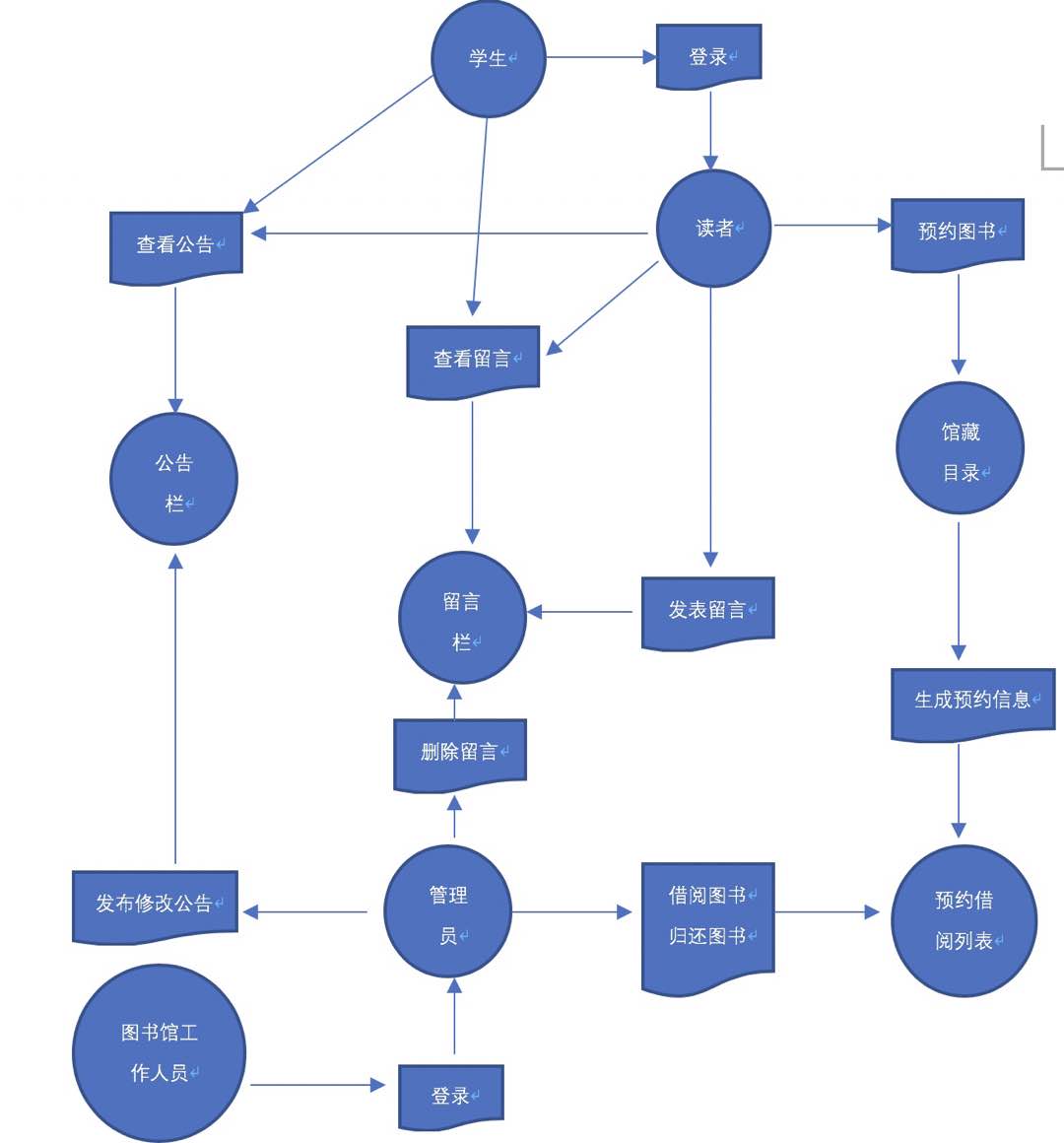


图3-3-1 系统数据流程图

## 3.3 数据库的设计

### 3.3.1逻辑结构设计

1. 文献信息(图书ISBN,图书名称，作者姓名，出版社名称，出版时间，馆藏位置，馆藏数量，封面图片地址，预约人数，借阅人数，图书状态)
2. 学生信息(学号，姓名，年龄，IdCard，联系方式，专业，住址，密码)
3. 管理员信息(管理员账号，管理员密码，管理员角色，管理员状态)
4. 预约借阅信息(预约号，预约图书名称，图书ISBN，读者姓名，学号，预约时间，借阅时间，还书时间，续借时间，预约状态)
5. 公告信息(公告号,公告标题，公告内容)
6. 留言信息(留言号，学号，留言内容)
7. 资源信息(资源号，资源标题，资源链接)
8. 骑手信息(微信openId,姓名，学号，状态)

### 3.3.2文献信息表结构设计

表3-3-1 图书信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段长度 | 是否可以为空 | 字段说明 |
| isbn | varchar | 255 | 否 | 图书isbn(主键) |
| bookName | varchar | 255 | 否 | 书名 |
| author | varchar | 255 | 否 | 作者 |
| press | varchar | 255 | 否 | 出版社 |
| bookDate | varchar | 255 | 否 | 出版日期 |
| bookNumber | int | 255 | 否 | 馆藏书目 |
| address | varchar | 255 | 否 | 馆藏位置 |
| bookUrl | varcahr | 255 | 否 | 图书封面图地址 |
| appNumber | int | 255 | 是 | 预约人数 |
| readNumber | int | 255 | 是 | 借阅人数 |
| type | int | 255 | 否 | 图书状态，“1”上架/“2”下架 |

### 3.3.3 读者信息表结构设计

表3-3-2 读者信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 是否可以为空 | 字段说明 |
| sno | varchar | 255 | 否 | 学号(主键) |
| name | varcahr | 255 | 否 | 姓名 |
| age | varcahr | 255 | 否 | 年龄 |
| idCard | varcahr | 255 | 否 | 身份证号 |
| address | varchar | 255 | 否 | 家庭住址 |
| school | varchar | 255 | 否 | 院系 |
| marjor | varchar | 255 | 否 | 专业 |
| tel | varchar | 255 | 否 | 联系方式 |
| pwd | varcahr | 255 | 是 | 登录密码 |
| token | varcahr | 255 | 是 | 登录验证信息 |

### 3.3.4 管理员信息表结构设计

表3-3-3 管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 是否可以为空 | 字段说明 |
| account | varchar | 255 | 否 | 管理员账号(主键) |
| pwd | varchar | 255 | 否 | 管理员密码 |
| status | int | 255 | 否 | 管理员类型 |
| type | int | 255 | 否 | 管理员状态 |
| token | varchar | 255 | 是 | 登录验证信息 |

### 3.3.5 预约借阅信息表结构设计

表3-3-4 预约信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 是否可以为空 | 字段说明 |
| appId | varchar | 255 | 否 | 预约号(主键) |
| isbn | varchar | 255 | 否 | 图书isbn |
| bookName | varchar | 255 | 否 | 书名 |
| sno | varchar | 255 | 否 | 学号 |
| studentName | varchar | 255 | 否 | 姓名 |
| appTime | varchar | 255 | 否 | 预约时间 |
| lendTime | varchar | 255 | 是 | 借阅时间 |
| backTime | varchar | 255 | 是 | 还书时间 |
| lendingTime | varchar | 255 | 是 | 续借时间 |
| status | int | 255 | 否 | 预约状态 |
| lendingNumber | int | 255 | 否 | 图书排队人数 |

### 3.3.6 预约排队信息表结构设计

表3-3-5 预约信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 是否可以为空 | 字段说明 |
| waitId | varchar | 255 | 否 | 排队号(主键) |
| sno | varchar | 255 | 否 | 学号 |
| isbn | varchar | 255 | 否 | 图书isbn |
| appTime | varchar | 255 | 否 | 预约时间 |
| status | int | 255 | 否 | 预约状态 |

### 3.3.7 通知公告表结构设计

表3-3-6通知公告信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 是否可以为空 | 字段说明 |
| noticeId | varchar | 255 | 否 | 公告号(主键) |
| title | varchar | 255 | 否 | 公告标题 |
| message | varchar | 255 | 否 | 公告内容 |

### 3.3.8 留言信息表结构设计

表3-3-7 留言信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 是否可以为空 | 字段说明 |
| readId | varchar | 255 | 否 | 留言号(主键) |
| sno | varchar | 255 | 否 | 留言学生学号 |
| message | varchar | 255 | 否 | 留言内容 |

### 3.3.9 资源信息表结构设计

表3-3-8 资源信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 是否可以为空 | 字段说明 |
| resourceId | varcahr | 255 | 否 | 资源号(主键) |
| title | varchar | 255 | 否 | 资源标题 |
| resourceUrl | varchar | 255 | 否 | 资源链接 |

### 3.3.10 骑手信息表结构设计

表3-3-9 骑手信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 是否可以为空 | 字段说明 |
| openId | varchar | 255 | 否 | 微信openId(主键) |
| studentName | varchar | 255 | 否 | 姓名 |
| sno | varchar | 255 | 否 | 年龄 |
| status | varchar | 255 | 否 | 状态 |

# 第四章 系统总体设计

## 4.1 系统功能模块图

图4-1-1 图书馆主页端功能模块图

图4-1-2 图书馆管理端系统功能模块图

图4-1-3 骑手小程序功能模块图

## 4.2学生登录模块设计



图4-2-1 学生登录图

在图书馆主页端，学生可以通过学号和密码(首次登陆密码为IdCard后6位数字)进行登录。在未登录的状态下，读者无法进行图书的预约和留言，并且能获取到你自己的个人信息。

该功能在前端上，再点击登录时会进行判断表单信息学号和密码是否填写完整，否则无法提交表单到后端，通过axios进行表单提交，因为提交信息中包含了密码这个敏感数据，所以在提交信息是采用post请求，不会使提交信息展示在url地址栏中，相对于get请求会更加的安全。成功提交信息后，后端进行账号和密码的校验返回给前端一个token值，表示登录成功。token是用于前后端请求接口时的一种约束，获取token值存于浏览器的session Storage中。

1. 该功能前端JS部分代码如下：

/\*调接口方法\*/

export let getStudnetLogin=(data)=>{

return Http.post(`/api/getStudentLogin`,qs.stringify(data));

}

/\*登录表单提交\*/

studentLogin:function(){

getStudnetLogin(this.formData).then(data=>{//this.formData为对象，即学号和密码

if(data.data.msg！=’0’){//不等于0,表示登录成功

windows.setSessionStorage(‘token’,data.data.msg);//存入token值

location.reload();//刷新当前页面

this.$message.success(“登录成功”);//登录成功提示

}else{

This.$message.error(‘登录失败’);//登录失败提示

}

this.model1=false;//关闭当前登陆窗口

})

}

1. 该功能后端代码如下

public TbStudent studnetLogin(String sno，String pwd){//获取前端传过来的账号密码

TbStudent tbstudent=new TbStudnet();//创建空返回对象

/\*获取账号学生信息\*/

Lit<TbStudent> tbStudentList=tbStudnetMapper.selectSnoStudent(sno);

if(tbStudntList.size()<=0){//没有该学生直接返回登录失败

tbStudent.setMsg(“0”);//注入验证信息

tbStudent.setPwd(“0”);

return tbStudent;

}

tbStudnet=tbStudnetMapper.selectOneStudnet(sno);

if(tbStudent.getPwd==null){

/\*判断学生信息有无密码，没有密码将截取身份证后6位密码添加\*/

tbStuentMapper.updatePwd(sno,tbStudnet.getIdCard().subString(12));

student.setPwd(student.getIdCard().substring(12));

}

if(pwd.equles(student.getPwd())){//密码验证成功

tbStudent.setMsg(tbAccountServiceImpl.getToken(20));//验证成功注入token

}else{

tbStudent.setMsg(“0”);

}

tbStudnet.setPwd(“0”);//安全起见，把密码覆盖

return tbStudent;//返回参数

}

1. Mapper接口映射层sql语句

<?xml version=”1.0” encoding=”utf-8”>

<mapper namespace=”com.example.demo.mapper.TbStudentMapper”>

<!—TbStudnet为实体类、TbStudntMapperw为映射接口方法-->

<select id=”selectSnoStudent”resultType=”TbStudent”>

select \* from tb\_student where sno like #{sno}

</select>

<update id=”updatePwd”>

update tb\_student set pwd=#{pwd} where sno like #{sno}

</update>

</mapper>

## 4.3学生书目检索设计



图4-3-1 书目检索图

在图书馆主页端的馆藏目录中，默认会展示图书馆管理后端添加的所有图书。在检索栏中可以根据(精确书名，模糊书名，作者，ISBN)进行分页检索。该功能在表单提交信息后时，有3个参数:检索内容，检索类型和当前页码，默认每页10条图书信息。

其中模糊书名的检索方式，表示书名中包含搜索字段即可(不区分大小写)。Sql语句如下:

select \* from tb\_book where bookName like contact(“%”,#{bookName},”%”)

## 4.4图书预约模块设计



图4-4-1 图书预约

图书预约成功后，在借阅方式上有2种方式：图书自提和送书上门。

图书自提流程：图书预约成功🡪3天内前往指定位置借阅🡪借阅成功🡪30天内归还图书。

送书上门流程：图书预约成功🡪管理员指派骑手🡪骑手接单完成取货🡪骑手根据联系方式完成送货🡪借阅成功🡪30天内归还图书。

## 4.5学生个人中心模块设计



图4-5-1 学生个人中心

学生个人中心内容分为个人信息查看、我的预约情况查看、我的留言查看、本馆的预约制度和退出登录。个人中心展示了当前登录账号的姓名、学号等一些基本个人信息，在此可以修改登录密码；我的预约中可以看到当前你所拥有的的预约信息，比如：图书名称、ISBN、图书馆藏位置，骑手的送货状态等等，也可进行图书预约的取消(选择送书上门无法取消预约)，图书续借；我的留言，可以看到自己的留言信息，若留言不文明或者不规范，图书馆管理员可以将你的留言删除，将无法展示；预约制度，展示图书馆预约系统的一些操作使用方法、出现异常的处理方式和图书馆的一些基本简介等等。

## 4.6管理员登录模块设计



图4-6-1 管理员登陆

图书馆借阅系统后台管理登录采用普通的账号密码的方式登录，在开发中我已经在数据库中初始添加了一个超级管理员账号，提供给系统使用方，后续的管理员账号添加可自行完成。

## 4.7 管理员预约管理模块设计

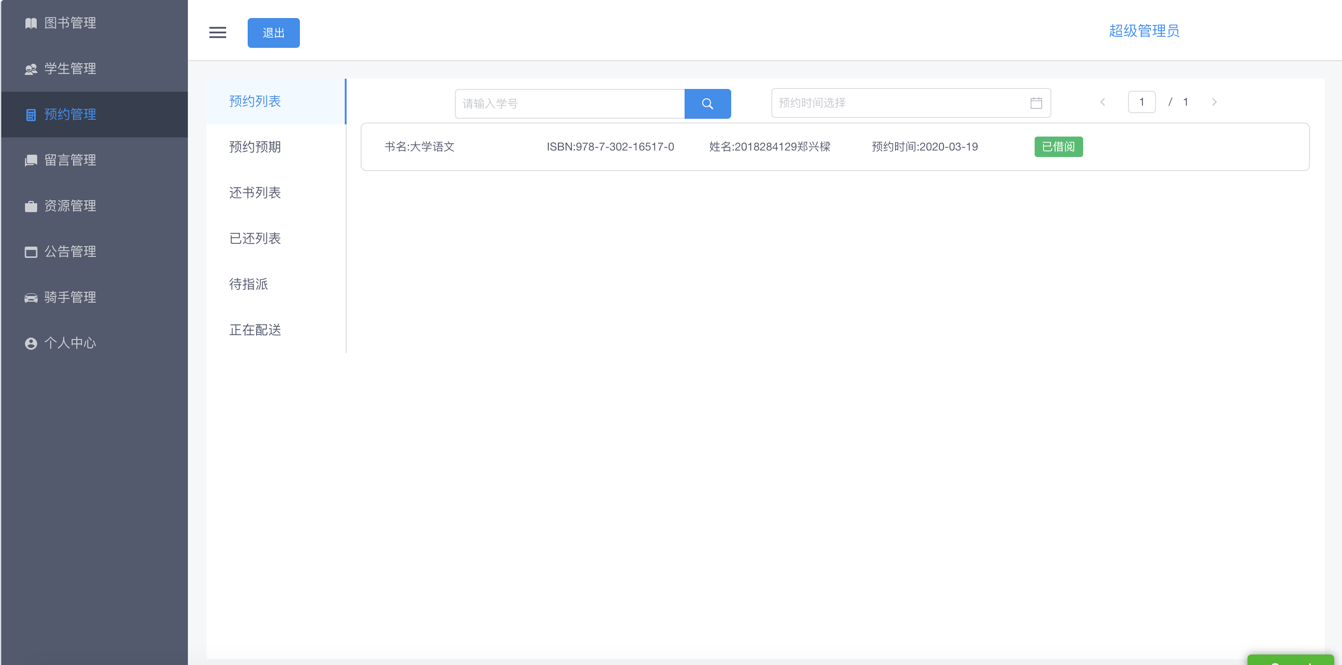


图4-7-1 管理员借还书

管理员可在登录后可以查看预约情况，比如：待借阅的预约信息、待指派的预约信息和预约预期信息等。有学生前来借阅图书只需要查询学生的学号，核对其书名等信息即可借阅。在查找预约信息时，也可通过预约时间获取相关预约信息。

## 4.8管理员个人中心模块设计



图4-8-1 管理员个人中心

## 4.9管理员图书管理模块设计



图4-9-1 图书管理

## 4.10骑手端模块设计



图4-10-1 图书骑手

骑手端微信小程序设计模块，既然是微信小程序，采用的是微信授权登陆的方式，提前注册了微信小程序并且获取appId和秘钥。在小程序的开发中openId为我们每个微信号的唯一标识，后端通过请求微信公众平台开发者文档接口来进行获取。用户登录后可以申请骑手，后台管理员进行审批。最后说明下，我开发骑手端的理由：图书预约的预约期限有3天，这个预约预期时间上还是很合理的，但依旧还有1到2天的空窗期，利用骑手端，可以达到当天17：00之前预约可以当天给你送达，之后则是次日送达，这样在时间上会更加紧凑；图书馆可以招募勤工俭学的学生进行送货，范围不会超过学校，不会有安全事故，也可以让读者借阅更加方便，促进了图书的利用；当然，如果认为骑手端无需使用，可以在后台管理个人中心中关闭，这样预约默认只有图书自提的预约方式。

# 第五章 系统测试与维护

## 5.1测试概述

### 5.1.1软件测试的目的

软件测试的目的就是在已经成型的业务逻辑条件下，对开发完成的软件进行充当用户的实际操作测试，通过测试去发现程序代码中的问题和不合适操作,这样可以让开发者去了解软件的质量的好坏，是否满足开发需求，然后对于该软件是否具备和现在大众软件对抗的资本。

软件测试是在软件开发后产生的。初期的开发过程中，该环节都是由开发者自行来进行代码测试，比较随意，测试的内容也比较浅，无非相当于我们在上编程课程时的调试代码，单元测试。现在“软件测试工程师”这个职位可谓是十分热门，是软件开发的重要角色，毕竟测试这关不过，程序员们还不算完成项目，而且该岗位的需求量越来越大，十分热门。

### 5.1.2软件测试的任务

我们在开发产品软件之前必定事先规定了产品开发方案 [15]，根据开发方案的业务流程制订测试规则。

成功测试：确定软件的主要功能，能完成什么业务，按照用户的正常流程去使用软件，不用去考验的其能力大小，来进行判断软件的功能是否齐全。

摧毁测试：该测试方法是整个测试过程的核心，是检查软件问题的重要测试方法。纯粹的去使用不合理使用方法，因为软件被其他用户使用的过程当中，不免出现另论别累的使用操作，该测试方法也叫作迫使出错测试，故意攻击软件的软肋位置[11]，软件质量的好坏在该测试环节体现的淋淋尽致。

黑盒测试：类似于成功测试，确认软件的主要功能，比如：能否正常登陆。把软件看作是黑皮质包装的包装盒，看不到其内部有什么内容，所以我们在进行黑盒测试的时候不考虑其内部结构和功能好坏。一般我们在该测试时，在后端提供的接口上进行调试，能否得到相应的输出数据作为判定的依据，还会用在前端界面的一些问题，比如：字体错位，颜色不够美观和原UI图多有不符等。

最后，根据测试结果指定修改方案，进行二次开发和优化。

## 5.2单元测试

### 5.2.1黑盒测试

黑盒测试是从SpringBoot中Controller层提供的接口地址上测试输出结果，具有接受性测试的特点，没有将前后端的内容相互关联，只有数据。在该项目中，我使用springboot配置了swagger的接口文档，这样可以更直观明了的配合测试，如图5-2-1所示：

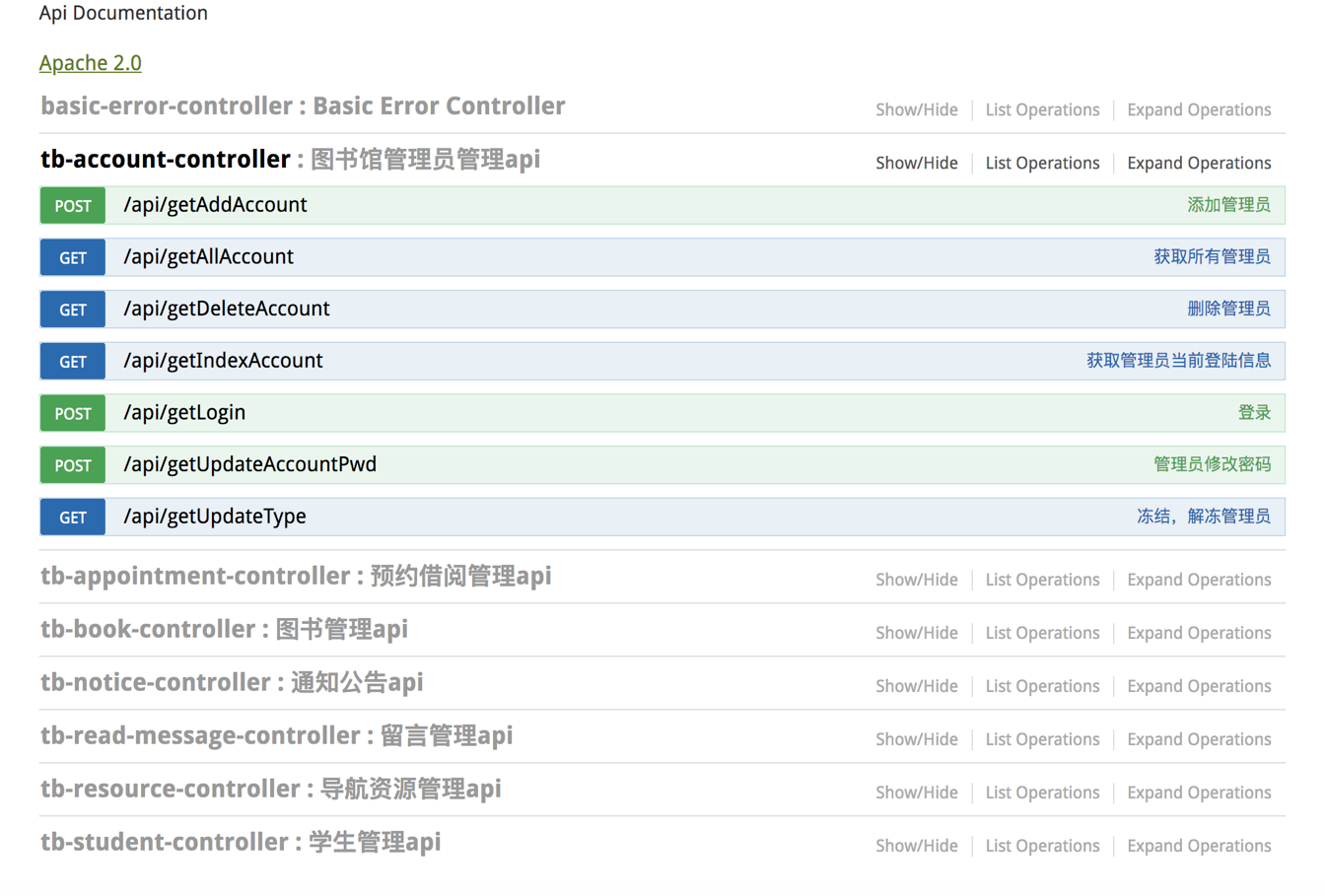


图5-2-1 swagger接口文档

等价类划分：

说明：接口前缀都为localhost:8081,字体加粗的为测试数据传参

1. 测试接口：/api/getAllBook?isbn=**7-121-01018-6**&start=**1**

接口参数：isbn为图书isbn;start为页码

接口功能：根据isbn来查询图书

测试结果：

表5-2-1 接口测试结果表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| key | isbn | bookName | author | press | bookDate | bookNumber | status |
| value | 7-121-01018-6 | JAVA优化编程 | 林胜利等 | 电子工业出版社 | 2005 | 1 | 1 |
| key | appNumber | readNumber | type | address | msg |  |  |
| value | 0 | 0 | 1 | 东校区 | 1 |  |  |

1. 测试接口：/api/getLogin?account=**123**&pwd=**123456**

接口参数：account为管理员账号；pwd为登录密码

接口功能：管理员登陆验证

测试结果:

表5-2-2 接口测试结果表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| key | account | pwd | status | type | msg |  |  |
| value | null | null | null | null | null |  |  |

1. 测试接口：/api/getAppointment?sno=**2018284129**&isbn=**7-121-01018-6**

接口参数：sno为学号；isbn为图书isbn

接口功能：学生预约图书

测试结果：

表5-2-3 接口测试表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| key | appId | bookName | isbn | studentName | sno | appTime | lendTime |
| value | null | null | null | null | null | null | null |
| key | status | msg |  |  |  |  |  |
| value | null | 1 |  |  |  |  |  |

### 5.2.2白盒测试

白盒测试也称为结构功能测试(逻辑驱动测试)，按照业务逻辑要求和最初的产品开发方案进行走流程式测试，检测软件使用每一种情况是否都能按照指定要求完成相应的功能，通过不同点的状态，判断测试状态和预想状态是否一致[14]。

测试内容：

1. 在学生已预约7-121-01018-6图书且没有取消预约的情况下，再次去预约该图书。

测试结果：弹出警告框“您已预约该书，无法预约”。

1. 在学生图书预约成功后，3天之内没有前往指定地点借阅(由于测试，我把3天改成1分钟)，观察后台管理该预约号的预约状态。

测试结果：预约状态变红，显示“预约预期”。

1. 在学生已归还图书，由于个人中心网页未刷新，任然显示“可续借”，点击“可续借”按钮。

测试结果：弹出警告框“预约状态已改变，请重新刷新页面”。

1. 在后台管理端已登录的状态下，把网页SessionStorage中token改变。

测试结果：页面路由跳转回登录页，表示未登录。

1. 后台进行图书修改、删除和下架，观察图书馆主页端馆藏目录的动态。

测试结果：主页端图书信息改变，需要重新刷新页

# 第六章 总结与展望

## 6.1总结

转眼之间毕业日子的已经到来。毕业设计，我以“图书馆借阅管理系统的设计与实现”为课题，在两个月的时间里，顺利的完成。毕业设计是我们在大学生涯的最后一次作业，也是对我们在大学生涯学习情况的一次考验。有的同学在考研中在经历了风风雨雨，最后的结果总是有人欢喜有人有忧；有的人还在忙忙碌碌的找工作，准备踏上事业，兢兢战战。我们在这种状态下来完成大学的最后一个环节，真可谓是对我们的一种考验。

在我之前看来，毕业设计无非是像我们平时实验课后的实验报告，在写作内容上翻了倍而已。但是通过这次的毕业设计，充分地证明了我之前的想法的是有多么的稚嫩，也对自己的专业知识水平有所高估，并没有我想象得那么简单。毕业设计不仅是对自己专业知识的一种检验，更加是理论联系实际，对自己能力的一种提高以便于以后在工作中能更快的融入其中。也明白了学习是永无止境的，是需要不断积累的，学校只是只不过是我们人生中的其中一站，以后在生活中、工作中也要不断的学习，努力修炼自己，以提高自己的综合素养和知识。我是一名计算机科学与技术专业的学生，本次毕业设计就运用了课堂上所学的专业知识，进行长远的科学研究调查，也结合生活中的运作实际情况，分析和解决了一个理论联系实际的问题，把自己的理论知识化为了实际训练。

在两个月的时间里，我已经基本上完成了毕业设计。在开发编程的过程中，我不仅巩固了学校中学到编程技术(Java、数据库)，也见识了许多课堂书中没有的技术，同时更加锻炼了自己的毅力和耐心，毕竟写代码是不能操之过急的，那么会酿成大错。经过一个月的时间，图书馆借阅系统已经基本开发完毕，但是其中还有很多的不足之处，比如：没有和硬件设备联系起来，借还书是人工模式；界面上可以进步优化等等，这些不足之处，后续可以做得更加完善。

在开发图书馆借阅管理系统中，出现了一系列的问题bug,比如：在调试的时候，学生预约图书需要学生登录后才能进行，而时而出现未登录就预约成功，导致数据库报错。还有在获取图书信息，学生信息时，若没有一个满足删选条件，Java代码就会报错等等一系列的问题，很难进行调试。后来经过网上查询和指导老师的细心讲解，渐渐地找出了程序中的问题，也让我学到了如何进行程序的调试。在做毕业设计的过程中，向指导老师请教，老师耐心仔细的检查程序，总是会给予我很大帮助，从而完善系统的功能。

## 6.2展望

基于WEB的图书馆借阅系统的设计与实现涉及到很多的理论知识、方法和技术层，本系统还有很多新的问题需要解决，需要在实际运用中不断的积累、提出新的需求，在今后可以实现更加完美地系统。目前，还有一下几个方面，还需要做进一步的调研和设计开发：

1. 论文的只考虑了课题的发展和功能完善性[2]，并没有从课题的数据统计方面进行调研发掘。系统考虑的用户使用情况比较简单，只考虑了一些基本的情况。还待加入一些大数据开发技术试题信息，可以进一步完善论文的完整性。
2. 由于毕设预算，仅仅只有软件层面的体现，没有参与硬件自动化设备(扫码枪，借书仪)，这样使用可以完全自动化，这方面后期还需要加入C/C#的嵌入式开发技术[8]来完善系统的运行流程。

# 参考文献

1. 张晓颖, 图书馆自动化发展中的一些问题及对策[J]. 晋图学刊，1997(4):55-56
2. 王宇涵, 图书馆借阅管理系统的设计与实现[D]. 天津大学,2016.
3. 庄礼金. 图书在线借阅系统的分析与设计[J]. 信息技术与信化, 2019(11):36-38.
4. 明日科技. Java Web项目开发实战教程. 吉林大学出版社，2017(4).
5. 赵国海，张岩. 基于HTML5开发跨平台在线教育系统[J]. 科技创业家, 2013（8）:46.
6. 陈鲱.Web前端开发技术以及优化方向分析[J].新媒体研究，2015（7）:39-40.
7. 赵柯蔚.移动互联网平台与移动图书馆服务.南阳医学高等专科学校,2017.
8. 孙胜进.图书馆智能化系统设计综述.辽宁省土木建筑学会电气专业委员会.沈阳.2017. 辽宁省土木建筑学会电气专业委员会.2017:30-35.
9. 柳伟卫.SpringBoot企业级开发实战.北京大学出版社.2018.
10. Internation Journal of digital library systems. Josiline Phiri Chigwada. Bindura University of Science Education, Bindura, Zimbabwe,2019(8).
11. 何军，陈倩怡.电脑编程技巧与维护.中国人民银行长沙中心支行.2019(000),002.
12. 梁灏.Vue.js实战.清华大学出版社.2017(10):1-12 .
13. 鹏程，高晶.Java与MySql环境下图书馆信息管理系统设计与实现.佳木斯大学信息电子技术学院.2019(015),008.
14. Glenford J.Myers、Tom Badgett.软件测试的艺术.机械工业出版社.2012(3)
15. 王秩辰，曹志钦.基于混合式学习的软件测试的课程设计.教育教学论坛.2019(000),021

# 附录1：图书借阅流程源代码

## 前端源码

<template>

<div id="lend">

<!--借阅方式-->

<Modal v-model="lendMethods" title="请选择借阅方式" width="400px" @on-ok="appointmentSubmit">

<RadioGroup v-model="border" @on-change="ridioChange">

<Radio label="图书自提" border></Radio>

<Radio label="送书上门" border v-if="bookDoor"></Radio>

</RadioGroup>

</Modal>

<!--借阅方式-->

<div class="el-menu-demo header">

<img src="../../assets/logo.png" height="100%" alt />

<el-input placeholder="请输入检索内容" style="width:40%;margin:40px auto" v-model="searchItem">

<el-select slot="prepend" placeholder="请选择" v-model="select" style="width:100px;">

<el-option label="题名(精)" value="1"></el-option>

<el-option label="题名(糊)" value="2"></el-option>

<el-option label="著者" value="3"></el-option>

<el-option label="ISBN" value="4"></el-option>

</el-select>

<el-button slot="append" icon="el-icon-search" style="background-color:orange;color:white" @click="searchBtn"

></el-button>

</el-input>

</div>

<br />

<Row type="flex" justify="center" style="min-height:600px;">

<Col :lg="1"></Col>

<Col :lg="17">

<span style="width:80%;background:#5d8fb6;">检索记录({{count}}条)</span>

<Page

:current="page" :total="sum" simple @on-change="pageChange"

style="display:inline-block;"

/>

<div style="height:350px;line-height:350px;font-size:40px" v-if="count==0">

<Row type="flex">

<Col :lg="8">

<img src="../../assets/baoqian.jpg" width="100%">

</Col>

<Col :lg="16">

抱歉没有检索到相关图书...

</Col>

</Row>

</div>

<Row

type="flex" v-for="(item,index) in searchBookList" :key="index"

style="border-top:1px solid #B5B5B5;margin-top:2px;"

>

<Col :lg="5">

<img :src="item.bookUrl" width="90px" height="100px" alt="错误,请检查链接" />

</Col>

<Col :lg="19">

<p>

<span style="font-size:15px;color:black;">{{item.bookName}}</span>

<span style="color:#8B3A3A;">

<Poptip trigger="hover" title="馆藏位置" :content="item.address"

>馆藏副本{{item.bookNumber}},已预约{{item.appNumber}},已借阅{{item.readNumber}}</Poptip>

</span>

</p>

<p>

<span>作者:{{item.author}}</span>

<span>ISBN:{{item.isbn}}</span>

<Button type="primary" @click="appointmentBtn(item)" v-if="item.type==1">预约图书</Button>

<el-button type="info" plain disabled="true" v-if="item.type==0">已下架</el-button>

</p>

<p>

<span>出版社:{{item.press}}</span>

<span>年份:{{item.bookDate}}</span>

</p>

<p>

<span>

格式:BK

<Icon type="md-cube" />

</span>

<span>电子书链接</span>

</p>

</Col>

</Row>

</Col>

<Col :lg="4" style="margin-left:10px;">

<div style="border:1px solid #5d8fb6;margin-bottom:10px;">

<p class="headBar">读者信息>></p>

<p class="headBarItem">您好!{{this.user}}用户</p>

</div>

<div style="border:1px solid #5d8fb6;margin-bottom:10px">

<p class="headBar">当前检索>></p>

<p class="headBarItem">{{select | selectFil}}=({{searchItemShow}})</p>

</div>

<div style="border:1px solid #5d8fb6;margin-bottom:10px">

<p class="headBar">主题词>></p>

<p class="headBarItem" style="font-size:20px;">计算机、临床医学、生物、物理、历史、数学、英语、中医药、机械、电气自动化、护理、光电、PLC、电子、船舶、制药、物联网、现代教育、日语、小语种...</p>

</div>

</Col>

<Col :lg="1"></Col>

</Row>

<div class="footer">

湖ICP备2018284129 湖州师范学院图书馆

</div>

</div>

</template>

<script>

import {

getAllBook,getIsbnBook,getBookName1,getAuthorBook,getBookName2,getIndexStudent,

getAppointment,getAllConfig} from "../../api";

export default {

name: "lend",

data() {

return {

select: "",

searchItem: "",

searchBookList: [], //检索列表

page: 1, //当前页码

sum: 0, //检索总数

user: "匿名", //当前登录学号

searchItemShow: "", //当前检索

studentName: "", //登录姓名

lendMethods: false, //借阅方式绑定

border: "图书自提",

bookDoor: true ,//送书上门有没有

appItem:{},//具体提交书本信息

};

},

created() {

this.select = this.$route.query.select || "1";

this.searchItem = this.$route.query.searchItem || "";

this.searchItemShow = this.searchItem;

getIndexStudent().then(data => {

if (data.data.msg == 0) {

this.user = "匿名";

} else {

this.user = data.data.sno;

this.studentName = data.data.name;

}

});

if (this.searchItem == "") {

getAllBook(this.searchItem, this.page).then(data => {

this.searchBookList = data.data;

this.sum = this.searchBookList[0].status;

});

} else {

this.startChanceBook();

}

getAllConfig().then(data => {

this.bookDoor = data.data.riderSwitch == 1 ? true : false;

});

},

methods: {

appointmentBtn(value) {//预约按钮

if (!window.sessionStorage.getItem("token")) {

this.$message.warning("请先登录");

} else {

this.lendMethods = true;

this.appItem=value;

}

},

pageChange(value) {//页码改变回调

this.page = value;

if (this.searchItemShow == "") {

getAllBook(this.searchItemShow, this.page).then(data => {

this.searchBookList = data.data;

this.sum = this.searchBookList[0].status;

});

} else {

this.startChanceBook();

}

},

startChanceBook() {//检索4者选择

switch (this.select) {

case "1": {//题目(精)

getBookName1(this.searchItemShow, this.page).then(data => {

this.searchBookList = data.data;

this.sum = this.searchBookList[0].status || 0;

});

break;

}

case "2": {//题目(模糊)

getBookName2(this.searchItem, this.page).then(data => {

this.searchBookList = data.data;

this.sum = this.searchBookList[0].status || 0;

});

break;

}

case "3": {//著者

getAuthorBook(this.searchItem, this.page).then(data => {

this.searchBookList = data.data;

this.sum = this.searchBookList[0].status || 0;

});

break;

}

case "4": {//ISBN

this.searchBookList = [];

getAllBook(this.searchItem, this.page).then(data => {

this.searchBookList = data.data;

this.sum = this.searchBookList[0].status || 0;

});

break;

}

}

},

searchBtn() {//当前页面检索

this.searchItemShow = this.searchItem;

this.page = 1;

if (this.searchItem == "") {

getAllBook(this.searchItem, this.page).then(data => {

this.searchBookList = data.data;

this.sum = this.searchBookList[0].status;

});

} else {

this.startChanceBook();

}

},

ridioChange(value) {//单选按钮选择回调

this.border = value;

},

appointmentSubmit() {//预约提交

var appid = `${this.appItem.isbn}${this.user}${new Date().getTime().toString()}`;

var apptime = this.timeFilter(new Date().toString());

getAppointment(appid,this.appItem.bookName,this.appItem.isbn,this.studentName,this.user,apptime,this.border).then(data=>{

switch(data.data.msg){

case '0':this.$message.warning('你已经预约该书');break;

case '1':this.$message.warning('排队人数过多不可预约');break;

case '2':this.$message.success('预约排队成功');break;

case '3':this.$message.success('预约成功');break;

case '4':this.$message.warning('该书已下架');break;

case '5':this.$message.warning('送书上门功能已关闭');break;

}

});

},

timeFilter(value) {

let list = [];

let str = "";

list = value.split(" ");

switch (list[1]) {

case "Jan":str = "01";break;

case "Feb":str = "02";break;

case "Mar":str = "03";break;

case "Apr":str = "04";break;

case "May":str = "05";break;

case "Jun":str = "06";break;

case "Jul":str = "07";break;

case "Aug":str = "08";break;

case "Sept":str = "09";break;

case "Oct":str = "10";break;

case "Nov":str = "11";break;

case "Dec":str = "12";break;

}

return `${list[3]}-${str}-${str}`;

}

},

filters: {

selectFil(value) {//4者显示过滤器

switch (value) {

case "1":return "题名(精)";break;

case "2":return "题名(糊)";break;

case "3":return "著者";break;

case "4":return "ISBN";break;

}

}

},

computed: {

count() {

if (this.searchBookList.length == 0) {

return 0;

} else {

return this.sum;

}

}

},

};

</script>

<style lang="less" scoped>

.header {

display: flex;

height: 130px;

}

span {

display: inline-block;

width: 40%;

color: #483d8b;

font-size: 14px;

}

.headBar {

background-color: #5d8fb6;

height: 25px;

font-size: 14px;

line-height: 25px;

color: black;

padding-left: 3px;

span {

width: 70%;

height: 100%;

color: black;

}

}

.headBarItem {

font-size: 15px;

padding: 2px;

}

</style>

## 后端源代码

### 实体类

@Getter

@Setter

Public class TbBook{

private Stirng isbn;

private String bookName;

private String author;

private String press;

private Integer bookDate;

private String bookUrl;

private Integer bookNumber;

private Integer appNumber;

private Integer readNumber;

private String address;

private String msg;

private Integer status;

private Integer type;

}

### Controller层

@Api(description = "图书管理api")  
@Controller  
@CrossOrigin  
@RestController  
public class TbBookController {  
 @Resource  
 private TbBookService tbBookService;  
 @ApiOperation(value = "获取所有图书", notes = "获取所有图书",httpMethod = "GET")  
 @RequestMapping(value = "api/getAllBook",method = RequestMethod.GET)  
 public List<TbBook> selectAllBook(String isbn,Integer start){  
 return tbBookService.selectAllBook(isbn, start);  
 }

@ApiOperation(value = "根据isbn获取图书", notes = "根据isbn获取图书",httpMethod = "GET")  
@RequestMapping(value = "api/getIsbnBook",method = RequestMethod.GET)  
public List<TbBook> selectIsbnBook(String isbn){  
 return tbBookService.selectIsbnBook(isbn);  
}

@ApiOperation(value = "根据bookName精确查询", notes = "根据bookName精确查询",httpMethod = "GET")  
@RequestMapping(value = "api/getBookName1",method = RequestMethod.GET)  
public List<TbBook> selectBookName1(String bookName,Integer start){  
 return tbBookService.selectBookNameBook1(bookName, start);  
}  
  
  
@ApiOperation(value = "根据author查询", notes = "根据author查询",httpMethod = "GET")  
@RequestMapping(value = "api/getAuthorBook",method = RequestMethod.GET)  
public List<TbBook> selectAuthor(String author,Integer start){  
 return tbBookService.selectAuthorBook(author,start);  
}  
  
  
@ApiOperation(value = "根据bookName模糊查询", notes = "根据bookName模糊查询",httpMethod = "GET")  
@RequestMapping(value = "api/getBookName2",method = RequestMethod.GET)  
public List<TbBook> selectBookName2(String bookName,Integer start){  
 return tbBookService.selectBookNameBook2(bookName,start);  
}

}

### Service层

@Service  
public class TbBookServiceImpl implements TbBookService {  
  
 @Resource  
 private TbBookMapper tbBookMapper;  
  
 @Resource  
 private TbAppointmentMapper tbAppointmentMapper;  
  
 @Resource  
 private TbWaitMapper tbWaitMapper;  
  
 /\*\*  
 \* 获取所有的图书  
 \* @return  
 \*/  
 @Override  
 public List<TbBook> selectAllBook(String isbn,Integer start) {  
 List<TbBook> tbBookList=null;  
 if(isbn.equals("")){  
 tbBookList=tbBookMapper.selectAllBook(new RowBounds((start-1)\*10,10));  
 if(tbBookList.size()>=1){  
 tbBookList.get(0).setMsg("1");  
 tbBookList.get(0).setStatus(tbBookMapper.selectAllJiShu().size());  
 }else{  
 TbBook tbBook=new TbBook();  
 tbBook.setMsg("0");  
 tbBook.setStatus(0);  
 tbBookList.add(tbBook);  
 }  
 return tbBookList;  
 }else{  
 tbBookList=tbBookMapper.selectIsbnBook(isbn);  
 if(tbBookList.size()>=1){  
 tbBookList.get(0).setMsg("1");  
 tbBookList.get(0).setStatus(1);  
 }else{  
 TbBook tbBook=new TbBook();  
 tbBook.setMsg("0");  
 tbBook.setStatus(0);  
 tbBookList.add(tbBook);  
 }  
 return tbBookList;  
 }  
 }  
  
 /\*\*  
 \* 添加图书  
 \*/  
 @Override  
 public TbBook insertBook(String isbn,String bookName,String author,String press,int bookDate,int bookNumber,String bookUrl,String address) {  
 TbBook book=new TbBook();  
 if(tbBookMapper.selectIsbnBook(isbn).size()<=0){  
 tbBookMapper.insertBook(isbn, bookName, author, press, bookDate, bookNumber,0,0, bookUrl, address);  
 book.setMsg("添加成功");  
 }else{  
 book.setMsg("添加失败");  
 }  
 return book;  
 }

/\*\*  
 \* 根据书名查询(精确)  
 \* @param bookName  
 \* @return  
 \*/  
@Override  
public List<TbBook> selectBookNameBook1(String bookName,Integer start) {  
 List<TbBook> tbBookList=tbBookMapper.selectBookNameBook1(bookName,new RowBounds((start-1)\*10,10));  
 if(tbBookList.size()<=0){  
  
 return tbBookList;  
 }  
 tbBookList.get(0).setStatus(tbBookMapper.selectBook1Number(bookName).size());  
 return tbBookList;  
}  
  
/\*\*  
 \* 根据author查询  
 \* @param author  
 \* @return  
 \*/  
@Override  
public List<TbBook> selectAuthorBook(String author,Integer start) {  
 List<TbBook> tbBookList=tbBookMapper.selectAuthorBook(author,new RowBounds((start-1)\*10,10));  
 if(tbBookList.size()<=0){  
  
 return tbBookList;  
 }  
 tbBookList.get(0).setStatus(tbBookMapper.selectAuthorNumber(author).size());  
 return tbBookList;  
}  
  
/\*\*  
 \* 根据书名查询(模糊)  
 \* @param bookName  
 \* @return  
 \*/  
@Override  
public List<TbBook> selectBookNameBook2(String bookName,Integer start) {  
 List<TbBook> tbBookList=tbBookMapper.selectBookNameBook2(bookName,new RowBounds((start-1)\*10,10));  
 if(tbBookList.size()<=0){  
 return tbBookList;  
 }  
 tbBookList.get(0).setStatus(tbBookMapper.selectBook2Number(bookName).size());  
 return tbBookList;  
}

}

### Mapper层

@Mapper  
public interface TbBookMapper {  
 List<TbBook> selectIsbnBook(String isbn);  
 List<TbBook> selectAllBook(RowBounds rowBounds);  
 void insertBook(String isbn,String bookName,String author,String press,int bookDate,int bookNumber,int appNumber,int readNumber,String bookUrl,String address);  
 void deleteBook(String isbn);  
 void updateBook(String isbn,String bookName,String author,String press,int bookDate,int bookNumber,String bookUrl,String address);  
 List<TbBook> selectBookNameBook1(String bookName,RowBounds rowBounds);  
 List<TbBook> selectAuthorBook(String author,RowBounds rowBounds);  
 List<TbBook> selectBookNameBook2(String bookName,RowBounds rowBounds);  
 void updateReadLend(String isbn,Integer appNumber,Integer readNumber);  
 void updateBookType(String isbn,Integer type);  
 List<TbBook> selectAllJiShu();  
 List<TbBook> selectBook1Number(String bookName);  
 List<TbBook> selectBook2Number(String bookName);  
 List<TbBook> selectAuthorNumber(String author);  
}

### Mapper映射层

<mapper namespace="com.example.demo.mapper.TbBookMapper">

<select id="selectIsbnBook" resultType="com.example.demo.entity.TbBook">  
 select \* from tb\_book where isbn like #{isbn}  
</select>

<select id="selectBookNameBook1" resultType="com.example.demo.entity.TbBook">  
 select \* from tb\_book where bookName like #{bookName}  
</select>  
  
<select id="selectAuthorBook" resultType="com.example.demo.entity.TbBook">  
 select \* from tb\_book where author like #{author}  
</select>  
  
<select id="selectBookNameBook2" resultType="com.example.demo.entity.TbBook">  
 select \* from tb\_book where bookName like concat('%',#{bookName},'%')  
</select>

</mapper>

# 致 谢

本论文是在导师申情老师的细心指导和严格要求下完成，我在此谨表示衷心的感谢。从课题的选取，系统开发的设计方案到开发完成后的调试排错，无不凝聚着申情老师的心血和汗水。在课堂的学习中，也是始终感受着老师的精心指导和无私的关怀，我受益匪浅。再次向申情老师表示深深的感谢和崇高的敬意。

不积跬步,无以至千里;不积小流,无以成江海。本课题毕业设计能够顺利完成，也多亏了各位任课老师认真负责的教学，让我学到了终伴一生的知识文化和技术，在该系统开发中也多有体现。你们的细心教学给与我很大的帮助，在此向全体老师表示由衷的感谢，感谢你们的辛勤栽培。

我还有感谢我的家人，是家人的支持和鼓励，能让我在这么美好的校园能够全身心的投入到学习中，顺利完成大学学业。最后衷心感谢在百忙之中评阅论文的老师和答辩的各位指导老师！