# 1.绪论

## 1.1 背景、目的、意义

随着互联网的不断发展，电子商务逐渐代替传统的交易方式。各种类型的电子商务网站层出不穷，但细分之下电子商务网站主要分为B2B，B2C，C2B，C2C这四大类，而C2C网站更是体现了当今电子商务领域的趋势与走向。越来越多的传统行业都进入的了电子商务时代，服装、餐饮、快速消费品、化妆品等。而传统出版行业，尤其是传统的书店形式更是因为电子商务的崛起，受到了莫大的挑战。书店也必将进行电子商务的转型。如果每个书店都自行建设电子商务网站，对于一些小型书店，建设及运营成本必然过高，而且还需要运营这个网站的书店进行前期的宣传，如果这些投入并不能带来收益上的提高，必然是亏损的。现在普遍有三种电子商务运营方式，一是全托付型，相关企业只要和专门建设和运营的企业签订开发运营协议，并每年支付一定的费用，就能拥有独立的电子商务网站，这样做的好处有，有独立的域名，独立的管理系统，遇到技术问题有专业人员及时处理，价格较企业自行运营开发较为低廉等，但缺点是显而易见的，网站的安全性，用户信息的保密性等信息势必会由第三方获知；第二种是独立开发独立运营，企业组建独立部门进行电子商务网站的开发与运营，这样带来的好处是便于管理与二次开发，但是缺点是价格高昂，除了需要购置专业的服务器，数据库软件等支出外，还要建设和维护服务器机房，要招募相关技术人员和运营维护人员，开发周期长，与企业原有制度冲突，导致网上交易流程变长，用户等待过久等问题；第三种就是通过在第三方电子商务平台上发布自己的电商系统，第三方平台开发较为完善，操作简便等优势。所以大部分中小型企业或卖家选择在第三方平台上发布电商网站。而针对个人来说，手中积压的书籍往往不能得到充分的利用。而一个开放的C2C的电子商务平台就能有效有针对性的解决这些问题。

## 1.2 优势、意义

对于传统意义上的书店，网络电子书店的优势是一目了然的：

⑴ 信息量大。与传统书店相比，电子书店能够提供海量图书信息。

⑵ 书目信息丰富。目前，大多数电子书店都提供尽可能完整详细的书目信息，如书目、作/译者，出版社、出版日期、书号、页码、内容提要、价格等，有的甚至提供图书的目录、前言、内容节选、精彩插图、书评、作者简介等。

⑶ 检索、订购便捷。网上电子书店提供了多种检索途径，可以从书名、作者等途径进行单独检索或组合，快捷准确。

⑷ 价格低廉。一般来讲，由于电子书店屏弃和简化了传统书店运作所需要的服务设施和多重环节，所以在电子书店买书，要比在普通书店便宜。

⑸ 个性化服务。电子书店不只单纯销售图书，还充分发挥网络优势，提供丰富多彩的个性化服务，满足读者特定的需求。

凭借上述独到的优势，网上书店一经问世便受到业内人士的青睐和效仿，在世界范围内如雨后春笋般地涌现。几经商战洗礼，事实证明网上书店具有强大的生命力和广阔的发展前景。在国内，网上书店的发展也是如火如荼，出版社、国有书店、民营书店等各种性质的图书营销实体均纷纷涉足这一领域，寻求新的经济增长点，并以此作为提高竞争力、扩大市场份额的有力手段。

# 系统相关技术

本章主要是对系统 研发涉及的相关技术进行介简要地介绍,涉及的技术主要包括 B/S 模式、[Maven](http://baike.baidu.com/view/336103.htm" \t "http://baike.baidu.com/_blank)项目对象模型(POM)、MySQL数据库、Tomcat服务器，后端使用java语言与 SpringMVC框架，前端方面主要涉及技术有html5,css3以及BootStrap框架，通过JavaScript,JQuery以及Ajax渲染页面以及发送异步请求,项目整体通过GitHub进行版本控制管理。

## 2.1 B/S模式

B/S[1] (Browser/Server)结构即浏览器和服务器结构。它是一种以Web技术为基础的新型的MIS系统平台模式。把传统C/S模式中的服务器部分分解为一个数据服务器与一个或多个应用服务器CWeb服务器),从而构成一个三层结构的客户服务器体系。

第一层客户机是用户与整个系统的接口。客户的应用程序精简到一个通用的浏览器软件。浏览器将HTML代码转化成图文并茂的网页。网页还具备一定的交互功能,允许用户在网页提供的申请表上输入信息提交给后台,并提出处理请求。这个后台就是第二层的Web服务器。第二层Web服务器将启动相应的进程来响应这一请求,并动态生成一串HTML代码,其中嵌入处理的结果,返回给客户机的浏览器。如果客户机提交的请求包括数据的存取,Web服务器还需与数据库服务器协同完成这一处理工作。第三层数据库服务器的任务类似于C/S模式,负责协调不同的Web服务器发出的SQL请求,管理数据库。这样就大大简化了客户端电脑载荷,减轻了系统维护与升级的成本和工作量,降低了用户的总体成本(TCO ) 0以目前的技术看,局域网建立B/S结构的网络应用,并通过Intemet/Intranet模式下数据库应用,相对易于把握、成本也是较低的。它是一次性到位的并发,能实现不同的人员,从不同的地点,以不同的接入方式(比如LAN, WAN, Intemet/Intranet等)访问和操作共同的数据库;它能有效地保护数据平台和管理访问权限,服务器数据库也很安全。

与传统的C/S模式相比,B/S具有很多的优点。首先,它简化了客户端,无需象C/S模式那样在不同的客户机上安装不同的客户应用程序,而只需安装通用的浏览器软件。这样不但可以节省客户机的硬盘空间与内存,而且使安装过程更加简便、网络结构更加灵活。其次,它简化了系统的 发和维护。系统的开发者无须再为不同级别的用户设计开发不同的客户应用程序了,只需把所有的功能都实现在Web服务器上,并就不同的功能为各个组别的用户设置权限就可以了。各个用户通过HTTP请求在权限范围内调用Web服务器上不同处理程序,从而完成对数据的查询或修改。现代企业面临着日新月异的竞争环境,对企业内部运作机制的更新与调整也变得逐渐频繁。相对于C/S, B/S的维护具有更大的灵活性。当形势变化时,它无须再为每一个现有的客户应用程序升级,而只需对Web服务器上的服务处理程序进行修订。这样不但可以提高公司的运作效率,还省去了维护时协调工作的不少麻烦。如果一个公司有上千台客户机,并且分布在不同的地点,那么便于维护将会显得更加重要。再次,它使用户的操作变得更简单。对于C/S模式,客户应用程序有自己特定的规格,使用者需要接受专门培训。而采用B/S模式时,客户端只是一个简单易用的浏览器软件。无论是决策层还是操作层的人员都无需培训,就可以直接使用。这使得企业的大部分书面文件可以被电子文件取代,从而提高了企业的工作效率,使企业行政手续简化,节省人力物力。

## 2.2 Maven

Maven是一个项目管理工具，它包含了一个项目对象模型 (Project Object Model)，一组标准集合，一个项目生命周期(Project Lifecycle)，一个依赖管理系统(Dependency Management System)，和用来运行定义在生命周期阶段(phase)中插件(plugin)目标(goal)的逻辑。当你使用Maven的时候，你用一个明确定义的项目对象模型来描述你的项目，然后Maven可以应用横切的逻辑，这些逻辑来自一组共享的（或者自定义的）插件。

Maven 有一个生命周期，当你运行 mvn install 的时候被调用。这条命令告诉 Maven 执行一系列的有序的步骤，直到到达你指定的生命周期。遍历生命周期旅途中的一个影响就是，Maven 运行了许多默认的插件目标，这些目标完成了像编译和创建一个 JAR 文件这样的工作。此外，Maven能够很方便的帮你管理项目报告，生成站点，管理JAR文件，等等。

## 2.3 MySQL

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司。MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性，它的主要功能有以下几点:

(1)基本的数据库操作:插入Insert、删除Delete、查询Select、更新Update。

(2)快速修复功能。

(3)安全性髙。

(4)对操作者简单易懂。

(5)增强了校对、字符支持和全球部署。

(6)加强了数据的复制性。

## 2.4 Tomcat

Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由[Apache](http://baike.baidu.com/subview/28283/5418752.htm" \t "http://baike.baidu.com/_blank)、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 5支持最新的Servlet 2.4 和JSP 2.0 规范。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用[服务器](http://baike.baidu.com/view/899.htm" \t "http://baike.baidu.com/_blank)，在中小型系统和并发访问用户不是很多的场合下被普遍使用，本系统采用Apache Tomcat 7.x，它在汲取了Tomcat 6.0.x优点的基础上，实现了对于Servlet 3.0、JSP 2.2和EL 2.2等特性的支持。除此以外的改进列表如下：

（1）Web应用内存溢出侦测和预防

（2）增强了[管理程序](http://baike.baidu.com/view/712297.htm" \t "http://baike.baidu.com/_blank)和服务器管理程序的安全性

（3）一般 CSRF保护

（4）支持web应用中的外部内容的直接引用

（5） 重构 (connectors, lifecycle)及很多核心代码的全面梳理

## 2.5 Java

Java是一种跨平台，适合于分布式计算环境的面向对象编程语言。具体来说，它具有如下特性：简单性、面向对象、分布式、解释型、可靠、安全、平台无关、可移植、高性能、多线程、动态性等。下面我们将介绍Java语言的面向对象、平台无关、分布式、多线程、可靠和安全等特性。

（1）面向对象

面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看作是对象。对象之间通过消息相互作用。另外，现实世界中任何实体都可归属于某类事物，任何对象都是某一类事物的实例。如果说传统的过程式编程语言是以过程为中心以算法为驱动的话，面向对象的编程语言则是以对象为中心以消息为驱动。用公式表示，过程式编程语言为：程序=算法+数据；面向对象编程语言为：程序=对象+消息。

所有面向对象编程语言都支持三个概念：封装、多态性和继承，Java也不例外。现实世界中的对象均有属性和行为，映射到计算机程序上，属性则表示对象的数据，行为表示对象的方法（其作用是处理数据或同外界交互）。所谓封装，就是用一个自主式框架把对象的数据和方法联在一起形成一个整体。可以说，对象是支持封装的手段，是封装的基本单位。Java语言的封装性较强，因为Java无全程变量，无主函数，在Java中绝大部分成员是对象，只有简单的数字类型、字符类型和布尔类型除外。而对于这些类型，Java也提供了相应的对象类型以便与其他对象交互操作。

多态性就是多种表现形式，具体来说，可以用“一个对外接口，多个内在实现方法”表示。举一个例子，计算机中的堆栈可以存储各种格式的数据，包括整型，浮点或字符。不管存储的是何种数据，堆栈的算法实现是一样的。针对不同的数据类型，编程人员不必手工选择，只需使用统一接口名，系统可自动选择。运算符重载（operatoroverload)一直被认为是一种优秀的多态机制体现，但由于考虑到它会使程序变得难以理解，所以Java最后还是把它取消了。

继承是指一个对象直接使用另一对象的属性和方法。事实上，我们遇到的很多实体都有继承的含义。例如，若把汽车看成一个实体，它可以分成多个子实体，如：卡车、公共汽车等。这些子实体都具有汽车的特性，因此，汽车是它们的“父亲”，而这些子实体则是汽车的“孩子”。Java提供给用户一系列类（class），Java的类有层次结构，子类可以继承父类的属性和方法。与另外一些面向对象编程语言不同，Java只支持单一继承。

（2）平台无关性

Java是平台无关的语言是指用Java写的应用程序不用修改就可在不同的软硬件平台上运行。平台无关有两种：源代码级和目标代码级。C和C++具有一定程度的源代码级平台无关，表明用C或C++写的应用程序不用修改只需重新编译就可以在不同平台上运行。

Java主要靠Java虚拟机（JVM）在目标码级实现平台无关性。JVM是一种抽象机器，它附着在具体操作系统之上，本身具有一套虚机器指令，并有自己的栈、寄存器组等。但JVM通常是在软件上而不是在硬件上实现。（目前，SUN系统公司已经设计实现了Java芯片，主要使用在网络计算机NC上。

另外，Java芯片的出现也会使Java更容易嵌入到家用电器中。）JVM是Java平台无关的基础，在JVM上，有一个Java解释器用来解释Java编译器编译后的程序。Java编程人员在编写完软件后，通过Java编译器将Java源程序编译为JVM的字节代码。任何一台机器只要配备了Java解释器，就可以运行这个程序，而不管这种字节码是在何种平台上生成的。另外，Java采用的是基于IEEE标准的数据类型。通过JVM保证数据类型的一致性，也确保了Java的平台无关性。

Java的平台无关性具有深远意义。首先，它使得编程人员所梦寐以求的事情（开发一次软件在任意平台上运行）变成事实，这将大大加快和促进软件产品的开发。其次Java的平台无关性正好迎合了“网络计算机”思想。如果大量常用的应用软件（如字处理软件等）都用Java重新编写，并且放在某个Internet服务器上，那么具有NC的用户将不需要占用大量空间安装软件，他们只需要一个Java解释器，每当需要使用某种应用软件时，下载该软件的字节代码即可，运行结果也可以发回服务器。目前，已有数家公司开始使用这种新型的计算模式构筑自己的企业信息系统。

（3）分布式

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上，操作分布是指把一个计算分散在不同主机上处理。

Java支持WWW客户机/服务器计算模式，因此，它支持这两种分布性。对于前者，Java提供了一个叫作URL的对象，利用这个对象，你可以打开并访问具有相同URL地址上的对象，访问方式与访问本地文件系统相同。对于后者，Java的applet小程序可以从服务器下载到客户端，即部分计算在客户端进行，提高系统执行效率。

Java提供了一整套网络类库，开发人员可以利用类库进行网络程序设计，方便得实现Java的分布式特性。

（4）可靠性和安全性

Java最初设计目的是应用于电子类消费产品，因此要求较高的可靠性。Java虽然源于C++，但它消除了许多C++不可靠因素，可以防止许多编程错误。首先，Java是强类型的语言，要求显式的方法声明，这保证了编译器可以发现方法调用错误，保证程序更加可靠；其次，Java不支持指针，这杜绝了内存的非法访问；第三，Java的自动单元收集防止了内存丢失等动态内存分配导致的问题；第四，Java解释器运行时实施检查，可以发现数组和字符串访问的越界，最后，Java提供了异常处理机制，程序员可以把一组错误代码放在一个地方，这样可以简化错误处理任务便于恢复。

由于Java主要用于网络应用程序开发，因此对安全性有较高的要求。如果没有安全保证，用户从网络下载程序执行就非常危险。Java通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁破坏。当Java字节码进入解释器时，首先必须经过字节码校验器的检查，然后，Java解释器将决定程序中类的内存布局，随后，类装载器负责把来自网络的类装载到单独的内存区域，避免应用程序之间相互干扰破坏。最后，客户端用户还可以限制从网络上装载的类只能访问某些文件系统。

上述几种机制结合起来，使得Java成为安全的编程语言。

（5）多线程

线程是操作系统的一种新概念，它又被称作轻量进程，是比传统进程更小的可并发执行的单位。C和C++采用单线程体系结构，而Java却提供了多线程支持。

Java在两方面支持多线程。一方面，Java环境本身就是多线程的。若干个系统线程运行负责必要的无用单元回收，系统维护等系统级操作；另一方面，Java语言内置多线程控制，可以大大简化多线程应用程序开发。Java提供了一个类Thread，由它负责启动运行，终止线程，并可检查线程状态。Java的线程还包括一组同步原语。这些原语负责对线程实行并发控制。利用Java的多线程编程接口，开发人员可以方便得写出支持多线程的应用程序，提高程序执行效率。必须注意地是，Java的多线程支持在一定程度上受运行时支持平台的限制。例如，如果操作系统本身不支持多线程，Java的多线程特性可能就表现不出来。

## 2.6 SpringMVC

Spring MVC框架是有一个MVC框架，通过实现Model-View-Controller模式来很好地将数据、业务与展现进行分离。从这样一个角度来说，Spring MVC和Struts、Struts2非常类似。Spring MVC的设计是围绕DispatcherServlet展开的，DispatcherServlet负责将请求派发到特定的handler。通过可配置的handler mappings、view resolution、locale以及theme resolution来处理请求并且转到对应的视图。Spring MVC请求处理的整体流程如下图：

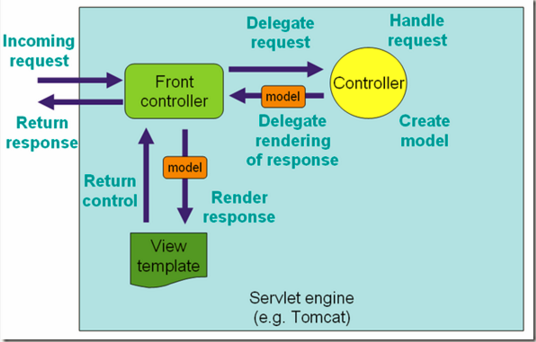


图 2-1 Spring MVC请求处理流程图

## 2.7 Html5,Css3

HTML5和CSS3不仅仅是两项新的Web技术标准，更代表了下一代HTML和CSS技术。虽然HTML5的标准规范还没有正式发布，但是未来的发展前景已经可以预见，那就是HTML5必将被越来越多的Web开发人员所使用，各大主流浏览器厂家已经积极更新自己的产品，以更好地支持HTML5。

Html5+css3相比较以前html和css主要有如下优势。

（1）更多的描述性标签：HTML5引入非常多的描述性标签，例如用于定义头部（header）、尾部（footer）、导航区域（nav）、侧边栏（aside）等标签，使开发人员非常方便的构建页面元素。

（2）良好的多媒体支持：对于先前的以插件的方式播放音频、视频带来的麻烦，HTML5有了解决方案，audio标签和video标签能够方便的实现应变。

（3）更强大的Web应用：HTML5提供了令人称奇的功能，某些情况下，你甚至可以完全放弃使用第三方技术。

（4）跨文档消息通信：Web浏览器会组织不同域间的脚本交互或影响，但是对于可信任的脚本或许就是麻烦。HTML5引入了一套安全且易于实现的应对方案。

（5）Web Sockets:HTML5提供了对Web Sockets的支持。

（6）客户端存储：HTML5的Web Storage和Web SQL Database API，可以在浏览器中构建Web应用的客户端持久化数据。

（7）更加精美的界面：HTML5+CSS3组合渲染出来的界面效果有时候是无法想象的精美。

（8）更强大的表单：HTML5提供了功能更加强大的表单界面控件，使用非常方便。

（9）提升可访问性：内容更加清晰，使用户的操作更加简单方便，体验提升。

（10）先进的选择器：CSS3选择器可以方便的识别出表格的奇偶行、复选框等，代码标记更少。

（11）视觉效果：精美界面的一部分，阴影、渐变、圆角、旋转等视觉效果。

（12）HTML5向后兼容

（13）HTML5废弃了不少常见的标签与属性

## 2.8 BootStrap

Bootstrap，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。[1] 它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言Less写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC（微软全国广播公司）的Breaking News都使用了该项目。[2] 国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。[3]

## 2.9 JavaScript JQuery Ajax

### 2.9.1 JavaScript

 JavaScript语言的前身叫作Livescript。自从Sun公司推出著名的Java语言之后，Netscape公司引进了Sun公司有关Java的程序概念，将自己原有的Livescript 重新进行设计，并改名为JavaScript。

JavaScript是一种基于对象和事件驱动并具有安全性能的脚本语言，有了JavaScript，可使网页变得生动。使用它的目的是与HTML超文本标识语言、Java 脚本语言一起实现在一个网页中链接多个对象，与网络客户交互作用，从而可以开发客户端的应用程序。它是通过嵌入或调入在标准的HTML语言中实现的。

JavaScript具有很多优点：  
　 (1).简单性 　JavaScript是一种脚本编写语言，它采用小程序段的方式实现编程，像其它脚本语言一样，JavaScript同样已是一种解释性语言，它提供了一个简易的开发过程。它的基本结构形式与C、C++、VB、Delphi十分类似。但它不像这些语言一样，需要先编译，而是在程序运行过程中被逐行地解释。它与HTML标识结合在一起，从而方便用户的使用操作。

(2).动态性　 JavaScript是动态的，它可以直接对用户或客户输入做出响应，无须经过Web服务程序。它对用户的反映响应，是采用以事件驱动的方式进行的。所谓事件驱动，就是指在主页中执行了某种操作所产生的动作，就称为“事件”。比如按下鼠标、移动窗口、选择菜单等都可以视为事件。当事件发生后，可能会引起相应的事件响应。

(3).跨平台性 　JavaScript是依赖于浏览器本身，与操作环境无关，只要能运行浏览器的计算机，并支持JavaScript的浏览器就可以正确执行。

(4).节省CGI的交互时间　 随着WWW的迅速发展有许WWW服务器提供的服务要与浏览者进行交流，确浏览的身份、需服务的内等等，这项工作通常由CGI/PERL编写相应的接口程序与用户进行交互来完成。很显然，通过网络与用户的交互过程一方面增大了网络的通信量，另一方面影响了服务器的服务性能。服务器为一个用户运行一个CGI时，需要一个进程为它服务，它要占用服务器的资源(如CPU服务、内存耗费等)，如果用户填表出现错误，交互服务占用的时间就会相应增加。被访问的热点主机与用户交互越多，服务器的性能影响就越大。

### 2.9.2 JQuery

JQuery是继prototype之后又一个优秀的Javascript库。它是轻量级的js库 ，它兼容CSS3，还兼容各种浏览器（IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+），jQuery2.0及后续版本将不再支持IE6/7/8浏览器。jQuery使用户能更方便地处理HTML（标准通用标记语言下的一个应用）、events、实现动画效果，并且方便地为网站提供AJAX交互。jQuery还有一个比较大的优势是，它的文档说明很全，而且各种应用也说得很详细，同时还有许多成熟的插件可供选择。jQuery能够使用户的html页面保持代码和html内容分离，也就是说，不用再在html里面插入一堆js来调用命令了，只需要定义id即可。

jQuery是一个兼容多浏览器的javascript库，核心理念是write less,do more(写得更少,做得更多)。jQuery在2006年1月由美国人John Resig在纽约的barcamp发布，吸引了来自世界各地的众多JavaScript高手加入，由Dave Methvin率领团队进行开发。如今，jQuery已经成为最流行的javascript库，在世界前10000个访问最多的网站中，有超过55%在使用jQuery。

jQuery是免费、开源的，使用MIT许可协议。jQuery的语法设计可以使开发更加便捷，例如操作文档对象、选择DOM元素、制作动画效果、事件处理、使用Ajax以及其他功能。除此以外，jQuery提供API让开发者编写插件。其模块化的使用方式使开发者可以很轻松的开发出功能强大的静态或动态网页。

### 2.9.3 Ajax

Ajax的全称是AsynchronousJavaScript+XML ，Ajax不是一个技术，它实际上是几种技术，每种技术都有其独特这处，合在一起就成了一个功能强大的新技术。Ajax结合了Java技术、XML以及JavaScript等编程技术，可以让开发人员构建基于Java技术的Web应用，并打破了使用页面重载的惯例。 Ajax是使用客户端脚本与Web服务器交换数据的Web应用开发方法。这样，Web页面不用打断交互流程进行重新加载，就可以动态地更新。使用Ajax，用户可以创建接近本地桌面应用的直接、高可用、更丰富、更动态的Web用户界面。

## 2.10 GitHub

Git是一个分布式的版本控制系统，最初由Linus Torvalds编写，用作Linux内核代码的管理。在推出后，Git在其它项目中也取得了很大成功，尤其是在Ruby社区中。目前，包括Rubinius、Merb和Bitcoin在内的很多知名项目都使用了Git。Git同样可以被诸如Capistrano和Vlad the Deployer这样的部署工具所使用。

利用GitHub，你可以将项目存档，与其他人分享交流，并让其他开发者帮助你一起完成这个项目。优点在于，他支持多人共同完成一个项目，因此你们可以在同一页面对话交流。

Git也有很多好处。他被视为一个预先维护过程，你可以按自己的需要恢复、提交出现问题,或者您需要恢复任何形式的代码，可以避免很多麻烦。Git最好的特性之一是能够跟踪错误，这让使用Github变得更加简单。Bugs可以公开，你可以通过Github评论，提交错误。同时使用Github提高了个人的容错率与方便，随时随地可以通过网络同步到本地生产代码。

# 3.需求分析

## 3.1 系统需求的描述

随着近几年来互联网科技的发展，各行各业越来越注重计算机的应用，同时也包括了商城这块，由于网上电子商城的出现和普及，很大程度的方便了顾客，同时也为商家节省了店铺的租赁费等一系列繁琐的问题。而网上书店的优越性更是显而易见，除了和普通网店的相似性之外还具备智能推荐功能和免费试读功能，以下是本系统的大概描述：

从商家自身角度：通过登录之后，以简洁明了的操作界面为商家提供服务，商家可以对自己书店的图书进行添加，修改，删除等操作，从而实现对商品的管理，还可以对销售情况进行查看和统计以及对订单的处理。

从顾客角度：顾客可以随意的浏览该书城的任何商品，登陆之后可以将自己满意的图书加入购物车进行购买以及和商家进行互动。

## 3.2 可行性分析

可行性分析(Feasibility analysis),是建立在对系统调查分析的基础上,对所开发的软件的系统分析和系统设计进行髙度抽象并进行客观分析的过程。为了避免投资开发的失误,尽可能减少人力物力上的损失,提高系统开发的成功率,针对系统的 发是否具备必要性和可能性,对系统的 发从技术、经济、操作等方面进行研究和分析。为了达到用最小的代价在尽可能短的时间内确定问题是否能够解决这个目的,必须分析几种主要的解决方案,从而综合权衡原定的系统目标和规模是否符合实际,以及系统完成后所能带来的最大社会和经济效益。因此对所要开发的系统进行可行性分析和研究是系统设计和开发前的不可或缺的重要过程。

下面结合智能推荐功能的图书销售系统的工作实际,从技术、经济和操作方面对本文所要开发的系统进行可行性分析。

### 3.2.1技术可行性

技术可行性是根据系统目标来衡量所需的技术是否具备,一般可从硬件,软件的性能要求,环境条件,技术人员水平和数量等方面去考虑和分析。该系统是一个售书系统,是基础的管理信息系统,在互联网金融行业发达的今天，大量学者对网络销售系统进行了深入和广泛的研究,其开发技术已比较成熟,硬件、软件的性能要求、环境条件等各项条件良好,利用现有的技术条件完全可以实现该系统的功能目标。本文在设计宿舍管理系统之前进行的技术可行性分析,主要考虑了硬件、软件是否能满足系统的开发,能否保证系统顺利地完成。采用得到广大开发人员认可的B/S模式进行系统的开发,利用面向对象方法结合数据库技术,所以能满足本系统系统的设计与实现系统的开发技术。

需要具备的硬件资源:PC机一台;软件资源:Windows 系统、mysql 数据库、Tomcat服务器、eclipse/spring tools suit等开发环境。

### 3.2.2经济可行性

经济可行性主要是对项目的经济效益进行评价,即估算系统的开发费用和今后的运行,维护费用,估计系统将获得的效益,并将费用与效益进行比较,看是否有利。网上售书系统是针对卖家和买家购书方便而开发的应用软件,属于一个小型软件系统,其开发,运行和维护的费用较低。使用本系统的卖家和买家只需要具有计算机、安装有操作系统软件、利用网络等便可以使用本系统,并且该系统本身也为用户提供了系统维护功能,减少了系统维护的费用。所以本系统在经济上是可行的。

### 3.2.3操作可行性

操作可行性[4]是对本系统运行后给现行系统带来的影响和后果进行估计和评价,本系统是基于B/S模式进行开发的,在Windows 操作系统环境下运行,系统界面友好,操作简单易懂,实用性强。用户对象是顾客和卖家管理工作人员,只需要仔细阅读操作方法和简单的计算机操作知识,就可以自由使用该系统。

## 3.3 需求分析

### 3.3.1 识别参与者和用例

通过对系统的分析，可以确定系统有三个参与者，分别是一般用户、商家。各个参与者的描述如下：

（1）普通用户（买家）：可以浏览图书和下订单，以及个人的基本信息管理

（2）商家：可以添加商品、删除商品、浏览商品，查看购买情况统计，以及操作订单

### 3.3.2 用例图

（1）普通用户（买家）的用例图



图3-1 普通用户（买家）用例图

（2）买家用例图说明如下：

●注册用例 初次登陆须进行账户注册，并提交个人信息。

●登陆用例 完成买家的登陆功能，验证身份以保证系统的安全。

●浏览图书用例 以最直观的形式顾客可以查看所有的图书

●浏览图书详情用例 可以为顾客提供详细的图书信息

●查看书评用例 为用户提供参考性意见

●免费试读用例 让用户进行试读

●查看购物车用例 为顾客提供已筛选商品

●新增订单用例 将本次购书信息生成订单

●删除订单用例 删除一次购书订单

●订单支付用例 为一次购物付款

●订单评价用例 对本次购物进行评价

●收货地址管理用例 管理买家的收获地址

●密码修改用例 修改登录密码，确保安全性

●在线交流用例 在线与商家进行沟通，让买家了解更多的关于图书信息

（3）卖家的用例图



图 3-2 卖家用例图

（4）卖家用例描述：

●登陆用例 完成卖家的登陆功能，验证身份以保证系统的安全。

●密码修改用例 为商家提供密码修改功能，确保安全性

●查看订单用例 对不需要的商品信息删除。

●删除订单用例 查询要寻找的商品信息。

●添加图书信息用例 添加新上架图书信息

●修改图书信息 对上架图书信息进行修改

●修改库存用例 修改图书的库存量

●试读章节添加用例 为图书添加试读内容

●试读章节修改用例 为图书修改试读内容

●试读章节删除用例 删除图书试读内容

●日销售额分时统计用例 销售额的分时显示

●按类型统计销量用例 按照分类进行统计销量

●图书销量排行用例 根据图书的销量进行排序

●在线交流用例 方便和顾客进行沟通

### 3.2.2用例描述

登录用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 登录 |
| 标识符 | UC001 |
| 用例描述 | 完成顾客、管理员和超级管理员的登陆功能，验证身份以保证系统的安全 |
| 参与者 | 顾客、管理员和超级管理员 |
| 前置条件 | 无 |
| 后置条件 | 登录到系统 |
| 基本操作流程 | 1.选择登录入口  2.系统提示输入用户名和密码  3.用户输入用户名和密码  4.系统验证用户名密码，若正确，用户登录到系统中 |
| 可选操作流程 | 如果用户输入无效的用户名和密码，系统显示无该用户并提示注册，如果用户输入的密码不正确，系统密码错误信息，并返回重新提示用户输入用户名和密码以及是否需要找回密码 |

注册用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 注册 |
| 标识符 | UC002 |
| 用例描述 | 初次购物须进行账户注册，并提交个人信息。 |
| 参与者 | 顾客 |
| 前置条件 | 无 |
| 后置条件 | 用户注册成功 |
| 基本操作流程 | 1.点击注册按钮  2.选择注册入口  3.用户输入有效的用户名、密码、确认密码  4.提交，系统检验输入信息是否合法  5.用户输入注册信息  6.系统检验输入信息是否合法，若合法，将该用户注册并提示注册成功 |
| 可选操作流程 | 1.如果用户名重复或不合法，提示重新输入用户名  2.如果密码和确认密码不同，提示两次输入的密码不相同，并返回提示重新输入密码  3.如果用户名输入不合法，提示用户名输入不合法，并返回重新输入 |

浏览用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 浏览 |
| 标识符 | UC003 |
| 用例描述 | 顾客 |
| 参与者 | 顾客 |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 基本操作流程 | 1.直接通过地址来浏览商品信息 |

查看商品详情用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 查看商品详情 |
| 标识符 | UC004 |
| 用例描述 | 反应给顾客详细的商品信息 |
| 参与者 | 顾客 |
| 前置条件 | 进入浏览界面 |
| 后置条件 | 无 |
| 基本操作流程 | 在浏览界面通过点击商品即可查看商品详细信息 |

查看购物车用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 查看购物车 |
| 标识符 | UC005 |
| 用例描述 | 顾客查看以加入购物车的商品 |
| 参与者 | 顾客 |
| 前置条件 | 顾客登陆之后 |
| 后置条件 | 无 |
| 基本操作流程 | 1.顾客登陆之后  2.点击查看购物车  3.进入购物车 |

查看书评用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 查看书评 |
| 标识符 | UC006 |
| 用例描述 | 顾客查看其他买家对书的评价 |
| 参与者 | 顾客 |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 基本操作流程 | 顾客选择某商品查看详情  查看书评 |

查看免费试读用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 查看书评 |
| 标识符 | UC007 |
| 用例描述 | 提前免费阅读书籍部分章节 |
| 参与者 | 顾客 |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 基本操作流程 | 1.顾客选择某商品查看详情  2.免费试读 |

查看免费试读用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 查看书评 |
| 标识符 | UC007 |
| 用例描述 | 提前免费阅读书籍部分章节 |
| 参与者 | 顾客 |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 基本操作流程 | 1.顾客选择某商品查看详情  2.免费试读 |

查看新增订单用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 新增订单 |
| 标识符 | UC007 |
| 用例描述 | 生成本次购物的订单 |
| 参与者 | 顾客 |
| 前置条件 | 成功登陆并且选择商品，选择收货地址 |
| 后置条件 | 成功生成订单为待支付状态 |
| 基本操作流程 | 选择具体商品并提交订单  填写订单信息  提交订单 |

查看删除订单用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 删除订单 |
| 标识符 | UC008 |
| 用例描述 | 删除某次购物的订单 |
| 参与者 | 顾客 |
| 前置条件 | 成功登陆后查看我的订单 |
| 后置条件 | 成功删除订单 |
| 基本操作流程 | 1.选择具体订单进行删除 |

查看订单支付用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 订单支付 |
| 标识符 | UC009 |
| 用例描述 | 对某订单进行支付 |
| 参与者 | 顾客 |
| 前置条件 | 成功生成订单 |
| 后置条件 | 成功支付订单 |
| 基本操作流程 |  |

查看订单评价用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 订单评价 |
| 标识符 | UC010 |
| 用例描述 | 对已经完成订单进行评价 |
| 参与者 | 顾客 |
| 前置条件 | 必须确认收货之后 |
| 后置条件 | 成功对本次交易评价 |
| 基本操作流程 | 确认收货  评价订单 |

管理收货地址用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 订单评价 |
| 标识符 | UC011 |
| 用例描述 | 对收货地址进行管理 |
| 参与者 | 顾客 |
| 前置条件 | 成功登陆之后 |
| 后置条件 | 成功管理收货地址 |
| 基本操作流程 |  |

密码修改用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 密码修改 |
| 标识符 | UC011 |
| 用例描述 | 修改登录密码，确保系统的安全性 |
| 参与者 | 买家，商家 |
| 前置条件 | 成功登陆并且验证旧登录密码 |
| 后置条件 | 成功修改登录密码 |
| 基本操作流程 | 成功登陆之后  验证旧登录密码  确认新密码 |

在线交流用例的描述：

|  |  |
| --- | --- |
| 用例名称 | 在线交流 |
| 标识符 | UC012 |
| 用例描述 | 在线进行沟通，让买家了解更多的关于图书信息 |
| 参与者 | 买家，商家 |
| 前置条件 | 双方都成功登陆 |
| 后置条件 | 实时沟通 |
| 基本操作流程 |  |

# 系统设计

## 4.1功能模块设计

### 4.1.1系统的架构设计模式选择

系统的架构反映了系统的整个功能逻辑结构,主要有C/S (Client/Server,客户机/浏览器)和B/S (Brower/Server,浏览器/服务器)及两者混合3种模式。其中B/S模式是一种把Web技术和数据库技术结合起来的新技术,实现了开发环境和应用环境的分离。基于B/S模式设计的系统具有如下优点:

(1)系统需要处理的数据集中存放在系统的数据库服务器,客户端不需要保存任何业务数据和数据库连接信息,减轻了客户端的负荷。

(2)它可以实时看到当前发生的所有业务,数据实时性更强。

(3)由于其数据是集中存放的,易于进行数据跟踪。

(4)服务器更新则全部应用程序更新。

因此,考虑到实际工作情况以及用户购物的方便、快捷的优势,本文设计的系统釆用了 B/S模式。

### 4.1.2系统功能模块及说明

1. 用户登录模块

登录模块主要包括卖家和买家两种角色，卖家管理员通过登录系统之后管理书籍，订单等等，买家通过登录之后成功购买书籍，下订单。

1. 订单处理模块

订单处理模块也分为买家和卖家两个角色，买家主要生成订单，卖家则是处理买家生成的订单，然后买家在交易成功之后可以对订单进行评价，因此完成了完整的购物流程。

1. 在线交流模块

买家在购物时可以向卖家咨询关于书籍以及订单等方面的任何问题，方便买家的同时促进买卖之间的交流，使得双方达成共赢。

1. 图书相关模块

卖家通过图书添加编辑等功能向系统录入图书信息，之后买家就可以通过系统浏览并且购买书籍。

1. 销量统计模块

此模块主要是针对卖家统计图书的销量信息，入账金额统计，使得交易数字化，从而更好地管理图书，尽可能的为买家提供服务的同时自我更新。

1. 公告/优惠信息模块

卖家：主要通过系统发布两种信息，第一是优惠信息显示给用户看，从而达到促进销量的目的，第二是发布系统消息，为其他系统管理员发布公告等等。

买家：主要通过系统查看优惠信息，第一时间享受优惠或者折扣使得购物更加实惠趣味化

1. 购物车模块

购物车主要是买家在下单时可以同时购买不同种类的书，省去了每次填写订单的步骤，为买家提供方便。

1. 智能推荐模块

本模块为此系统与其他普通购物网站不同的模块，卖家可以通过选择推荐方式来为买家推荐图书，不仅方便买家省去了盲目选择书籍，同时也促进了卖家的销量。

### 4.1.3系统功能模块结构

根据系统的设计目标及功能性需求分析,具有智能推荐功能的图书销售系统的总体模块结构图如图4-1所示。

图 4-1

## 4. 2数据库设计

### 4.2.1数据库设计方案

结构化的数据库[5]是一种对数据进行有效管理的手段,同时结构化的数据库也是产生正确数据的一个重要保证。数据是系统运行的核心,系统数据库是整个系统的基础和灵魂,数据库存储的数据是用户使用系统的基本数据,系统的任何操作均围绕数据进行,因此一个好的数据库设计方案可以有效地降低数据的冗余度,实现数据共享,避免资源浪费,同时一个合理的数据组织方式可以使得系统的逻辑更容易理解与实现,减少系统开发的难度。相反,如果数据库设计不恰当有可能造成数据库极度膨胀,严重时会影响系统中的软件的运行效率和稳定性,甚至可能造成程序的瘫 。因此数据库在设计的时候要充分考虑系统的数据需求,理清数据之间的逻辑关系,充分考虑数据之间的相互影响,并且数据库中表的各种关系和完整性约束条件要能够真实地反映出现实数据之间的关系。

### 4. 2. 2 E-R模型设计

E-R(Entity-Relation,实体-联系)数据模型[25]是一种描述现实信息世界中数据的静态特性的重要方法。E-R数据模型就是利用实体、联系和属性这三种成分来描述反映一个客观信息世界内容的数据模型。它是从现实信息世界到数据库系统中数据逻辑组织结构设计之间的一个中间阶段。本系统总共有以下几个实体的E-R图：

1. 顾客（买家）实体

主要记录买家个人登录信息如图4-2所示



图 4-2

1. 管理员（卖家）实体

主要记录管理员登录信息，权限信息以及是否正常使用状态，如图4-3所示



图4-3

1. 图书实体

主要存储详细的图书信息如图4-4所示



图4-4

1. 订单实体

订单实体主要存储记录订单的相关信息，比如时间状态等等 如图4-5所示



图4-5

1. 订单详情实体

该实体主要记录相应订单的详细商品信息，结合有图书实体和订单实体内容如图4-6所示



图 4-6

1. 公告实体

该数据表主要记录保存优惠信息和系统公告，通过表中字段进行分类，如图4-7所示



图 4-7

(7)图书大类实体

存储记录图书所属大类信息 如图4-8所示



图 4-8

1. 图示细类实体

为了便于管理，记录相应大类下的图书小分类信息 如图4-9所示



图 4-9

1. 图书评价实体

记录图书等级评价以及评价内容，便于卖家统计买家需求以及为其他买家提供参考 如图4-10所示



图4-10

1. 顾客兴趣记录实体

简单记录用户注册时候的问卷调查信息 如图4-11所示



图 4-11

1. 免费试读实体

存放图书免费试读信息，方便买家了解图书信息 如图4-12所示



图 4-12

1. 搜索记录实体

存放用户搜索记录，更加准确的为买家推荐图书 如图4-13所示



图4-13

1. 购物车实体

临时存放买家想购买的图书信息，方便买家（省去买多种书籍需要多次填写订单信息）如图4-14所示



图4-14

1. 推荐方式实体

存放不同类型的买家的推荐方式 如图4-15所示



图4-15

### 4.2.3数据库的逻辑设计

根据以上设计的系统实体的E-R图,将数据库的概念结构模型转化为以下逻辑结构模型,即设计出具体的数据库表来存放数据。分别描述如下:

1. 买家表(customer) 如图4-16所示。

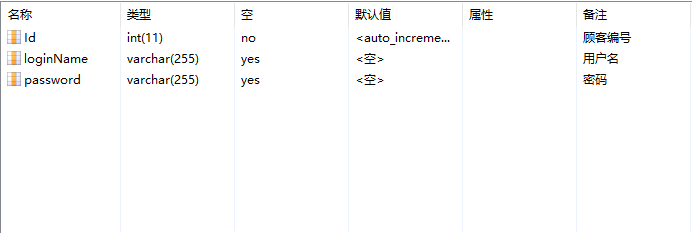


图 4-16

1. 订单表(allorder) 如图4-17所示。

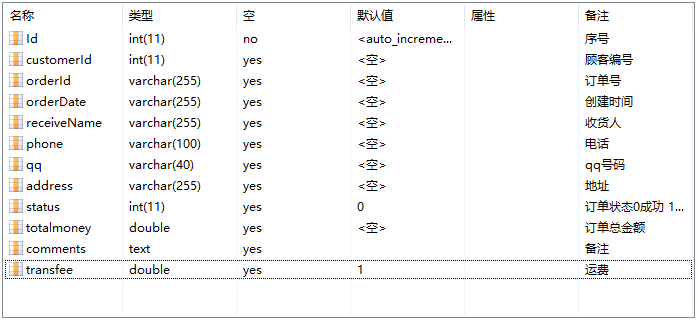


图4-17

1. 订单评价表(assess) 如图4-18所示。

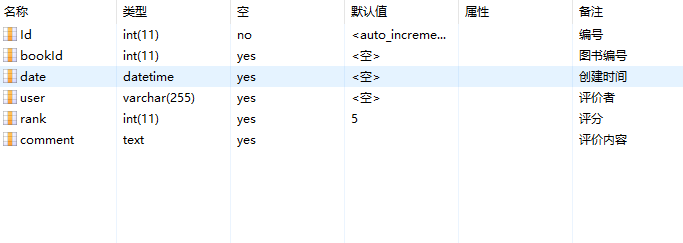


图4-18

1. 图书表(book) 如图4-19所示。

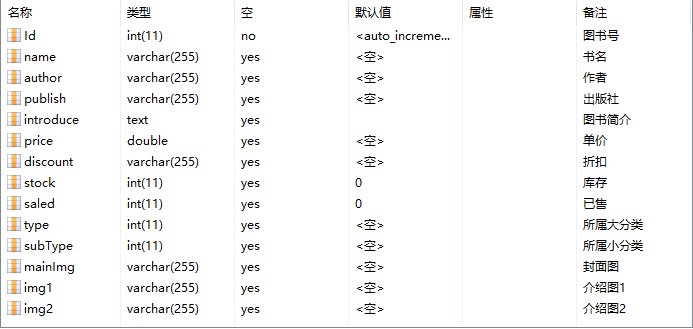


图4-19

1. 买家兴趣表(customer\_interesting) 如图4-20所示。

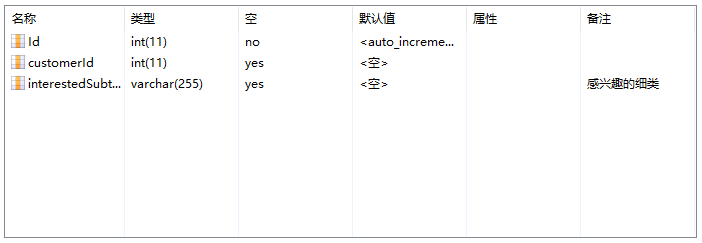


图4-20

1. 卖家表(manager) 如图4-21所示。

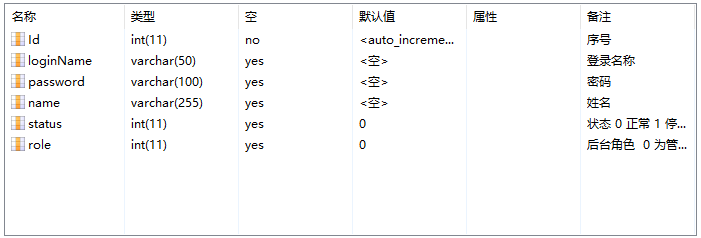


图4-21

1. 公告表(customer) 如图4-22所示。

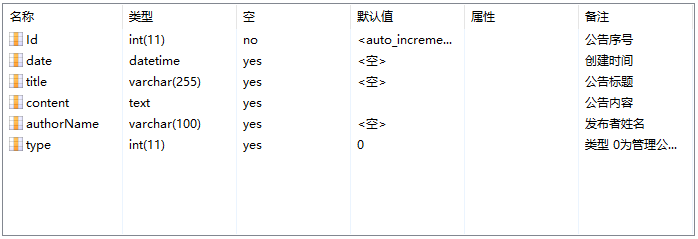


图4-22

1. 订单详情表(order\_detial) 如图4-23所示。

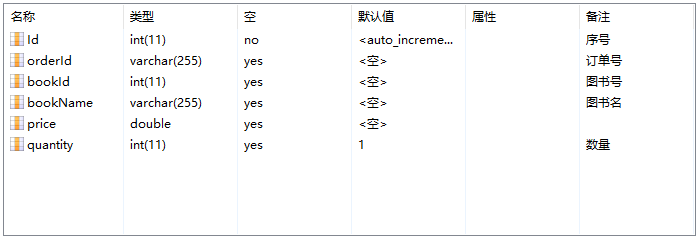


图4-23

1. 免费试读表(read\_free) 如图4-24所示。

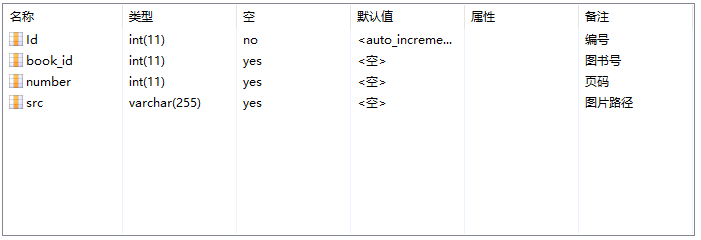


图4-24

1. 买家搜索记录表(searchinfo) 如图4-25所示。

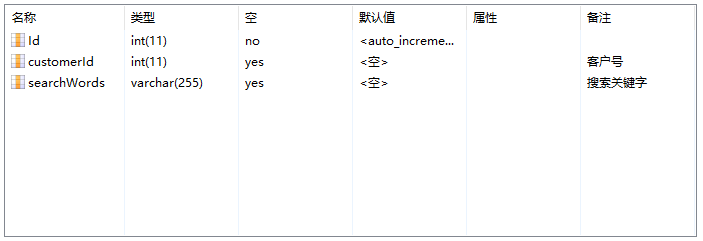


图4-25

1. 购物车表(shopcar) 如图4-26所示。

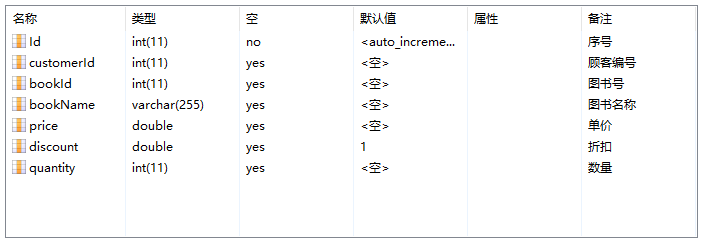


图4-26

1. 图书大类表(type) 如图4-27所示。

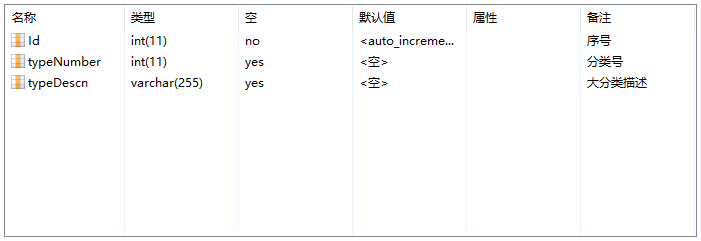


图4-27

1. 图书细类表(subtype) 如图4-28所示。

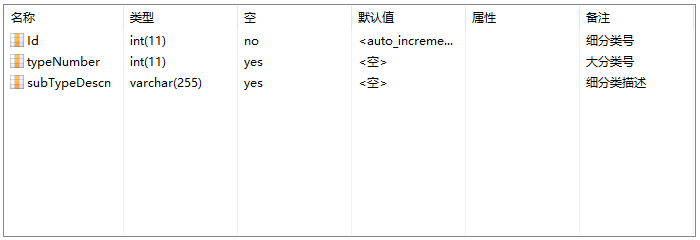


图4-28

1. 只能推荐表(suggest) 如图4-29所示。

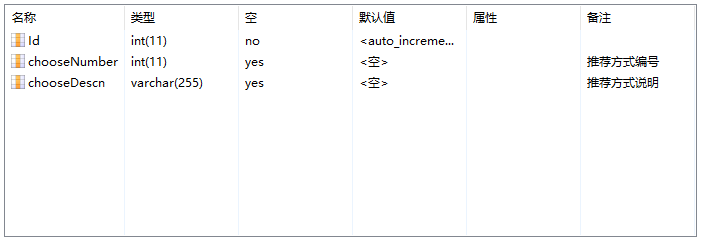
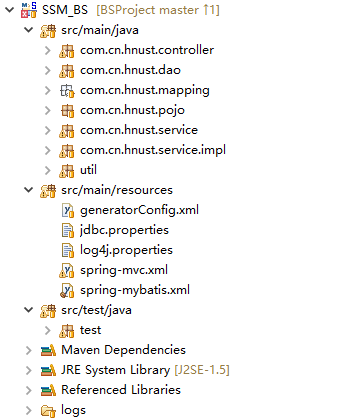


图4-29

## 4. 3 系统目录设计

根据以上所有的功能需求分析，结合开发实际情况，为了便于管理，维护及系统的整体规范性，对系统目录进行如下设计，具体如图4-30所示。



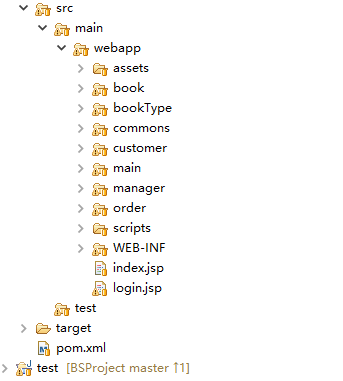


图 4-30

# 系统实现

## 5.1 卖家（管理员）模块

### 5.1.1 登录模块

作为后台管理模块，为了保证系统的安全性，登录是必不可少的一部分，并且由于是管理员模块，所以账号都是超级管理员通过系统添加的，不提供注册接口，只有符合条件的账号才可以登录系统并完成相应的系统操作，登录界面如下（图5-1）：



图 5-1

登录模块的主要逻辑代码（controller）如下：

/\*\*

\* 管理员登录

\* **@param** request

\* **@param** model

\* **@param** manager

\* **@param** session

\* **@return**

\* **@throws** ParseException

\*/

@RequestMapping("/turnIndex")

**public** String toIndex(HttpServletRequest request,Model model,Manager manager,HttpSession session) **throws** ParseException{

**int** error=0;

Map<String, String> condition= **new** HashMap<String, String>();

condition.put("loginName",manager.getLoginname());

condition.put("password",manager.getPassword());

List<Manager> managers= managerService.getManager(condition);

**if**(managers.size()!=1){

error=1;

model.addAttribute("error", error);

**return** "login";

}

//添加登录者session信息

session.setAttribute("managerId", managers.get(0).getId());

session.setAttribute("managerLoginname", managers.get(0).getLoginname());

session.setAttribute("managerPassword", managers.get(0).getPassword());

session.setAttribute("managerName", managers.get(0).getName());

session.setAttribute("managerRole", managers.get(0).getRole());

**return** "redirect:/main/index";

}

### 5.1.2 管理员首页模块

首页是管理员登录之后的最直接能看到的部分，因此首页的设计应该包含一些整体的最重要的内容，从而达到以最简单最直观的方式向管理员呈现最重要的信息。首页设计图如图5-2：

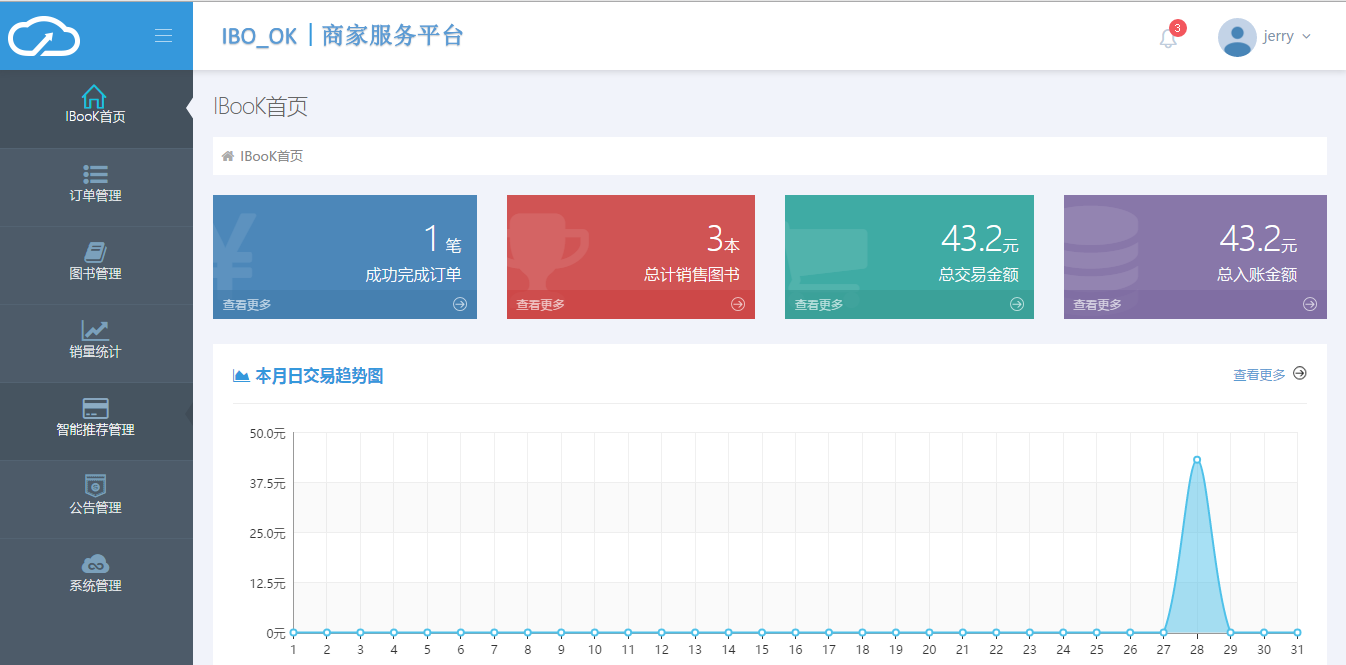




图5-2

登录模块的主要逻辑代码（controller）如下：

@RequestMapping("/index")

**public** String turnIndex(HttpServletRequest request, Model model) {

Date date = **new** Date();

DateFormat format = **new** SimpleDateFormat("yyyy-MM");

String dateParam = format.format(date);

// 查询当月交易

List<Order> list = **new** ArrayList<Order>();

List<MonthMoney> moneyList = **new** ArrayList<MonthMoney>();

list = orderService.findByMonth(dateParam + "%");

**double** allMoney = 0;

**for** (**int** i = 1; i <= 31; i++) {

MonthMoney monthMoney = **new** MonthMoney();

**double** daycount = 0;

**for** (**int** m = 0; m < list.size(); m++) {

String day = list.get(m).getOrderdate().substring(8, 10);

**if** (i == Integer.*parseInt*(day)) {

daycount += list.get(m).getTotalmoney();

}

}

monthMoney.setDay(i);

monthMoney.setMoney(daycount);

moneyList.add(monthMoney);

allMoney += daycount;

}

model.addAttribute("moneyList", moneyList);

model.addAttribute("allOrder", list.size());

model.addAttribute("allMoney", allMoney);

// 查询本月共销售图书量

**int** allBook = 0;

**for** (Order orderlist : list) {

List<OrderDetial> detialList = **new** ArrayList<OrderDetial>();

Map<String, Object> condition = **new** HashMap();

condition.put("orderid", orderlist.getOrderid());

detialList = orderDetialService.getOrderDetialNoPage(condition);

**for** (OrderDetial orderDetial : detialList) {

allBook += orderDetial.getQuantity();

}

}

model.addAttribute("allBook", allBook);

// 查询首页显示公告

**int** number = 3;

List<Notice> allNotice = noticeService.SellectAll();

List<Notice> managerNotice = **new** ArrayList();

List<Notice> activeNotice = **new** ArrayList();

**for** (Notice notice : allNotice) {

**if** (notice.getType() == 0) {

managerNotice.add(notice);

} **else** {

activeNotice.add(notice);

}

}

**if** (managerNotice.size() > number) {

**for**(**int** i=managerNotice.size();i>number;i=managerNotice.size()){

managerNotice.remove(3);

}

}

**if** (activeNotice.size() > number) {

**for**(**int** i=activeNotice.size();i>number;i=activeNotice.size()){

activeNotice.remove(3);

}

}

model.addAttribute("managerNotice",managerNotice );

model.addAttribute("activeNotice",activeNotice );

**return** "/main/index";

}

### 5.1.3 订单模块

订单作为买家和卖家直接唯一的也是最重要的一份协议，不但是一份汇总信息，也是买卖几件的桥梁。具体设计如下：

（1）订单总体管理界面如图5-3：

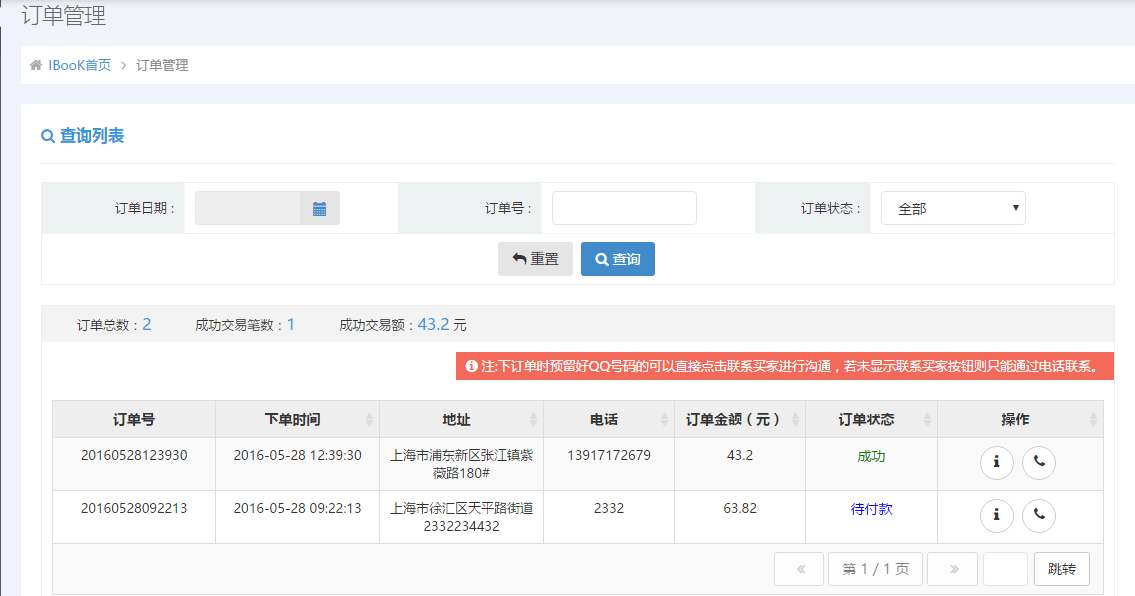


图 5-3

1. 订单详情界面设计如图5-4：



图5-4

（3）订单操作信息界面设计如图5-5：



图 5-5

（4）订单查询模块界面书籍如图5-6所示：



图5-6

### 5.1.4 图书分类管理模块

为了便于图书的管理与用户的检索以及根据用户喜欢类型进行推荐，图书分类是图书特别重要的一部分，他不仅使得图书规范化，直观化同时也可以减少管理员的工作量，具体界面设计如下：

1. 图书分类查询与显示如图5-7所示

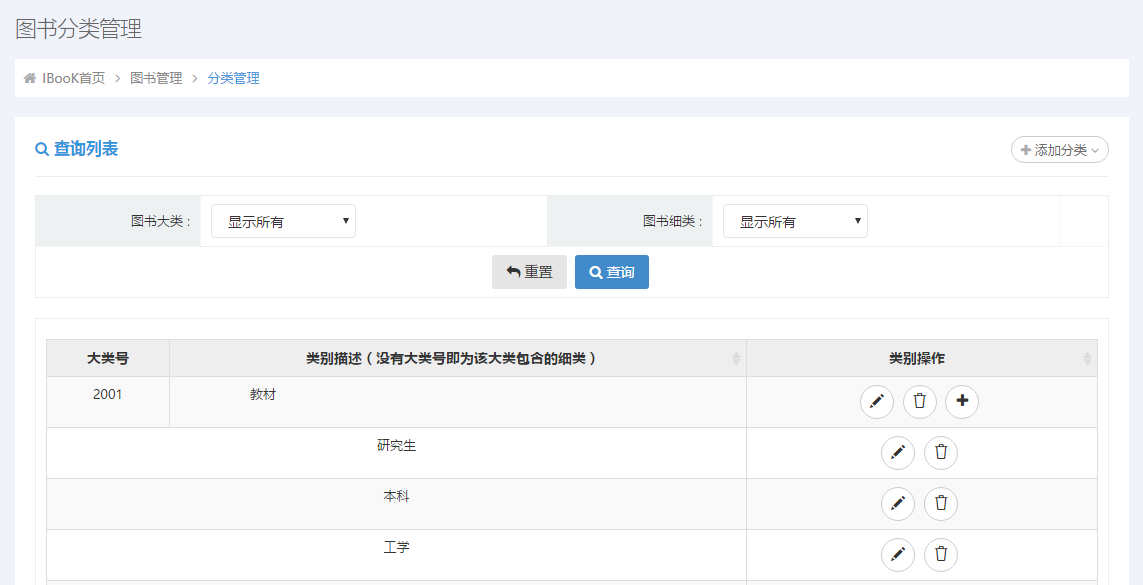


图5-7

1. 图书大类添加界面如图5-8所示



图5-8

1. 图书细类添加界面如图5-9所示



图5-9

1. 图书类别编辑界面如图5-10所示



图5-10

1. 图书类别具体操作如图5-11所示



图5-11

### 5.1.5 图书信息管理模块

图书作为本系统的主要虚拟实体，所有的操作都建立在此基础之上，因此该模块的设计尤为重要，具体的界面如下所示：

1. 图书查询与列表展示 如图5-12所示



图5-12

1. 图书上架界面 如图5-13所示

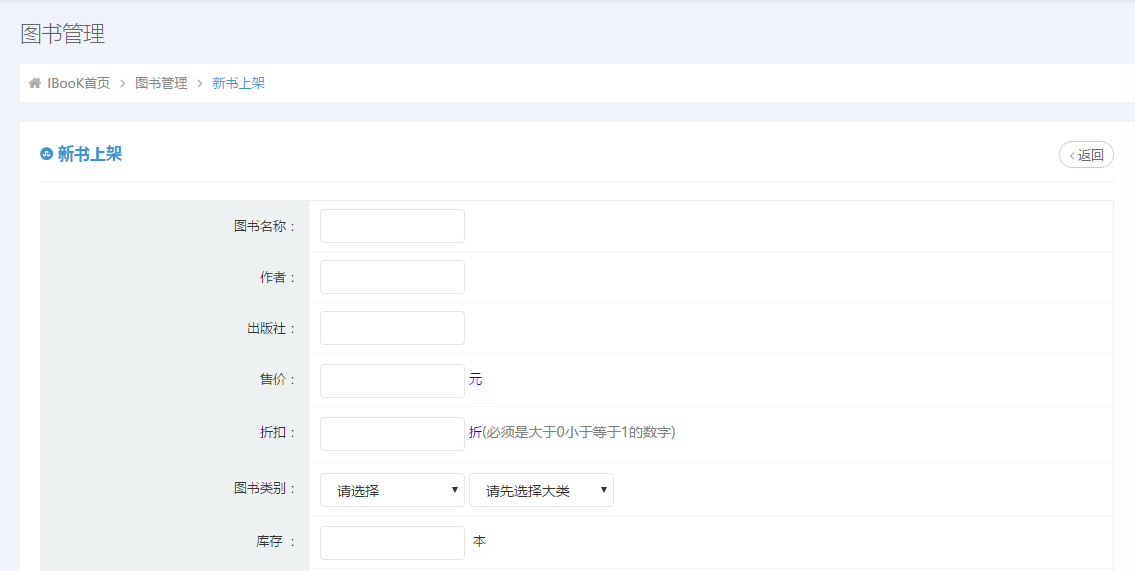






图5-13

1. 图书详情界面 如图5--14所示





图5-14

1. 图书库存修改与图书操作 如图5-15，图5-16所示



图5-15

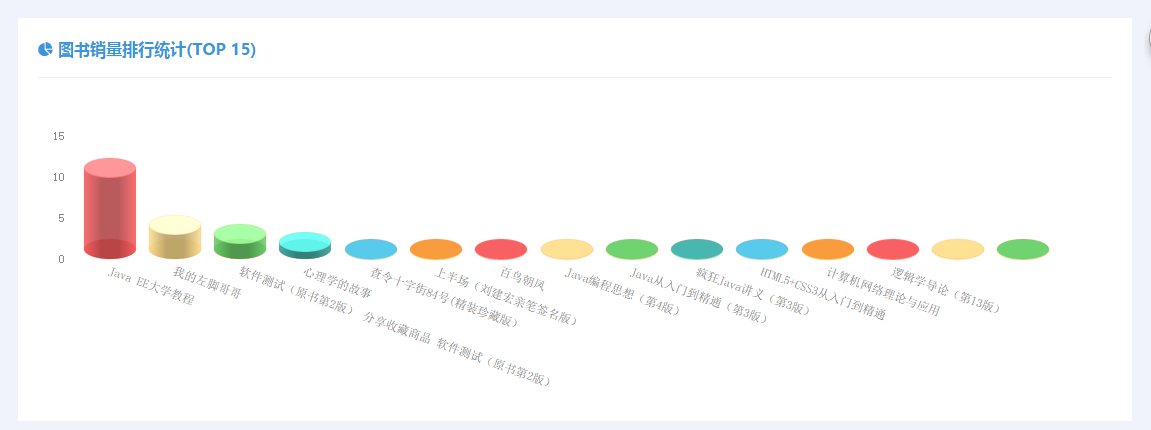


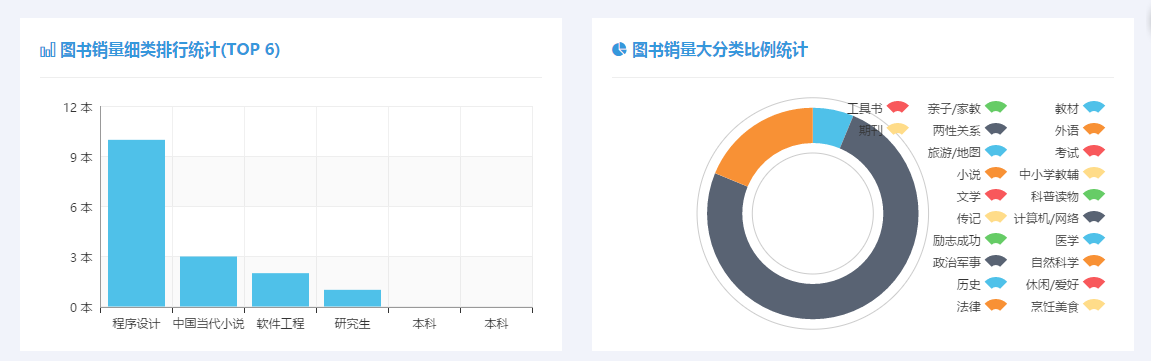
图5-16

### 5.1.6 销量统计模块

销量统计可以很直观的为商家展示近期的销量情况，并且查看统计好的各类需要的信息。具体如图5-17所示：







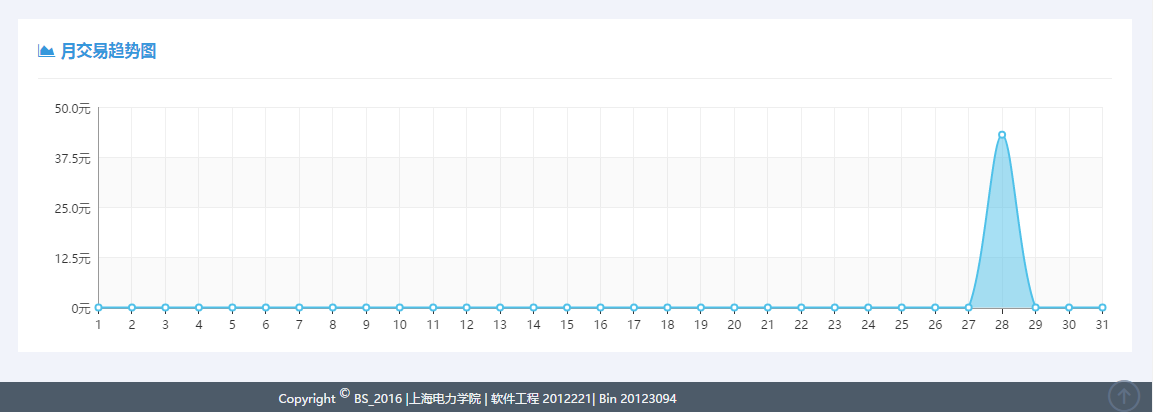


图 5-17

### 5.1.7 智能推荐管理模块

作为本系统与其他系统不同的特点，智能推荐是很重要的一部分内容，不仅可以促进销量而且可以减少用户盲目搜索图书的繁琐步骤。具体设计如图5-18所示

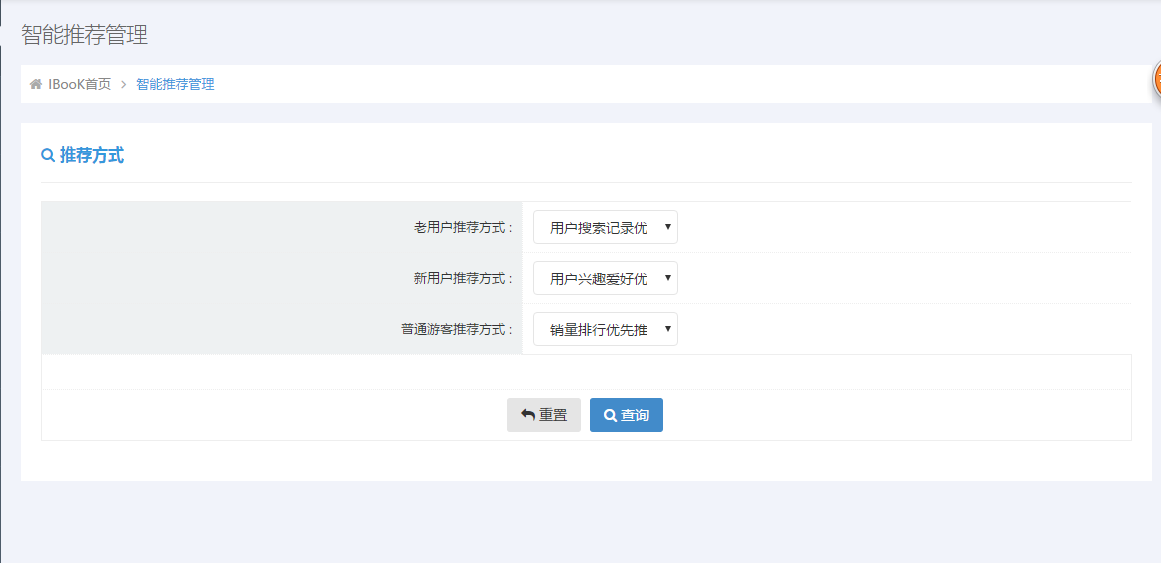


图5-18

### 5.1.8 公告管理模块

公告是系统之中超级管理员给普通管理员分配任务的一种途径，也是卖家发布优惠信息给买家的平台，在购物系统中是很重要的一部分，具体如下：

1. 公告列表查询与显示如图5-19所示

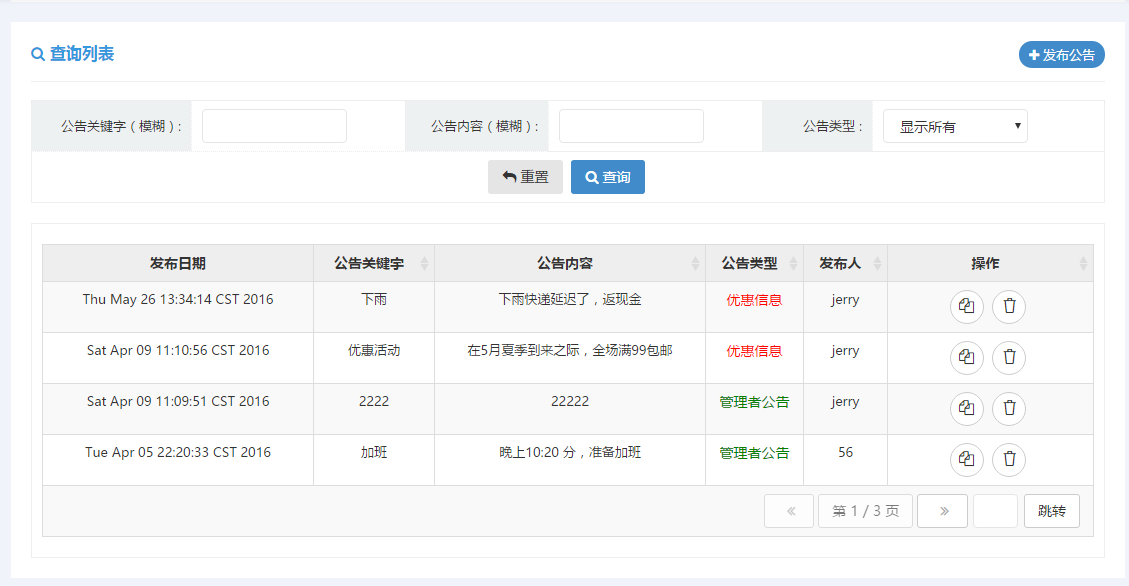


图5-19

1. 公告发布设计如图5-20所示

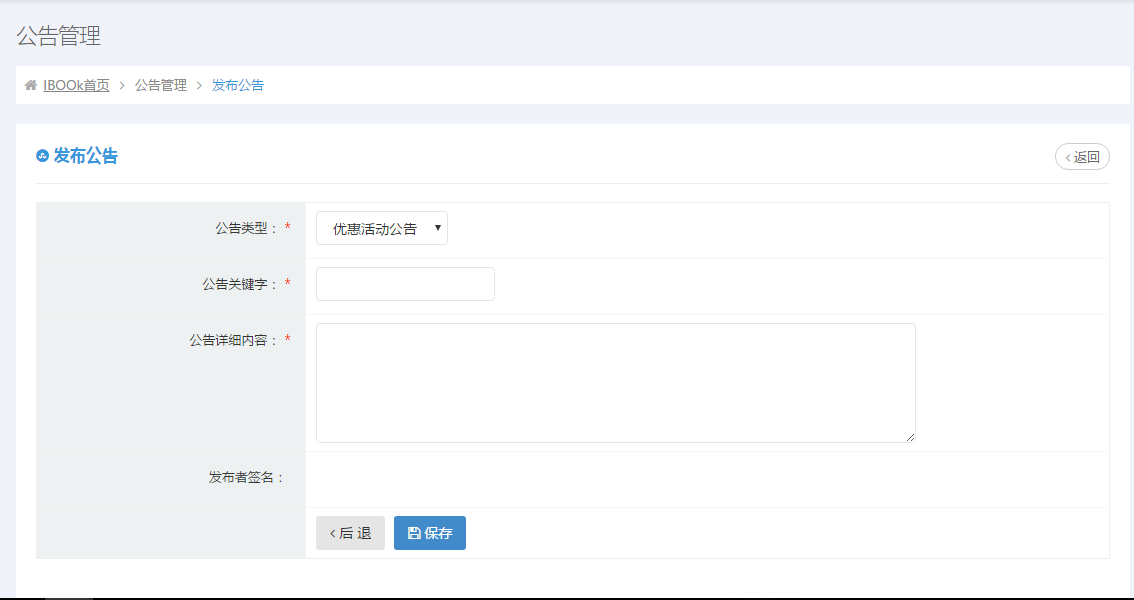


图5-20

核心controller代码如下：

@RequestMapping("/insertNotice")

**public** String toIndex(HttpServletRequest request,Model model,Notice notice) **throws** ParseException{

Date date=**new** Date();

DateFormat dateFormat=**new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date time=dateFormat.parse(dateFormat.format(date));

notice.setDate(time);

noticeService.insert(notice);

**return** "redirect:/notice/getNoticeByPage";

}

### 5.1.9 超级管理员账号与权限管理模块

作为一个后台管理系统，必须考虑系统的安全性，因此作为超级管理员可以随时管理账号与权限，所以本模块是系统不可或缺的一部分，具体设计实现如下：

1. 超级管理员查询与管理界面 如图5-21所示

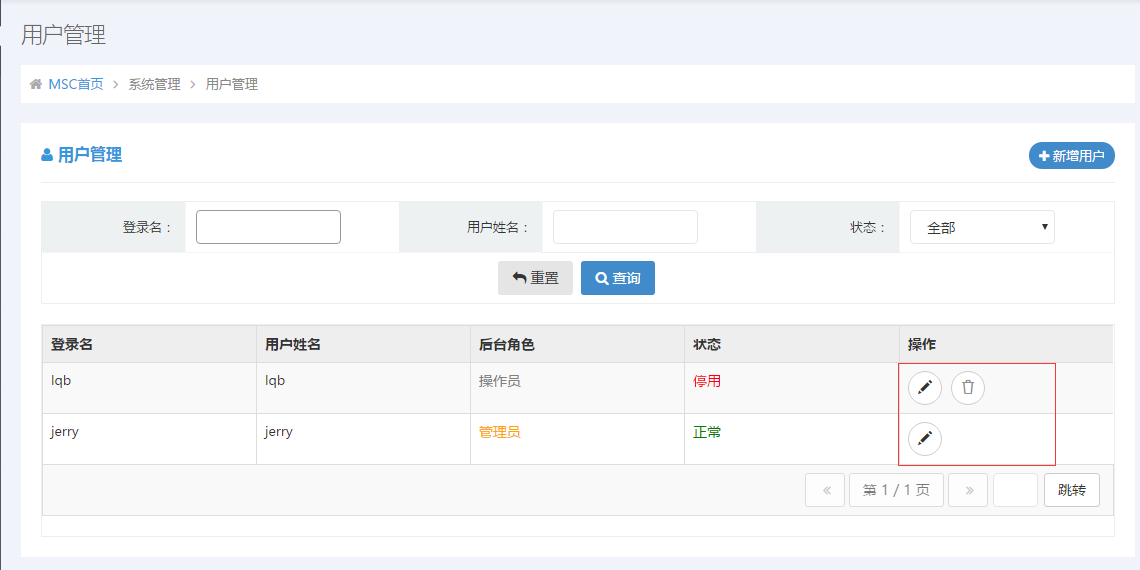


图5-21

1. 超级管理员修改信息界面 如图5-22所示

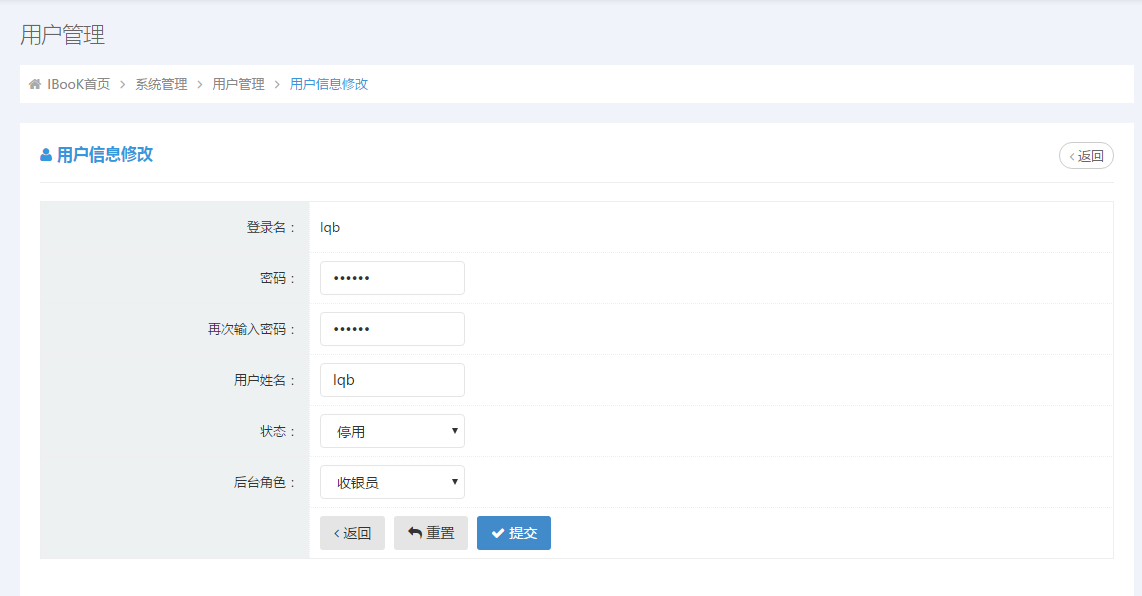


图5-22

1. 超级管理员添加管理员界面 如图5-23所示

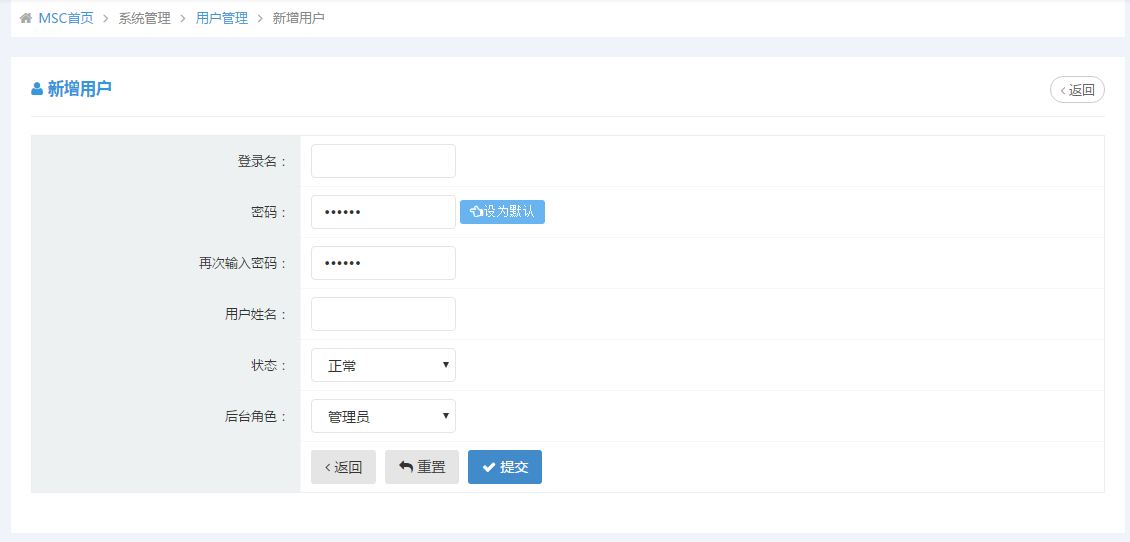


图5-23

### 5.1.10 普通操作员账号与权限管理模块

普通操作员就不可以添加账号和随便更改其他操作员的信息，只能修改个人信息

效果如图5-24所示

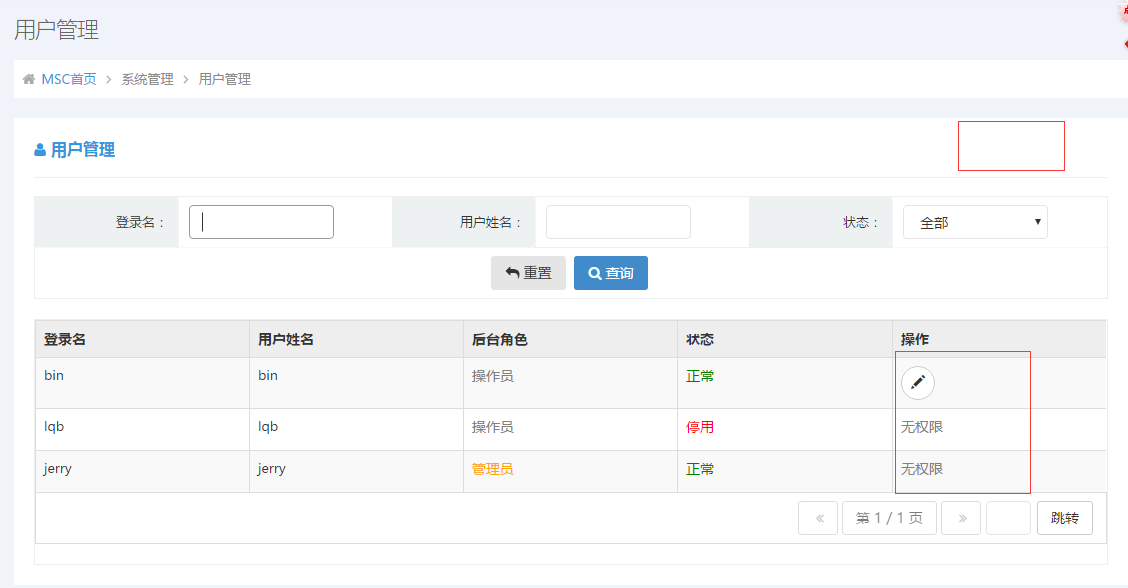


图5-24

### 5.1.11 修改密码模块

为了操作的安全性，在确保旧密码正确的情况下允许管理员自己修改密码 效果如图5-25所示

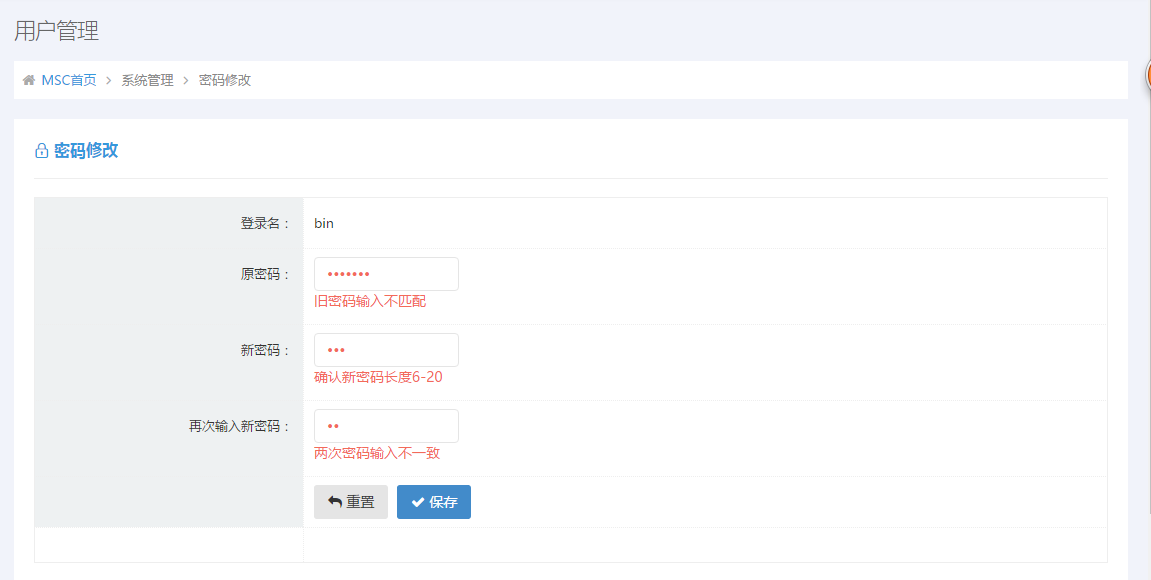


图 5-25

## 5.2 买家（普通用户）模块

### 5.1.1 首页

首页系统的为买家展示销量前三图书，优惠信息，以及新书上架 效果如图5-26所示

图5-26

### 5.1.2 分类列表与搜素

为用户提供查询与选择图书浏览接口 具体实现如图 5-27所示

图5-27

### 5.1.3 图书列表

为用户展示某类别或者查询条件下的所有图书列表 具体实现如图5-28，图5-29

图5-28

图5-29 当没有符合条件图书情况

### 5.1.4 图书详情

为用户展示具体的图书信息，方便用户更加详细的了解图书 具体实现如图5-30 ，图5-31

图 5-30 图书详情

图 5-31 该图书所有评价

### 5.1.5 登录

为方便购物为用户提供的登录入口 具体实现如图5-32所示

图 5-32

### 5.1.6 注册

新用户购物时必须注册，否则无法记录个人信息 具体实现如图5-33所示

图5-33

### 5.1.7 购物车

用户成功登陆系统之后，可以查看个人购物车 具体如图5-34，图5-35

图 5-34 购物车为空时

图 5-35 购物车不为空

### 5.1.8 订单填写

用户在购物时填写个人订单信息 具体实现如图5-36所示

图 5-36

### 5.1.9 个人订单

用户在登录系统之后，可以查看自己的交易订单记录，并且操作与评论，具体实现如图5-38，图5-39，图5-40所示

1. 查看个人订单列表与操作订单 图5-38

图5-38

1. 查看订单详情 图5-39

图5-39

1. 评价订单 具体如图5-40

图5-40

# 6.系统测试

参考文献

【1】[Bootstrap 教程](http://baike.baidu.com/redirect/8ae2CU2wnw3jS4Vr9yHIsf3P62razIdWSXbbjKo4cixAii5pMe72TUw8TBj2Afw8H1CucyN0KJEq8NHjAwsPr5pzCbcCX-egkUuM0N8fT77hRf8XejKnUua7XiM" \t "http://baike.baidu.com/_blank) ．自强学堂[引用日期2015-01-29]

http://www.ziqiangxuetang.com/bootstrap/bootstrap-tutorial.html

【2】[Bootstrap前端框架](http://baike.baidu.com/redirect/7e2bko55_BQ8stjRGezC6xkdBJbpRwfyWebgEwv8wiZ_KWQ79lozXGIo3RjngHyGOxee" \t "http://baike.baidu.com/_blank) ．Bootstrap中文网

http://www.bootcss.com/

【3】[Bootstrap Less](http://baike.baidu.com/redirect/dda8PaMH4-6YIygmLqgZJDmZONKyn-xchdrSIqRvAg5oNioxht6_bqMxsZrJtzCzLuhlhoNpSEMadJ9lNxFg20sxDzWukWRJohtAR7m7jCFbAg" \t "http://baike.baidu.com/_blank) ．w3cschool菜鸟教程．2013-08-30[引用日期2014-01-20]

<http://www.runoob.com/bootstrap/bootstrap-less.html>

[4]曹哲.软件工程[M].北京:中国水利水电出版社,2004.

[5]萨师煊,王珊.数据库系统概论[M].北京:高等教育出版社,2005.