

基于 Node.js 和 Bootstrap 的图书管理系统设计与研究

陈明雪, 赵勇, 周小丽, 邓思风

(四川托普信息技术职业学院, 四川 成都 611743)

摘 要: 随着高职院校的持续扩招, 图书馆藏书数量不断增加, 这对图书的管理提出了越来越高的要求。通过图书管理设计系统, 可以提高图书管理水平, 实现图书的添加、删除、修改等功能。文章采用 Node.js 技术中的模块化思想完成图书管理系统的功能设计, 应用 Bootstrap 框架完成页面设计, 并从技术介绍、环境搭建、系统功能、数据库设计等几个方面对图书管理系统进行设计与研究。

关键词: 图书管理系统; Node.js; Bootstrap

中图分类号: TP311

文献标识码: A

文章编号: 2096-4706 (2021) 24-0012-04

Design and Research of Book Management System Based on Node.js and Bootstrap

CHEN Mingxue, ZHAO Yong, ZHOU Xiaoli, DENG Sifeng
(Sichuan Top IT Vocational Institute, Chengdu 611743, China)

Abstract: With the durative expansion of higher vocational colleges and ever-increasing number of library books, the management requirements of books are getting higher and higher. Through the book management system, we can improve the level of book management and realize the functions of adding, deleting and modifying books. This paper adopts the modular idea in Node.js technology to complete the functional design of the book management system, applies the Bootstrap framework to complete the page design, and designs and researches the book management system from the aspects of technology introduction, environment construction, system function and database design.

Keywords: book management system; Node.js; Bootstrap

0 引言

近年来, 随着各大高校学生的扩招, 学校的图书馆规模扩大, 图书管理存在越来越多的问题, 如学生借阅过程中不知道藏书是否存在, 在图书馆中寻找是比较麻烦的; 或者需要获取一些电子版的图书相关资料时, 图书馆却没有。这时可以设计一款图书管理系统对图书进行管理。本文采用 Node.js 技术和 Bootstrap 框架设计图书管理系统, 在浏览器端可对图书管理系统进行登录、注册、添加、删除、修改、

查询、上传、下载操作。

1 设计过程

1.1 相关技术介绍

在国内开发者大多数采用 JavaWeb、JavaEE、PHP、C++ 等技术来进行图书管理系统的开发, 以 JavaEE 为例, 现从采用的框架或模块、安装环境、数据库连接、学习兴趣、应用优势等几个方面与 Node.js 技术进行对比阐述, 如表 1 所示。

表 1 JavaEE 与 Node.js 技术对比

Java EE 与 Node.js 技术对比					
技术	采用的框架或模块	安装环境	数据库连接	学习兴趣	优势
JavaEE	Spring、SpringMVC、Mybatis 三大框架	安装环境相对复杂, JavaEE 中需要服务器 Tomcat8.0、Eclipse 开发环境、数据库 Mysql5.5 以上版本、谷歌或者火狐浏览器	进行数据库连接过程中对数据库的版本、引入的 jar 包、数据字段统一性都有严格要求	对大多数学生来说, 学习 JavaEE 课程较为困难, 使用 JavaEE 实现图书管理系统困难较大, 没有足够的成就感, 导致学习兴趣较低	有安全性优势, 一般情况下不能访问底层做一些破坏性编程, 利于各层逻辑的复用、层与层之间的依赖性较低。在代码相同的情况下, 不同电脑可能出现不能正常运行的问题, 这是由于安装的软件版本不同或者版本不统一等问题

续表

Java EE 与 Node.js 技术对比					
技术	采用的框架或模块	安装环境	数据库连接	学习兴趣	优势
Node.js	模块化思想、Express 框架、formidable 模块、art-template 模块、http 模块、fs 模块、url 模块等	安装环境较为简单,只需要 Node.js 最新稳定版安装包	连接数据库操作较为简单	学习 Node.js 相对简单,成就感较高,兴趣较大	实现功能较为简单,Node.js 为前端开发语言 JavaScript 提供了服务端的执行环境,实现了前后端开发语言的统一,开发成本低,完成图书管理系统时间较短

本文采用 Node.js 技术和 Bootstrap 框架设计图书管理系统,在浏览器端对图书管理系统进行登录、注册、添加、删除、修改、查询、上传、下载操作,将发生改变的数据发送给服务器端,再由服务器端反馈给数据库。首先使用 Node.js 模块化思想进行开发,把一个图书管理系统具体到某一个具体功能点(如添加功能点或者删除功能点),把每一个功能点看做一个模块,然后通过相应的规则把这些小的模块组合到一起,构成模块化系统;这种思想的优势之处在于,当某一个功能模块发生改变或者出现问题时不影响其他模块的使用,可实现高效开发、低成本维护。其次采用 Node.js 中的 http 模块实现浏览器客户端和服务端之间的连接,通过 http 模块提供的 listening 方法可以对客户端浏览器进行监听,客户端浏览器和服务端之间可以相互发送数据,一旦操作客户端浏览器发生数据改变,这些数据就会传递给服务端,服务端再将数据反馈给数据库,使用 http 模块提供的 request 方法可以获取客户端发送的数据。其次使用 Express 框架、fs 模块、path 模块、art-template 模块、formidable 模块、url 模块可实现登录、注册、添加、删除、修改、查询、上传、下载功能。

而 Bootstrap 框架提供了一些 CSS 布局、组件(如按钮、下拉菜单、导航、Flex 布局)、Javascript 插件(如轮播图、滚动监听、模态框),便于图书管理系统的前端页面设计。

1.2 环境搭建

采用 JavaEE 需要服务器 Tomcat8.0、Eclipse 开发环境、数据库 Mysql5.5 以上版本、谷歌或者火狐浏览器,系统环境搭建中需要准备 Spring、SpringMVC、Mybatis 的 jar 包和数据库连接的 jar 包等,简言之整个系统准备的 jar 包较多。

而 Node.js 和 Bootstrap 安装环境较为简单,只需要 Node.js 最新稳定版安装包、Bootstrap 版本 4 安装包,Node.js 中所用到的 Express 框架、formidable 模块、art-template 模块、http 模块、fs 模块、url 模块中前面三个需要使用前安装,后面几个在安装好 Node.js 环境后可以直接使用,安装时间很快但需要网络情况较好。

1.3 数据库设计

图书管理系统主要设计的功能是登录、注册、添加、删除、修改、查询、上传、下载,在此过程中涉及用户信息表、图书管理表。用户信息表包含用户名和密码字段如表 2 所示;图书管理表包含 id、书名、主编、图书相关资料、图书封面等字段如表 3 所示。

表 2 用户信息表(customer)

字段名	类型	长度	主键	备注
name	varchar	8	否	用户名
password	varchar	50	否	密码

表 3 图书管理表(book)

字段名	类型	长度	主键	备注
id	int	10	是	编号
book_N	varchar	20	否	书名
author	varchar	20	否	主编
book1	varchar	50	否	学习资料
pic	varchar	100	否	书籍封面

1.4 系统功能

该图书管理系统主要实现用户的图书登录、注册、添加、删除、修改、查询、上传、下载功能。

用户可以在该系统中进行账号注册、使用账号进行登录,当用户需要寻找书籍时可以在该系统中进行查询,在查询图书信息后也可下载该图书的相关电子版资料;当图书馆购置新书时可将新书信息添加在该系统中以供查询,在添加图书信息时也可将图书相关的电子版资料进行上传;当书籍信息发生改变时,如有外借或者其他情况需要修改信息,可在该系统中进行修改;当图书不在馆中时可以将图书信息删除。

下面以图书管理系统的添加模块为例进行详细的叙述。

1.4.1 添加功能实现流程

在图书管理系统首页中点击添加图书按钮,进入添加页面,添加页面中可以添加书名、主编等图书信息,同时可以上传图书相关学习资料、图书封面信息,点击提交按钮,添加数据会发送到服务器,服务器将数据存放在数据库进行保存,具体流程如图 1 所示。

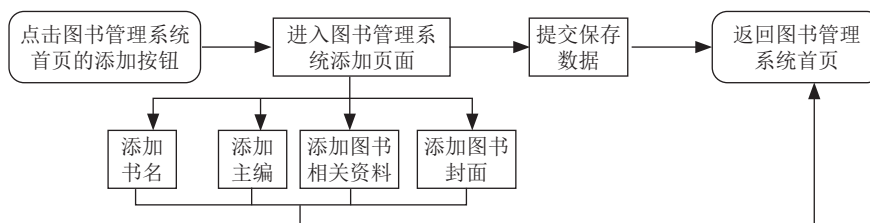


图 1 添加页面实现流程图

1.4.2 首页页面和添加页面部分展示

首页页面的添加功能按钮展示如图 2 所示,在首页页面中通过点击添加图书信息按钮进入到添加页面。图 3 为图书管理系统添加页面,添加页面中可实现添加图书字段信息,包含书名、主编信息、上传图书相关资料、上传添加图书封面信息,信息添加点击提交,数据会保存在数据库,在查询功能中可查到该图书信息。



图 2 图书管理系统首页部分页面展示



图 3 图书管理系统添加页

1.4.3 实现首页和添加页面的部分展示代码

下述代码主要是实现首页中的排版,包含 Bootstrap 框架的 Flex 布局、表单样式、按钮、文字样式。同理添加页面、修改页面也按照 Bootstrap 框架提供的组件、CSS 样式、Javascript 插件进行设计。

```
<div class="d-flex mt-5 align-items-end"> // 使用 Flex 布局排版,实现文字居底
  <div class="display-4 text-success">图书管理系统</div> // text-success 绿色文字样式
  <div class="text-secondary" style="font-size: 30px;">首页</div> // text-secondary 灰色文字样式
</div><hr><br><br>
<a href="/add">
  <button class="btn btn-success">添加图书信息</button> //btn btn-success 绿色按钮样式
</a><br><br>
<form action="/select" method="post" enctype="multipart/form-data"> // 表单样式
  <input type="text" name="id" placeholder="请输入书名" class="form-control"/><br><br>
  <input type="submit" value="查询" class="btn btn-success"/>
</form>
```

1.4.4 实现添加功能的部分展示代码

以下为部分代码, module.exports.showAdd_POST 表示

向外开放方法 showAdd_POST,该方法为自定义方法,该方法中使用 Node.js 提供的 formidable 模块实现添加功能,并将上传的文件信息保存在数据库, form.parse(request, function(err, fields, files) 中使用 fields 保存浏览器客户端发送的字段信息, files 表示保存浏览器客户端发送的文件信息。

```
module.exports.showAdd_POST = function(request, response) {
  var form = new formidable.IncomingForm();
  form.uploadDir = "/upload";
  form.parse(request, function(err, fields, files) {
    var id = 0
    music.forEach(function(item) {
      if (item.id > id) {
        id = item.id
      }
    })
    music.push({
      id: id + 1,
      title: fields.title,
      singer: fields.singer,
      musicN: files.file.originalFilename,
      poster: files.picture.originalFilename,
    })
  })
  index(request, response)
}
```

1.5 软件测试

在图书管理系统设计完成后需进行测试,测试各功能模块是否能够运行,注册中,用户名不超过 8 个字符,密码不能为纯数字;注册后测试账号是否能够正常登录。登录成功进入首页后测试查询、添加、删除、修改功能是否能够正常使用,在首页中测试添加图书按钮是否会进行添加页面,测试删除能够是否正常,点击修改按钮是否能进入修改页面。同时在首页中测试是否能够正常下载文件资料,在添加和修改页面中测试是否能够上传资料。根据不断地测试和不断的修改最终成功实现图书管理系统的各部分功模块,符合用户的需求,可以正常使用。

在测试过程中也发现一些问题,如在查询页面点击添加图书按钮无法进入到添加页面,这是由于在对客户端浏览器进行数据反馈判断时,对 url 地址判断错误或者 http 请求方式判断错误,正确的代码应该是 if (request.url === "/add" && request.method === "GET")。

2 结果分析

使用 Node.js 的 Express 框架、formidable 模块、art-template 模块等可成功的实现图书管理系统的登录、注册、添加、删除、修改、查询、上传、下载等功能;在环境搭建过程中配置 Node.js 环境需注意要为 Node.js 配

置环境变量;就各个实现功能而言,注册功能实现了用户名长度不超过7位,密码不能为纯数字;查询功能中,成功的完成了对书名的查询,在查询功能中成功地实现了下载图书的相关资料;添加功能中,成功实现了图书信息的添加以及图书资料的上传;修改功能中,成功地实现了对图书信息的修改;删除功能中,成功实现了对图书信息的删除。

3 结 论

本文主要讲述基于Node.js和Bootstrap两种技术设计图书管理系统,采用Node.js技术中的模块化思想、Express框架、formidable模块、fs模块、Path模块等完成设计,同时应用Bootstrap框架技术完成图书管理系统的页面设计。详细介绍使用Node.js和Bootstrap进行设计的步骤,主要包含环境搭建、系统功能、数据库设计,以及选择其中一个添加功能模块进行详细叙述,同时将现有的一些技术如JavaEE与Node.js进行对比发现,对于学生来说,Node.js技术学习难度较低,安装的环境也较为简单,当页面请求数据较大时,

Node.js依然能够轻松实现;而JavaEE技术引用的jar包较多,安装环境较为复杂,很多时候由于安装环境的问题导致即使代码没有问题依然无法运行,所以学生学习Node.js技术相对简单,成就感较高,兴趣较大。

参考文献:

- [1] 张慧娥.基于SPC与Bootstrap方法的高校教学质量评价的研究[D].西安:西安电子科技大学,2011.
- [2] 王崇杰,袁露,金托弟,等.基于vuejs和nodejs的换书软件设计与实现[J].电脑迷,2018(33):115.
- [3] 李博洋.基于Node.js的分布式数字资源开放服务系统的设计与实现[D].北京:北京邮电大学,2014.
- [4] 黄扬子.基于NodeJS平台搭建REST风格Web服务[J].无线互联科技,2015(16):57-59.
- [5] 戎玫,张广泉,刘艳.基于软件体系结构和UML的图书管理系统设计与实现[J].计算机科学,2005(6):224-227.
- [6] 王立萍.基于B/S的图书管理系统的设计与开发[J].承德民族师专学报,2004(2):27-28.

作者简介:陈明雪(1991—),女,汉族,四川成都人,电子计算机工程师,助教,硕士研究生,研究方向:计算机技术。

(上接11页)

4 结 论

对于企业来说,客户是企业的基石,负责管理客户信息的CRM系统就是企业发展的重要支撑。CRM系统的设计目标是减少用户的交互行为,使他们能够用更短的时间去做更多的事,这是CRM系统设计的关键所在。为了有效解决这个问题,我们研究了许多市场上的产品,不断加强与客户沟通,不断了解客户需求,并不断对产品及服务进行改进和提高,以满足客户日益变化的多样化需求,最终设计出当前的版本。

由于时间和技术有限,系统本身还是存在许多问题有待解决。我国CRM系统发展仍处于巩固阶段,随着企业对CRM系统的成熟运用,CRM系统会逐渐被更多的企业认可,所以还有很漫长的路要走。希望我们的系统也能与时俱进,随着用户随时随地查看信息需求的增长,未来移动端CRM管

理系统将成为发展的又一重点,我们定会加强研究、不断进取。

参考文献:

- [1] 李立.CRM客户管理系统的优势和发展研究[J].中小企业管理与科技(下旬刊),2020(6):30-31.
- [2] 赵蕊斐.新疆电信新一代客户关系管理系统建设项目管理研究[C].南京:南京邮电大学,2019.
- [3] 潘冬宁.中小企业使用CRM客户管理系统存在的问题及对策[J].商场现代化,2021(10):76-78.
- [4] 林静.客户关系管理的文献综述及发展趋势[J].智富时代,2018(1):108.
- [5] 刘文瑛,吴冠男.基于SSM的客户关系管理系统分析[J].产业科技创新,2020,2(8):101-102.

作者简介:生世龙(2000.02—),男,汉族,山东平度人,本科在读,研究方向:服务器开发;王志涛(1985.09—),男,汉族,山东武城人,高级工程师,本科,研究方向:数据库设计;赵志民(2001.12—),男,汉族,山东汶上人,本科在读,研究方向:前端设计。