真实姓名：王先坤

用户名：141341690465005

所属服务站：潞安职业技术培训学校奥鹏远程教育学习中心

指导教师：谭智勇

**图书销售管理信息系统设计**

【摘要】：

随着互联网的蓬勃发展，传统的线下图书销售模式已经满足不了当今社会更加庞大的供给需求，线上图书销售体系慢慢地成为趋势，因此对于线上图书销售管理信息系统的设计与开发具有很强的现实意义。本文设计与开发是以图书销售系统为中心，整理线下销售体系需求并参考当今电商平台的特点，利用JAVA语言和HTML网页技术，通过RestFull前后分离的设计思想，基于 MySQL数据库之上建立了一套线上销售与后台管理流程。此业务流程分为用户系统，图书管理系统，订单系统以及售后反馈系统，通过以上几个系统，对线下图书功能总结，实现了线上销售的信息化，敏捷化。其对于提高图书销售管理水平有了极好的促进作用。

**【关键词】：**图书销售管理系统、前后端分离、Java、MySQL、RESTfull API

【正文】：

一、绪论

（一）研究背景

随着人们生活水平的提高，人类也进入了信息化社会，网上销售体系早已经建立起来，从PC,平台电脑，智能手机到穿戴设备，各种新的电子产品已经慢慢融入到我们的生活中，去图书馆看书，在图书城购书早已经不是现代人的需求，人们只需要按按鼠标，点点屏幕就能买到自己需要的商品。当下传统的线下书店也慢慢会走出历史舞台，新时代网络图书销售模式也是时代所趋。

图书具有易于包装，不易过期，方便运输的等特点。而且图书在印刷上质量得到保障的情况后，售后方面的纠纷也会相应更少些，其于这个些特点，其非常适合这种线上销售模式。

（二）研究意义

通过实现线上图书销售体系，能够改现在老套的销售模式，实现技术创新，通过高速的互联网推广自己的产品，并且减少销售流程，通过电子信息技术，可以较大规模的增大商品库存量并无太大成本，高效的后台系统操作方便，不仅可以节省人力成本，还能提高图书销售效率。网上销售系统可以有选择的制定优惠政策提高图书的销售竞争力和，提高利润，创造更大的价值。

二、相关理论概述

（一）Java技术

Java 是由 Sun Microsystems 公司于1995年5月推出的高级程序设计语言，Java可运行于多个平台，如 Windows, Mac OS及其他多种UNIX版本的系统，具有很高的可扩展性和可移值性

（二）MySQL数据库

MySQL，它是瑞典公司开发的一个款小型的关系型数据库管理系统。MYSQL被广泛应用于WEB程序开发中。其速度快、体积小、成本低，并且是开放源码的。

（三）B/S架构

B/S，即浏览器/服务器架构模式，他是Web互联网兴起后的一个典型代表，Web浏览器为其主要客户端，主要代表有火狐和谷歌和微软的IE。服务器则以JAVA为代表，随便着2016年Google发布了新的浏览器引擎V8，更多有服务器架构也慢慢涌现。本文的图书销售系统是以JAVA技术为核心，并且J2EE框架作为平台体系，基本于JRE运行环境的软件系统。

（四）RESTfull API概念

REST，表示性状态转移（representation state transfer）。简单来说，就是用URI表示资源，用HTTP方法(GET, POST, PUT, DELETE)表征对这些资源的操作，RESTful API是目前比较成熟的一套互联网应用程序的API设计理论。

（五）HTML和Jquery

超文本标记语言（英语：HyperText Markup Language，简称：HTML）是一种用于创建网页的标准标记语言，本系统后续提及的前台和后台界面，都是通过此语言实现的。Jquery是一个操作HTML的功能辅助框架，文中提及较少。

三、系统需求分析

（一）系统功能性需求

对于线下的图书消费流程，简概如下：用户选书，用户下单结帐，图书出库。通对线下图书商店的销售逻辑理解，我们将线上销售体系会有两大平台，分别为前台和后台，前台实现店铺的基本功能，如用户选书，用户下单等功能，其它包括用户操作导航，用户登录与注册，新书展示，销售排行，下单购买等。后台实现图书出库功能，即用户下单确定后，给用户发货。除此之后，后台还要实现图书，用户，订单的运营，即管理书集，新书上架，用户权限，还有订单处理等等。当然图书管理系统还有最重要的业务，就是要实现用户级别。我们将用户分为三类，即普通用户，注册用户，和管理员，不同的用户在前台和后台两大平台，呈现的功能也会有所不同。比如普通用户只能浏览前台系统，只有注册用户才有购买图书的权限，当然更高级别的管理员会拥有所有权限。

1. 前台功能需求

前台功能功能主要有导航、用户登录与注册、书库、用户订单、购物车，还会辅助支持个性化推荐，新书推荐，搜索等。辅助模块用来提高用户的购书体验。

导航模块，它是我们书店的网站地图，它让用户很直观的了解图书店销售系统的主要业务内容，方便引导用户购书。导航上有“首页”，“书库”，“购物车”，“我的订单”，“用户登录与注册”，“售后”几个子模块。

首页，顾名思义，当用户在操作中找不到方向，可以直接通过首页功能回到网站初始状态，首页是图书商店的入口，可以方便找到用户喜好的图书，其中包括很多子模块，如新书推荐，售销排行榜等

书库，这里用户查书线上图书仓库里所有可以售卖的图书，支持书名，分类，发行时间等不同维度的搜索方式，方便用户以最短的时间找到自己喜欢的图书。

购物车，此模块用户展示已经下单的图书列表，用户可以随时进行批量下单并支付，只有注册用户才有此功能

订单列表，此功能用户展示用户已经支付的订单，用于查看所有购买的书籍。

用户登录与注册，此功能是个动态模块，当用户注册并登录后，只会显示当前登录的用户信息，用户未登录时，会引导用户注册或登录书店，并进行购买图书。

售后，此模块为反馈模块，方便用户在操作异常，或者有购买疑问时，通过邮件的方式通知管理员，是一个用户与管理员沟通的入口。

2.后台管理功能需求

图书管理，此为系统的重点需求，支持对图书入库，下架和编辑等操作，并支持不同维度的搜索，便于管理员方便的管理书籍。

用户管理，首先它于给管理员对注册用户提查找与浏览用户信息，其次是对用户进行管理操作，如屏蔽用户，但管理员只能修改它状态，不能修改其基本信息，以保护用户隐私，管理员可以配合搜索，很快能找到异常用户，对于冻结。

订单管理，用来管理员管理用户生成的购买的图书清单，并对清单用效性进行管理，可以修改订单的状态或者拒绝订单，并通知购买用户。同时能够搜索订单，找到异常订单等。但是管理员不能删除用户订单信息。

反馈管理，这个是用于查看用户反馈建议等，相当于一个收件箱，收集用户反馈的信息。

通过线上其它平台的理解，我们的后台管理系统支持管理权限控制，管理员在几上管理模块中操作，必须要经过管理登录入口授权后才可以操作。当然还有注销模块，方便管理员清空授权信息，以保障系统的安全性，防止非法用户进来操作。

（二）其它需求

1.兼容性

根据本系统客户端是Web浏览器的特点，考虑当今环境不同内核的浏览器引擎层出不穷，不同浏览器引擎对Web内容渲染展示上会有各种细节上的不同，我们的系统要在不同浏览器引擎上要求有一致的界面表现性，功能的正确性。

2.安全性

系统的安全性主要体现在用户安全，系统安全，以及数据安全。所以系统的设计应当保证用户的隐私信息安全，不可外流，还要保证商店信息的安全性，不被同类商品竞争者盗取，除此之外，还要防止不用户权限的越级使用，避免非常用户对图书内部资料的破坏，保障系统的操作安全。最后是数据库的稳定性，确保数据库异常后能够快速恢复。

3.稳定性

稳定性表现在高并发场景下，平台服务也能够正常运作，数据被意外破坏或攻击时能够快速恢复。此特性是评估产品的一个很重要的标准。

4.可移植性

可移值性可以提高软件的生命周期，是软件的一个增强，即在不修改太多配置的情况下，能够将软件运行在其它系统。

（三）系统设计可行性分析

1.技术可行性

按照该类子系统总体目标来衡量，一般能从计算机硬件以及软件的使用性能要求、大坏境前提、操作管理人员技术水平以及数量等方面去顾全以及分析。考顾及到系统实施的适用性及复杂性，在软件方面我们选用了Java技术手段来展开开发管理服务平台的设计，并使用SQLServer2012数据仓库存储数据资料，客户端层面使用非常普及的HTML技术，以及各种开源框架前端框架如jquery,bootstrap运作在种大电商平台，Web兼容性问题基本也不再存在。架构方面采用RESTfull前后端分离思想。同时在硬件方面，可供抉择的空问也很大，硬件的配置越高系统运行会快，更佳流畅。于此同时，我们考虑到当今的软理件的普及程充以及家用或者商用电脑硬件的整体软硬件配置水平，此系统在当前环境下是可行的。Java技术和SQLserver2012技术经过十多年的发展，经历互联网的考验，无论从安全性、可用性或者稳定性上来说，都是毋庸置疑的，RESTfull架构在平板，手机等各种API上层出不穷，因此，软件和硬件上都是可行的。

2.经济可行性

管理软件的的经济可行性，主要体现在生产中产生的经济效益成本与开发设计所需要的人力以及投资成本相比，是否相适宜，不仅如此，我们不光要看商业效益，还要看软件能否给用户带来更多的经济效益和使用成本。我们开发的这套系统就是为了解决客户和商家的问题，从在线销售系统的使用者或者图书店主来说，它不仅难提高工作效率，节省开销，还能方便操作，能够应付未来的更多多变的需求。本系统从开发成本来说，只是个人的独立设计，可以省略很多成本费用，同时也要提高个人动手能力，也能辅助其它平台，得到技术上的提升。对于客户来说，增加购书的便捷性，还有可能得到图书销售量数据，购书也不再盲目。

3.操作可行性

互联网的普及，Internet网络互联，这种新型的产业形式天天在影响着我们的工作和生活。不仅如此多企业为了扩大他们的生产规模，不仅将他们的市场遍布国内，同时也影响着部分国外市场，市场越大，客户越多，面临的线下销售模式早已满足不了他们的日益增长的需求。在线销售图书对于这种大范围的客户管理，有着无以伦比的优势，无论在国内网，多大的客户，只要网络传输到的地方，都能很快的解决问题。

在线图书管理销售系统，使用简单，不需要做太多的技术培训也能够迅速容易上手，加上模块的精细化，类别化，以及高度可扩展化，方便迎接未来的挑战，不仅对商业，还是用户都是很方便的。经上所述，从实施上来看也是完全可行的。

四、系统概要设计

（一）系统整体架构设计

考虑到项目的可移植性与可扩展性，我们采用前后台分离的技术架构，后端目前采用JAVA语言和MySQL数据库提供数据服务，前端采用HTML和Jquery图形操作框架，关系层用JSON数据格式定义前后端数据接口规范。在保持接口规范不变的情况下，可以随时升级前端和后端的技术类型。比如前端可以升级为交互更强的Vue和React界面框架，后端可以升级为时下更为火热的NodeJS等。具体结构如图4.1所示：

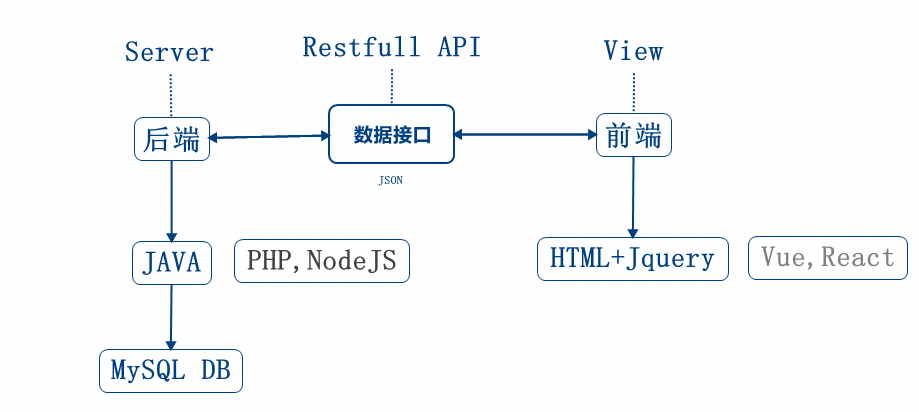


图4.1前后端架构图

在线图书销售管理系统分为前台和后台两大体系，每个体系的功能如图4.2所示：

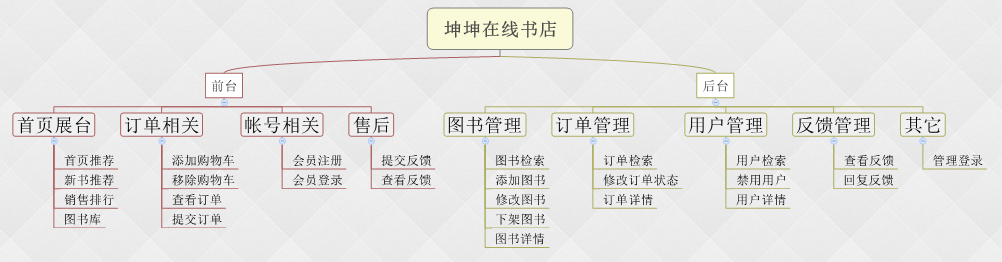


图4.2前台和后台功能结构图

（二）系统数据库设计

1.数据库结构化设计

通过对线下图书商店的业务逻辑分析，我们可以将业务实体简化为三个，即用户，图书，订单。一个用户可以产生一个订单或多个订单，所以他们的关系是一对多关系。一个订单可以包含多本图书，所以图书与订单是多对多的关系。因此我们建立两个基本数据表用于存储图书和订单，它们通过图书ID和订单ID建立关联。对于参与图书与订单关互的实体用户，也需要建张数据表用于存储其基本信息。对于线下我们将用户会分购买者和店长两类，对于网上书店我们需要进一步划分，我们将用户分为购书用户，普通的顾客以及管理员用户三类，通过用户表中的状态字段标识用户级别，只有购书用户才真正参与订单流程。

建立了抽象的数据实体后，我们下一步进行实体细化，针对图书的特点，它包括的基本信息有书名，作者，发行商，发行时间，价格，封面图等基本信息，还有必要增加下架状态和新书状态以及购买量信息方便管理员对图书下架和推荐操作。订单实体会用订单号，以及与订单相关联的用户信息，还有与之关联的图书信息，当然还包括订单的属性，如联系人，联系电话，收件地址等。最后是用户实体，除了上述的用户级别状态，还应该包括用户唯一Id标识，邮箱，联系电话和联系地址等。除此之外，我们还建立了售后数据块，用它建立顾客和店主用户间的关系，方便两间间接的沟通与交流。

综上分析，我们的系统明确了三大结构实体，即图书，订单以及用户，并对每个类型实体进行了简要的细节设计，并辅助分析了售后实体，所以上结构体的关系E-R图如4.3所示：

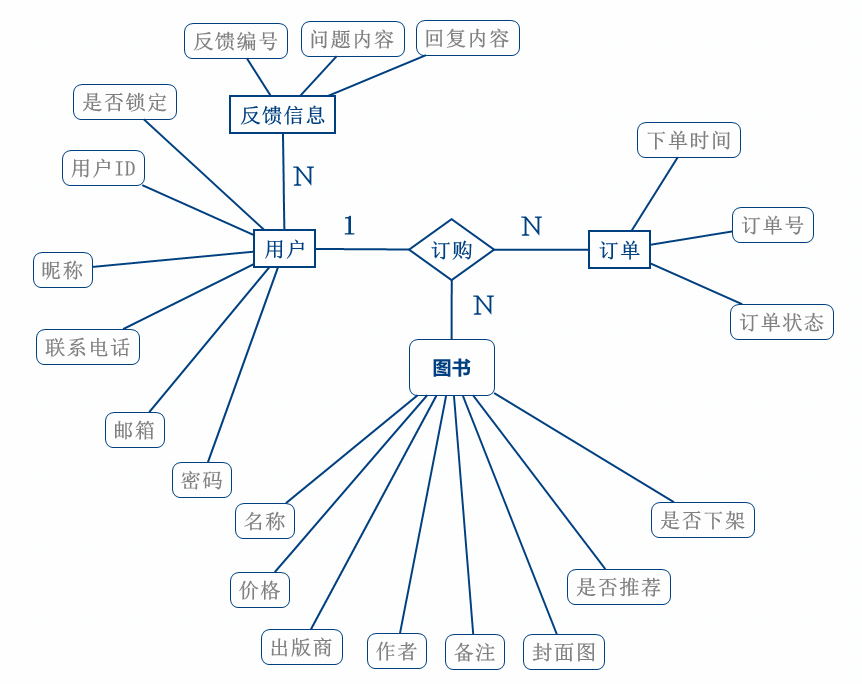


图4.3实体关系E-R图

2.数据库表的设计

首选打开并启动SQL Server2012服务，通过Navicat的图形界面连接数据库，并新建一个数据库名为db\_shops。根据上述实休关系图，我们先创建三张表，分别名为：db\_books, db\_users, db\_order，db\_books为图书表，存用于存储图书基本信息，db\_users为用户表，用于存储注册用户信息和管理员信息，db\_order为订单表，用于存储用户下的订单信息。辅助建立db\_feedback表用来存储用户反馈建议信息。具体的表内容字段如下。

表4.1图书信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 是否允许为空 | 备注 |
| Book\_id | varchar | 10 | 否 | 主键，唯一ID |
| Book\_isbn | varchar | 10 | 否 | ISBN编号 |
| Book\_name | varchar | 20 | 否 | 名称 |
| Book\_type | varchar | 20 | 否 | 分类 |
| Book\_publisher | varchar | 20 | 否 | 图书出版社 |
| Book\_author | varchar | 20 | 否 | 图书作者 |
| Book\_intro | text | 16 | 是 | 说明 |
| Book\_price | money | 8 | 否 | 单价 |
| Book\_publishdate | varchar | 10 | 是 | 发行日期 |
| Book\_pic | varchar | 30 | 是 | 封面图 |
| Book\_count | Int | 4 | 否 | 库存数量 |
| Book\_createdate | datetime | 8 | 否 | 记录创建日期 |
| Book\_isnew | Int | 4 | 否 | 是否是新书 |
| Book\_isrecommend | Int | 4 | 是 | 是否推荐 |
| Book\_isdeleted | Int | 4 | 是 | 是否已经删除、下架 |

表4.2用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 是否允许为空 | 备注 |
| User\_id | Int | 10 | 否 | 主键，唯一ID |
| User\_name | varchar | 18 | 否 | 用户名 |
| User\_level | Int | 4 | 否 | 用户级别标识,0为超管，1为普通用户，2,为登录用户 |
| User\_nickname | varchar | 18 | 否 | 真实姓名 |
| User\_pass | varchar | 18 | 否 | 密码 |
| User\_address | text | 60 | 否 | 收件地址 |
| User\_sex | Int | 6 | 是 | 性别 |
| User\_telphone | varchar | 12 | 是 | 联系电话 |
| User\_email | varchar | 30 | 否 | 邮件 |
| User\_locked | Int | 4 | 是 | 是否被禁用 |

表4.3订单信息主表

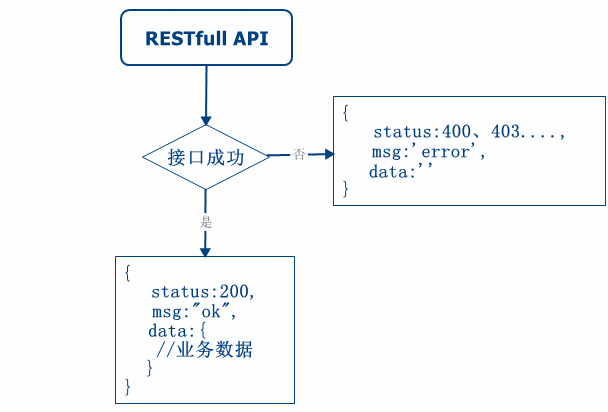
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 是否允许为空 | 备注 |
| Order\_id | Bigint | 4 | 否 | 订单唯一ID,主键 |
| User\_id | Int | 10 | 否 | 用户ID |
| Order\_shops | Varchar | 500 | 否 | 与订单关联的书集描述信息 |
| Order\_address | varchar | 60 | 否 | 订单收件地址 |
| Order\_nickname | varchar | 18 | 否 | 订单收件人 |
| Order\_tel | varchar | 12 | 否 | 订单联系人电话 |
| Order\_createdate | varchar | 20 | 否 | 订单创建时间 |
| Order\_state | Int | 4 | 否 | 订单状态 |
| Order\_desc | varchar | 200 | 否 | 订单备注信息 |
| Order\_price | money | 8 | 否 | 总价 |

表4.6用户反馈信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 是否允许为空 | 备注 |
| Feedback\_id | VarChar | 4 | 否 | 主键 |
| Feedback\_content | Varchar | 500 | 否 | 反馈内容 |
| Feedback\_reply | Varchar | 500 | 是 | 反馈回复内容 |
| Feedback\_createdate | datetime | 8 | 是 | 创建日期 |
| Feedback\_isdeleted | Int | 4 | 是 | 是否删除 |
| Feedback\_nickname | VarChar | 18 | 是 | 反馈方的昵称 |
| Feedback\_telphone | varchar | 12 | 否 | 订单联系人电话 |
| User\_id | Int | 10 | 是 | 反馈作者id |

（三）RESTfull API的设计

RESTfull 的响应数据都通过JSON格式表达，无论响应成功或失败，都会返回完整的JSON数据，对于成功会以status为200响应，对于失败状态，status会有不同的状态，比如服务器错误为400,用户权限错误为403等。前台和后台通过JSON数据标准完成数据交互，所以一定要确保后台输出的JSON数据准确性。RESTfull API的结构如图4.4:

图4.4　RESTfull API结构标准

前台和后台具体的API内容如下：

1.前台的Restfull API

前台API主要定义的接口有商品查询，订单查询，反馈信息查询，用户下单，用用户登录，用户注册等，具体定义如下：

商品展示相关接口

（1）推荐图书‘get/’‘/web/book/suggest’

（2）销售排行榜‘get/’‘/web/book/top/’

（3）图书库查询‘get/’‘/web/book/all/{$name}&{$publish\_date}&{$type}

用户相关接口

用户注册‘post/’‘/web/user/register/’

参数: user\_name&password&password\_confirm&email&address&telphone

响应：{status:200,user\_id, token}

用户登录‘post/’‘/web/user/login’

参数:user\_name&password

响应：{status:200,user\_id, token}

订单相关接口

（1）查看用户所有订单‘get/’‘/web/order/list/{$name}&{$order\_id}’

说明：支持按商品名称和订单号查询

响应：{status:200,list:[] }

（2）用户下单‘post/ ‘‘web/order/create/{$shop\_ids}&{$shop\_counts}’

说明：支持传和多个图书以及每本图书的数量

响应：{status:200,order\_id }

其它接口

（1）用户反馈　‘post/’ ‘web/feedback/{$content}&{$user\_id}&{$token}

说明：注册用户和普通用户都可以反馈信息给管理员

（2）查看反馈‘get/’‘web/feedback/list’

2.后台主要RestAPI

后台所有的API接口都要传入user\_id和token以验证用户的权限，只有权限值（level为0）为管理员时，RestAPI才有数据返回，否则返回错误，并跳至后台登录页。

商品相关管理接口

（1）添加商品‘post/’‘admin/book/add/{$name}&{$price}……{$user\_id}&{$token}’

（2）删除商品　‘post/’‘admin/book/delete/{$book\_id}……{$user\_id}&{$token}’

（3）修改商品　‘post/’‘admin/book/modify/{$book\_id}&{$name}&{$price}……{$user\_id}&{$token}’

（4）搜索商品　‘get/’ ‘admin/book/list/{$name}&{$price}……{$user\_id}&{$token}’

（5）推荐商品至首页‘post/’ ‘admin/book/modify/{$book\_id}&suggest=1&{$user\_id}&{$token}’

订单相关管理接口

（1）查看所有订单　‘get/’ ‘admin/order/list/{$name}&{$order\_id}……{$user\_id}&{$token}’

（2）确认订单并发货‘post’‘admin/order/mody/status=1’

说明：提交订单状态为2标记为待出库（0:未处理，4:拒绝，1:完成交易）

（4）拒绝订单　‘post’‘admin/order/mody/status=4’

用户相关接口

（1）查询所有用户　‘get/’ ‘admin/user/list/{$name}&{$user\_id}

（2）修改用户状态（禁用、开启）‘get/’ ‘admin/user/modify/{$user\_id}&{$status }

（3）其它接口

（4）管理员登录‘post/’ ‘admin/user/login{$name}&{$password}’

（5）管理员注销‘post/’ ‘admin/user/logout’

其它接口

（1）查看反馈信息‘get/’‘admin/feedback/list’

（2）删除反馈信息‘post/’‘admin/feedback/delete/{$feedback\_id}

（3）管理员回复用户反馈‘post/’‘admin/feedback/reply?content

五、系统详细设计及实现

（一）系统业务流程设计

1.前台基本业务流程

前台的基本业务是实现丰富的图书内容展示，并按照不同维度的展现方式吸引用户购买图书，同时提供便捷的交互方式简化购买流程，方便用户选购。

前台系统主要分以下几个功能模块：

（1）首页内容模块

首页内容模块是在线图书销售系统的入口，它罗列了网站导航，所有图书，新书推荐，销售排行，提供了用户登录和注册，售后入口。

（2）新书模块，它是一个自动运营模块，系统通过发布时间信息，自动化展示的数据内容，无需维护的数据，可以方便用户快速了解最新动态

（3）销售排行模块，此模块和新书模块类似，也是一个自动运营模块，系统会根据图书的销售量，将销售量靠前的书集展示出来，销售排行对购书者购书提供了有益的参考，同时吸引消费者。

（4）书库模块，它是在管理系统的图书仓库，所有未下架的图书都可以在这个模块找到。目前支持通过书名，出版社，图书作者等多个维度检索，还支持分类检索，缩小检索范围，方面购书者在短时间内找到自己想要的书籍，用户可以通过检索到的结果，随时将找到的书添加至购物车。

（5）登录与注册模块，主要是用户进入网站的入口，只有登录用户才有购书的权利。所用用户参于购，必须先要注册成为网站会员才可以，建立起用户关系，然后登录后进行用户与图书的销售关系绑定，才能完成订单的生成。对于用户特征信息登录后，管理员是无权修改的，管理员只能对期进行锁定操作。当然，对于注册用户，管理员通过了后才可以使用，网站才能为他提供服务。与此同时，用户要真正完成订单流程，填写正确的联系方式也是非常有必要的，不然是无法进行售后收货的。

（6）购物车模块，是方便用户购物的一个辅助工具，过去的网购只能对一件商品下单，有了购物车，可以随便将看到的商品放入车中，并能够随时从车中移除，大大方便了用户。同时通过将车中所有的书籍一并发放至收银行台，一并结帐，简化了很多操作流程。在结帐前，还可以一键清空购物车，为下次重新购物做准备。

（7）我的订单模块，它用来方便用户随时查看历史订单，也可以方便用用户跟踪正在进行的订单，了解订单的处理进展。订单模块支持按订单号维度搜索订单。订单是结帐的凭证，在购物车下单后，即生成购物商品的唯一清单信息。

（8）订单结算，它是一最重要的模块，以上所有模块都是为此功能服务，当用户将图书添加至购车，通过“结算”进入此流程。如图：5.1

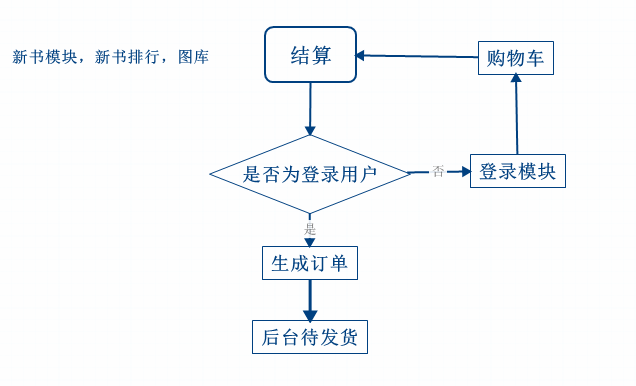


图5.1订单结算的流程图

2.后台基本业务流程

后台的基本流程基本是实现对图书，用户，订单的数据管理，进行基本信息的入库和修改，维持前台销售平台数据高度可用性，提高线上书城的销售量。

后台主要分以下功能划块：

（1）图书管理模块，此为在线图书销售系统的最为核心一部分，其功能包括新书的入库，图书的基础信息的编辑（如封面，书名，版号等等），书的下架，精编推荐等功能，此模块还支持搜索功能，如按名书，ISBN，入库时间范围等，方便图书管理员的日常管理。

（2）用户管理模块，此模块主要用于维户管理系统的用户数据，同时支持用户权限锁定，对于异常用户给予禁止访问，以免干扰图书销售平台的正常次序。用户管理模块同时支持搜索，方便管理员快速定位用户。

（3）订单管理模块，它主要是用于管理员查看和修改订单状态，然后将下单状态反馈给购书的用户。但是图书管理员绝对不允许修改订单的基础信息，否则未导致不可预估的后果。

（4）售后管理械模块，它是一个用户与管理员交互的模块，方便用户与管理员间接的通过平台进行问题沟通，解决用户的问题。

（5）登录模块，其实它是一个不可见的中间模块，每当管理员发送接口请求时，都要校验user\_id和token信息，一但校验不是管理员身分，系统自动跳转至登录模块，要求进行权限认证，认证通过后才可以进行后续操作，当然如果通过前台系统登录后，如果是管理员身份，也可以直接通过前台进行后台，而不必要再次经过后台登录模块登录。

（6）退出模块，主要是清空登录权限，防止陌生人误操作造成安全隐患。

后台业务流程如图5.2所示：

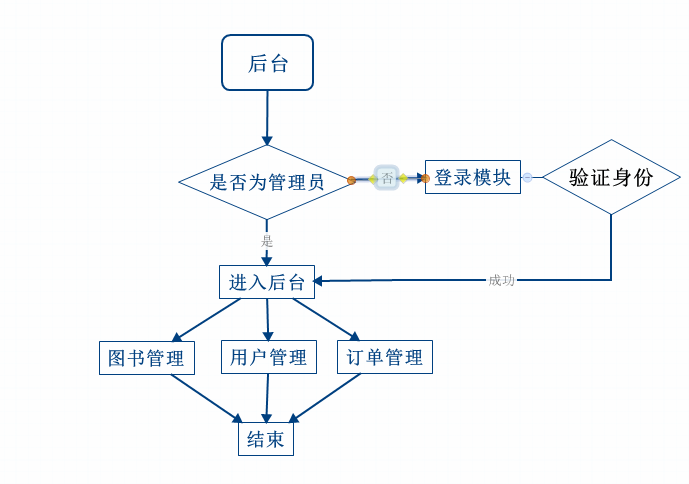


图5.2后台业务流程图

（二）系统前台功能及模块的实现

1.首页模块模块

首页是一个站点地图，它有导航，提供登录注册，订单，购物车等二级入口，还有推荐图书，热点图功能。如图5.3所示：



图5.3前台首页模块

2.用户注册与登录页

通过对其它电商平台的分析，我们的销售平台也有必要支持帐号系统，所有注册和登录环境节是必不可少的。普通用户只有浏览查阅图书的能力，只有注册会员并登录前台系统，才可以进行后续的购买操作，当然管理员也为用户的一部分，管理员拥有最高权限。

用户登录界面如图5.4，登录流程如图5.5：

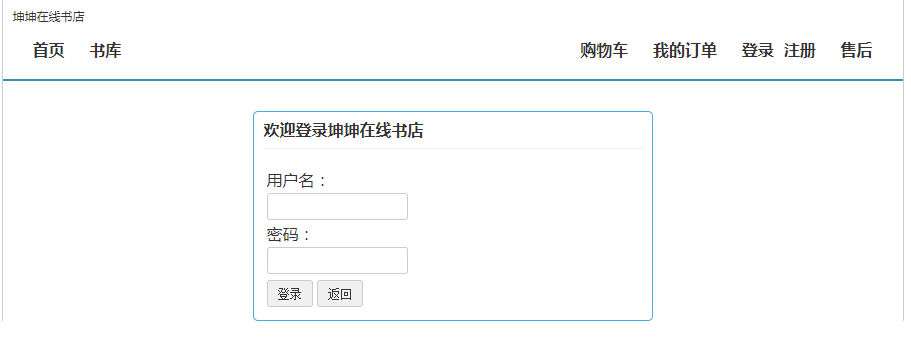


图5.4前台登录模块

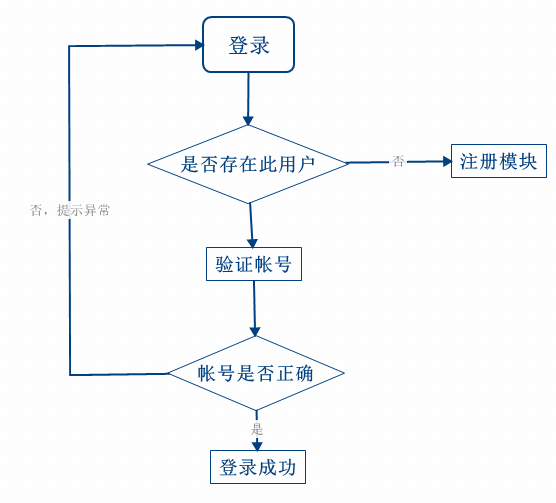


图5.5前台登录模块流程图

当管理员用户登录时，登录后还要验证是否为管理员，如果为管理员时，网站导航栏还要显示管理员入口，方便管理员直接进入，省去从管理登录页登录的麻烦，如图5.6：



图5.6前台导航登录态

注册模块功能和登录流程类似，注册前判读数据库是否存在同用户名，如果存在，则跳转登录，否则进行后续验证，表单信息填写正确后，则完成注册并跳至网站首页。

如下图：



图5.7前台注册模块

3.购物车模块

购物车模块是个临时数据模块，当用户进行选书时，将用户选的书的信息记录在此数据区，它不依赖后台API，数据全是通过前端JS脚本存储在本地浏览器缓存中。用户可以有选择的清空选择的项目，也可以清除所有项目，这个和线下购物有些同样的操作体验。用户确定好要购买的图书，直接点击“结算”后提交并生成订单，同时从购物车清空已经购选的书集。



图5.8购物车模块

4.我的订单模块

我的订单询模块，此功能此模块是为了方便用户查看订单信息的执行状态，用户通过购物车结算并下单后，所产生的订单信息会展示在此模块中，当然历史订单也会这此模块展示。用户可以根据查看订单的状态，了解管理员的处理进展。订单模块支持按照订单号搜索订单，界面如图5.9：。

图5.9前台订单模块

5.书库模块

此模块前文已经简要提及，当会员在首页无法找到喜欢的图书时，可以在图书库中检索出自己喜欢的图书。用可以从首页导航“书库”，或者从首页的图书“搜索”框中搜索进入。书库右方有图书分类，用户可以根据自己的兴趣爱好进行分类查找，然后通过“书名”，“作者”等不同维度抽像搜索，精确的找到自己喜欢的图书。当用户查到自己喜爱的图书，可以随时通过检索结果中的“购买”按纽进行下单添加至购物车，并进行结算。

图5.10前台书库模块

（三）系统后台功能及模块的实现

1.图书管理模块

图书管理模块是此系统最为复杂的模块，它除了要实现图书的基本功能，如新书入库，书籍编辑，旧书下架，还要支持图书推荐，是否新书等，同时支持图书按照书名，日期，ISBN等多维度的条件搜索。

（1）图书信息的添加

点击操作添加图书信息时，首先要确保入库书集不能重复，这个就要首先判断输入的图书唯一代号（即ISBN号）在数据库中是不能够重复，即是唯一的，当用户输入了重复的ISBN信息时，界面要相应的给予错误提示，告之管理员已存在同名书籍，并重新修改后再提交。但唯一的图书入库后，回到图书列表页，刷新列表，显示最新入库的书籍，继续等待新书入库。

如图5.11所示：



图5.11后台图书管理模块

（2）修改图书信息

修改书籍，业务功能基本同添加模块，也要进行入库书籍标识鉴别，如果库中不存在的书籍也应该提醒管理员，不允许修改。具体流程，首先要确定和判断ISBN的参数值，当ISBN的参数值不为空时，就通过ISBN查询图书的详细信息，并将相应的信息显示显示在我们规定好的表单里，对于普通数据直接通过文本框显示，对于boolean类的数据进行逻辑处理，显示单选框，用户提交后，调用restfull api. /admin/book/modify，实现对ISBN相应数据的更新。当然，出现异常ISBN或者数据库服务异常，还要给予错误提示，以友好的方式通知管理员。

（3）删除除图书信息

删除图书信息相对业务简单，完成ISBN的校验时，如果校验通过，则将ISBN数据中的删除标记置为真，否则不处理或者给予管理员异常提醒。删除时不建议将数据删除，如果删除了数据，对于用户还未完成的订单就无法查询到订单关联的图书信息，也对后续恢复数据造成困扰。

（4）图书的查询与分页

在完成以上增添改时，最终会落到列表页，此模块就是方便整理并显示所有图书信息，包括已经删除的信息。它可以将数据库中保存的图书信息，按照书名、作者、出版社、入库时间、删除状态等以列表的形式展现出来，同时能够通过以上维度进行搜索操作。在每一条列表信息后面还要支持各种操作功能，方便管理员有针对的进行“添加”“修改”和“删除”操作。当然点击标题文本，还能快速地查看书籍详情以及销售量等等。

图书管理的页面如图5.12后台图书管理模块。



图5.12后台图书管理模块

2.用户管理界面

用户的管理功能除了能够浏览和查询用的基本信息，同时还能对购物有不良记录的用户进行禁用操作，以禁止此用户扰乱正的图书管理秩序，同时具有提醒用户功能，以警示用户。



图5.13后台用户管理模块

3.订单管理模块

此模块涉及用户，以及图书商品，还有用户与商品的关联关系。主要功能是通过订单ID信息，反馈出与其关联的用户信息和商品的详细信息，当然还包括了订单的状态信息（如发货，拒绝等），对于订单，管理员只有的查看的权限，对于已经订单信息是不可以随便进行修改的，否则会产生不可补救的后果。

对于订单的操作功能要做如下控制，比如已经完结的订单（已发货）或者已经拒绝（商品下架）的订单，只支持“详情”操作，不允许二次发货或者拒绝，只有待处理的订单才有发货和拒绝两个操作功能。具体展示如图5.14：



图5.14后台订单管理模块

4.售后管理模块

此功能为售后模块，用于处理注册用户和普通用户问题，与用户进行线上交流。用户可以通过前台系统提交问题，管理员在后台系统处理回复，此时前台用户在反馈入口也能即时的看到管理员的回复内容，实现交流。目前暂不支持二次回复功能，已经回复的问题，不再支持回复操作。



图5.15后台反馈管理模块

5.部分Java服务程序

（1）数据库连接模块

package utils;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBUtil {

// 数据库连接路径

private static final String URL = "jdbc:mysql://127.0.0.1:3306/db\_shops?characterEncoding=utf8&useSSL=false";

private static final String NAME = "root";

private static final String PASSWORD = "root";

private static Connection conn = null;

static{

try {

// 加载驱动程序

Class.forName("com.mysql.jdbc.Driver");

// 获取数据库的连接

conn = DriverManager.getConnection(URL, NAME, PASSWORD);

} catch (ClassNotFoundException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

}

...

}

（2）Restfull API基本实现

通过HttpServlet对象获取request参数，根据参值进行数据查询或保存操作，如下：

// GET /admin/book/list/?name

@Override

public void doGet(HttpServletRequest request, HttpServletResponse response) {

String param = request.getParameter("name");

Integer key = (param == null) ? null : Integer.valueOf((param.trim()));

boolean json = false;

String accept = request.getHeader("accept");

if (accept != null && accept.contains("json")) json = true;

String payload = "";

Object list = null;

if (key == null) {

list = this.dao.query(null);

if (json) payload = JSONObject.fromObject(list);

}

// Otherwise, return the specified Novel.

else {

HashMap map = new HashMap();

map.put(0, key);

list = this.dao.query(map);

}

this.sendResponse(response, payload);

}

（3）数据DAO的增，添，改，查

/\*\* 添加图书信息

\*

\* @param book

\* @throws Exception

\*/

public void addBook( Book book ) throws Exception{

// 首先拿到数据库的连接

Connection con = DBUtil.getConnection();

String sql="insert into tb\_books"

// isbn、书名、图书价格、图书作者、出版社

+ "(BOOK\_isbn, BOOK\_name, BOOK\_price, BOOK\_author, BOOK\_publisher,"

+ "BOOK\_type, BOOK\_pic, BOOK\_intro) "

+ "values("

+ "?,?,?,?,?,?,?,?)";

PreparedStatement psmt = con.prepareStatement(sql);

// 先对应SQL语句，给SQL语句传递参数

psmt.setString(1, book.isbn);

psmt.setString(2, book.getBookName());

psmt.setFloat(3, book.price);

psmt.setString(4, book.author);

psmt.setString(5, book.publisher);

psmt.setString(6, book.type);

//执行SQL语句

psmt.execute();

}

修改图书：

/\*\*

\* 更新图书信息

\*/

public void update(Book book) throws SQLException{

// 首先拿到数据库的连接

Connection con=DBUtil.getConnection();

String sql="update tb\_books "

+ "set BOOK\_isbn = ?, BOOK\_name = ?, BOOK\_price = ?, BOOK\_author = ?"

+ ",BOOK\_publisher = ?, BOOK\_type = ?, BOOK\_intro = ?"

+ "where ID = ?";

// 预编译sql语句

PreparedStatement psmt = con.prepareStatement(sql);

// 先对应SQL语句，给SQL语句传递参数

psmt.setString(1, book.isbn);

psmt.setString(2, book.getBookName());

psmt.setFloat(3, book.price);

psmt.setString(4, book.author);

psmt.setString(5, book.publisher);

psmt.setString(6, book.type);

psmt.setString(7, book.intro);

psmt.setInt(9, book.ID);

// 执行SQL语句

psmt.execute();

}

六、系统测试

（一）系统测试环境搭建

测试环境包括硬件环境、软件环境两部分。

系统测试的硬件环境如下所示：

硬件环境：CPU 类型：Inter(R) Core(TM) i5-3470 CPU @3.20 GHZ ；

CPU个数：4个

CPU一级缓存：4\*64kbytes

CPU二级缓存：4\*1024kbytes

总线类型：FSB

内存类型：DDR3

内存容量：8GB

硬盘容量：500GB

系统测试的软件环境如下所示：

软件环境：

操作系统：Windows win7

网络服务器：Tomcat5.3；

JDK:java version "1.8.0\_181"

数据库：SQL Server 2012；

前端框架：Jquery1.8

浏览器环境：InternetExplorer10, GoogleChrome 58.0.3029.110, Firefox39.0

（二）部分功能测试

1.前台功能测试

测试目的，验证软件功能的可用性，功能稳定性以及在不同客户端环境的兼容性。

测试内容：

（1）测试前台页面在不同浏览器环境下的布局展现，验证有没有出现较大的布局问题。

（2）测试帐号功能，验证用户登录注册流程是否正确，同一个用户ID是否能反复注册等。

（4）验证不能同级别的用户权限，导航菜单用户区域功能是否有异常。

（5）测试网站首页图书推荐，书库等模块按不同维度搜索时，响应内容是否有异常

（6）验证购物车功能的正确，包括清空购物车，删除所选内容功能。验证 “购买”功能，购买操作后，商品是否能添加至购物车。

（7）验证售后模块功能是否正常，能否正确提交数据。

2.后台功能测试

测试目的：验证后台服务的功能可用性，数据准确性

测试内容：

（1）验证退出登录后，再次访问图书管理，帐号管理等模块时，是否还能正常操作。

（2）验证图书管理模块，添加图书，下架图书后，前台页面在呈现上是否有变化。

（4）验证通过前台注册用户，后台用户模块里是否能够显示新注册的用户。

（5）测试禁用用户后，当前用户ID是否能正常登录或注册。

（6）验证处理订单为发货状态时，前台页面的状态是否能正确更新。

（7）验证前台下单后，后台能否看到正确的订单状态。

（8）验证后台回复售后信息时，前台用户登录后是否能正确看到内容。

（三）测试结果

1.前台功能测试结果

（1）网站前台在InternetExplorer,Chrome,firefox三大主流浏览器上，界面表现基本一致，满足要求，如图5.16：



图5.16前台测试结果1

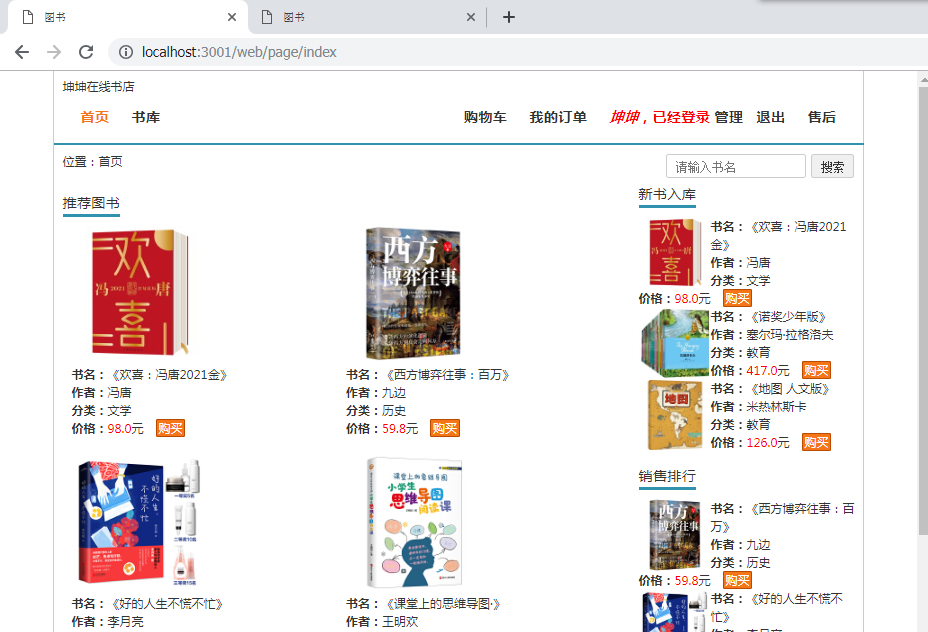


图5.17前台测试结果2

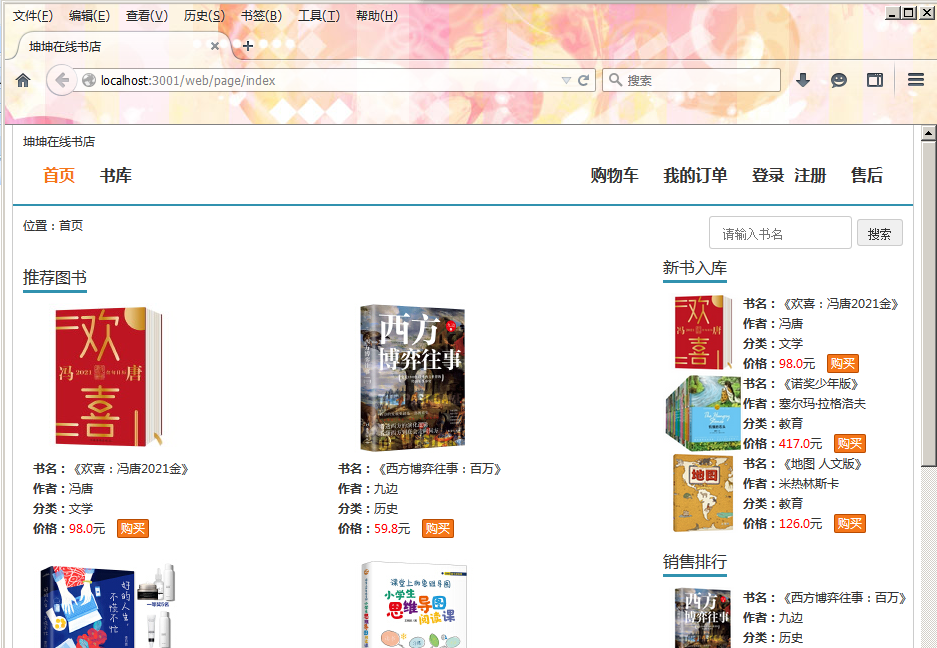


图5.18前台测试结果3

（2）通过wangxk多次注册，第二次通过wangxk注册，失败，提示重复用户，满足需求。

（3）wangxk为管理员帐号(昵称坤坤)，登录后有“管理入口”，xiaoz为普通用户(昵称小张)，登录后无管理入口，功能满足需求。如图5.15和5.16



图5.19前台测试结果4



图5.20前台测试结果5

（4）首页推荐图书，书库搜索功能正确，通过分类搜索，能正确匹配结果，满足需求。

（5）通过首页，书库“购买”，相应的图书都能正确添加至购物车，购物车清空，删除功能都能正确响应，满足需求。

（6）售后功能能够正确搜索，并且能够提交数据，功能正常。

2.后台功能测试结果

（1）管理员通过“退出”功能退出，再次输入管理页，无法访问并自动跳至登录页，功能满足需求

（2）通过后图书模块添加或修改图书，前台能正确展示，后台下架图书操作后，前台也不再展示下架的图书，功能满足需求。

（4）前台注册用户后，后台系统也能正确看到相应的用户信息，用户功能模块正确。

（5）后台禁用用户后，前台无法通过当前用户登录或注册，功能满足。

（6）后台处理订单，并发货，前台能正确展示为“发货中”功能满足需求。

（7）前台下单后，后台正确看到正确的订单状态，功能正确。

（8）回复售后信息时，前台用户登录后能正确看到内容。

后台的操作能实时的反馈给前台，数据库相关字段也能准确更新，基本实现现在的设计，功能满足了需求，在接口调用读取过程中未出现异情况，RestFull API的这种架构是可行的。

经过上述对于系统的功能测试，并得以下总结：

（1）该系统的软件界面响应速度时间小于1秒（目前是单用户测试）。

（2）软件对客户端软件的兼容性，以及稳定性，并且在标配或低配环境表现出很好的可靠性和健壮性。

综上所述，线上图书销售系统的设计与开发实践，完全符合预期。

七、总结

本文主要介绍了在线图书销售系统的设计和实现，通过对线下销售的流程总结并系统的进行需求分析，基于RESTfull API前后端分离的设计思想，通过JAVA，SQLServer，WEB等相技术，进行了详细地设计，并且通过兼容测试以及前后端各模块功能测试验证了该系统。通过这次设计经验取得的重要成果有：

（1）能够运用现在学习到的技术解决现实生活问题，帮助改善并优化线下销售的销售效率。

（2）技术上将RESTfull API前后端分离架构思应用于实践，高效的完成前后台技术解耦。

（3）通过对线下图书销售进行业务流程分析，根据当下国内线上各个销售平台，提出可行性方案，并根据所学知识点完成流程设计、功能设计、时序图设计和数据库设计，同时参考前辈的设计理念，最终确定前台和后台架构加上API架构，提升了自己的问题分析能力以及产品设计能力。

（4）图书销售系统通过前台和后台两大系统进行了功能总结，前台实现图书商店，给用户提供图书展台，用户入口，用户交互功能，如加购物车，查订单，反馈，后台实现对前台业务的管理，如图书管理，用户管理，交互订单管理，前台和后台配合，基本实现了线下图书销售所有的业务。线下新书推荐，销售排行，分类搜索这种功能增强，是线下图书销售书店不可比拟的。同时后台支持不同维度的图书搜索，订单搜索，很大程度上方便了管理员，提高了工作效率。

（5）通过对软件系统的功能测试以及兼容测试，在不同浏览器环境都表现了良好稳定和健壮性。

八、致谢

转眼间，经过几个年头的在线学习，让我的人生有了不同的轨迹，感谢老师对我的严厉教导让能够时时刻刻的自我反省与批评，我会更努力把每件事认真地做好...

【参考文献】

[1]赵娜.图书销售系统流程分析——以无锡某出版社为例[J].中外企业家,2020(17):99.

[2]余兵兵.关联规则挖掘在图书销售系统设计中的使用[J].科学技术创新,2019(01):86-87.

[3]孙加兵,施阳.网上图书销售系统的分析与设计[J].微处理机,2018,39(01):32-35.

[4]李小为,张健.中小型书店信息管理系统的设计与实现[J].电脑知识与技术,2016,12(36):14-16.

[5]况昊.网上书店详细设计与开发[J].现代商贸工业,2016,37(13):176-179.

[6]胡能发.基于UML图书销售系统分析与设计[J].电脑知识与技术,2015,11(26):67-68.

[7]郭健,任永功.云计算环境下的关联挖掘在图书销售中的研究[J].计算机应用与软件,2014,31(11):50-53.

[8]杨巍.基于VFP的图书销售管理系统的设计与开发[J].中国管理信息化,2014,17(21):32-33.

[9]春喜 陈猛 . 基于B/S的图书管理系统的设计与实现 [J]. 电脑知识与技术, 2012,(11)

[10] 程莹,张云勇,房秉毅,等.云计算时代的数据库研究[J].电信技术,2011,(1).

[11]刁宇飞.网上图书销售系统的设计与实现[J].电子技术与软件工程,2013(16):84.