

学生姓名： 尚学

用户名： shangxue

所属服务站： 北京教学服务中心

指导教师： 张三老师

“未来书店”图书销售管理信息系统设计

【摘要】:

随着互联网与电子商务的飞速崛起，网上书店应运而生，凭借价格便宜、选择多样、购买方便等优势快速的抢占图书市场，给实体书店带来了巨大的冲击；另一方面持续上涨的房租、高额的人力成本，资金的短缺等因素都在桎梏实体书店的发展，特别是对于面向个人的中小型书店，这些因素的影响尤其突出。本文针对该趋势，从实体书店和消费者需求出发，对图书销售管理信息系统进行设计和开发。

本文图书销售管理信息系统包含了前端销售系统和后台管理系统两大部分。其中前端销售系统，供消费者使用，包含功能有网站用户注册/登录，浏览网站，查找图书，购买图书，查看订单；后台管理系统，供书店人员使用，包含有图书管理，供货商管理，消费者管理，销售订单管理，销售评价管理，销售额管理，销售报表查询，网站通知管理，系统设置。

本文系统基于 Java 技术，采用主流 SSM 框架（Spring、Spring MVC、MyBatis）开发，数据库采用 MySQL，易操做，平台适应性强。通过系统调试工作，验证其具有良好的操作性和稳定的运行性。

【关键词】: 图书销售管理； Java ； MySQL

【正文】:

1 绪论

1.1 选题背景

我国进入互联网时代已有很多年，互联网的发展在行业的变革和人们生活习惯的改变方面做出了很大的贡献。未来人们的生活和工作将越来越依赖于数字技术的发展，越来越数字化、网络化、电子化、虚拟化。在当前“互联网+”概念冲击下传统的商业模式开始发生质的变化，从单纯的依靠线下销售，开始逐渐的形成了线下和线上相结合的购物模式，更多的商家将销售的侧重点放在了网上销售，这也就促使了互联网金融模式的诞生、发展和日渐成熟。

截至 2021 年，“全民阅读”已经连续八年写入政府工作报告，报告指出要推动公共文化服务体系一体化，创建实施文化惠民工程，进而倡导全民阅读，掀起新时代大众阅读热潮。过去一年中，中国图书阅读率近 60%，阅读已经渗透入大众的生活，疫情也进一步激发了读者的读书热情，调研显示，疫情期间中国读者阅读量提升人数占比遥遥领先。当前国内已经有较多的成熟网上购书网站，如“京东图书”，“天猫图书”，“当当网”，“新华书店”等。2021 年 8 月第一财经商业数据中心发布《2021 图书消费半年报告》，报告显示，线上图书零售持续增长，过去半年，消费者在天猫人均买书 4.6 本，阅读人口增速是去年同期的 4 倍。同时国内因为有支付宝和微信支付以及银行卡支付，有了相对完善的金融结算方式，且前国内物流的发展已经远比国外物流的送货速度快，促使国内网络购物需求不断增加(注 1)。

2020 年初疫情来袭，给整个实体行业都带来了重击，图书市场也不例外，特别是面向个人的中小型书实体店，反观网络购物市场，在应对社会性突发问题上优势体现明显，这也必然会加速中小型实体书店转型进度。虽然像“天猫”，“京东”，“当当”这种大型网站都支持商家入驻，但因这些网站都有自营书店，不管是在图书价格，活动力度，还是引流力度，入驻商家基本都无法比及，因此入驻无法给中小型的实体书店提供太多的经济效益，同时高昂的入驻费用，广告费用还会增加更多成本(注 2)。

1.2 选题意义

本文的意义是为中小型实体书店提供一个好的平台，帮助其实现线上+线下的销售模式，同时实现销售信息数字化管理，帮助中小型实体书店转型。其中前端销售系统，书店可以直接展示图书

信息，活动信息等，实现销售自动化和线上营销，另一方面消费者也可在网站直接查看购买书店书籍，大大减少了到实体店购买的成本；后端管理系统，书店可以使用图书管理，供货商管理将图书基本信息，类别，库存信息数字化管理，使用消费者管理，销售订单管理，可以将消费者信息，销售订单信息数字化管理；通过通知管理发布书店最新动态，实现线上营销和引流，通过销售评价管理，销售额管理，以及销售报表分析帮助书店做各类经营数据分析，调整书店营销方向，进而促进中小型实体书店的良性循环发展。

1.3 毕业设计主要工作

图书销售管理信息系统的主要工作内容有以下几个方面：材料收集调研，确定技术方案、专研技术，系统分析、设计，系统实现，论文写作。

1.3.1 材料收集调研

材料收集调研是设计工作的基础，要通过查阅书籍，网络搜索等方式，了解选题背景，确认选题对象，明确选题意义。

1.3.2 确定技术方案、专研技术

确定技术方案、专研技术是系统实现的技术基础，要明确系统开发使用的语言，采用的技术框架和数据库，考虑系统的可扩展性。

1.3.3 系统分析、设计

系统分析是项目启动的基础，主要包含可行性分析，需求分析。可行性分析主要有两部分组成，即技术可行性，经济可行性。需求分析是系统设计工作的前提，直接决定了后续的功能设计和实现，主要有四部分组成，即业务需求，用户需求，功能需求和性能需求。系统设计是系统实现的基础，包含总体功能设计，数据库设计，各模块详细设计。

1.3.4 系统实现

系统实现是根据系统设计方案搭建开发环境，建立好相对应的数据库、表，然后进行框架的搭建。在系统框架搭建完成后进行功能的开发，包括前台界面、后台程序等内容。

1.3.5 论文写作

论文写作是对系统设计的一个总结，描述调研的系统背景、意义；描述确定的技术方案，描述系统分析和设计，展现系统实现结果，总结系统开发中的问题。

2 系统开发关键技术

2.1 采用 SSM 框架技术完成后端功能开发

SSM 框架是由 Spring、Spring MVC、MyBatis 三个开源框架组合而成。Spring 框架主要优势是分层架构，通过 Spring 框架进行项目中的分层管理，将 Spring MVC 框架和 MyBatis 框架整合在一起，建立一个连贯的体系。Spring MVC 框架实现了 Web MVC 设计模式的请求驱动型框架，将简化 Web 层，提高程序解耦。MyBatis 框架是一种半自动化 ORM 框架，灵活支持 SQL 语句。

2.1.1 采用 Spring 框架完成业务对象的管理

Spring 是一个开放源代码、轻量级的 Java 开发框架。其对常用的 API 都做了一些简化和封装（比如，用 spring jdbc 访问数据库，就不用考虑如何获取连接和关闭连接）；通过 Spring 提供的 IoC 容器，我们可以将对象之间的依赖关系交由 Spring 进行控制，避免硬编码所造成的过度程序耦合；Spring 可以将其他框架集成进来，比如我们之后要介绍的 Mybatis 框架。Spring 容器是 spring 框架中的一个核心模块，用来管理对象（包括对象的创建，销毁和初始化等）。它帮助完成类的初始化与装配工作，让开发者从这些底层类的实例化，依赖关系装配类的工作中脱离出来，专注于更有意义的业务逻辑开发工作。在使用过程中，我们只需要在配置文件加上以下配置：

```
<context:component-scan base-package="指定包" ></context:component-scan>
```

Spring 容器启动之后会检查指定包以及子包下面的所有的类，如果类前面有特定注解（比如 @Component, @Named, @Repository, @Service, @Controller），则容器会将该类纳入容器进行管理（就相当于配置了一个 bean 元素）。在本文系统中，主要配置使用如下

```
<!—自动注入 com.bookshop 下的 bean -->
```

```
<context:component-scan base-package="com.bookshop" />
```

```
<!-- 导入配置文件 log4j, mybatis-->
<util:properties id="configProperties" location="classpath:*.properties" />
<context:property-placeholder
    properties-ref="configProperties" ignore-unresolvable="true" />
<!-- 事务管理器 -->
<bean id="transactionManager"
    class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource" />
</bean>
<!-- 用注解来实现事务管理 -->
<tx:annotation-driven transaction-manager="transactionManager" proxy-target-
class="true"/>
```

2.1.2 采用 Spring MVC 框架来负责请求的转发和视图管理

Spring MVC 是 Spring 框架最重要的模块之一，借助 Spring IOC 简化框架配置，是 Spring 为用户开发提供的基于 MVC 的主流 Web 框架。SpringMVC 主要组件有前端控制器，处理器映射器，处理器适配器，处理器，试图解析器和视图。通过处理器映射器来选择使用哪个控制器来处理请求，选择结果通过试图解析器处理结果并解析。在系统中起到请求地址分发处理的作用，将前端用户提交的操作数据及请求路径交给后端配置的相应的处理方法，完成数据处理后返回给对应的试图或者对应格式的数据。核心的处理方式是注解的形式引入，常用的有以下几种：@RequestMapping 注解进行请求地址的分发；@Controller 注解标注相应处理类为控制类，通过 Spring 的自动扫描功能加入实例池共后续调用；@Service 注解标注该类为对应的逻辑实现类，通过接口的方式解耦，在需要的时候注入到相应的类中使用；@Mapper 注解标注 dao 层类与数据源绑定的接口类；@ResponseBody 注解标注处理方法返回的数据格式为 json 对象。在本文系统中，使用配置如下

```
<!-- 注解驱动 -->
<mvc:annotation-driven/>
<!-- 定义 Controller 扫描包 -->
<context:component-scan base-package="com.bookshop.controller"/>
<!-- 视图解释类 -->
<bean
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/page/" />
    <property name="suffix" value=".jsp" />
    <property name="viewClass" value="org.springframework.web.servlet.view.JstlView" />
</bean>
```

2.1.3 采用 MyBatis 框架来作为数据对象的持久化引擎

MyBatis 是一款实现实体类与 SQL 之间映射的半自动 ORM 持久层框架，支持自定义 SQL、存储过程以及高级映射，免除了几乎所有的 JDBC 代码以及设置参数和获取结果集的工作。可以通过简单的 XML 或注解来配置和映射原始类型、接口和 Java POJO 为数据库中的记录。同时其可以使用 mybatis-generator-core.jar 等工具包根据数据库表自动生成 DAO 接口类，PO 实体类以及对应映射文件，实现实体类与数据库表的基本增、删、改、查操作，节省了繁琐的基本代码重复编写，提高开发效率，可以集成到 Spring，在本系统中主要使用的配置如下：

```
<!-- Mybatis 配置 -->
<bean id="sessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource" />
    <property name="configLocation" value="classpath:conf/mybatis/mybatis-config.xml"
```

```

/>
    <property name="mapperLocations" value="classpath:conf/mybatis/mapping/**/*.xml" />
</bean>
<bean id="sessiontemplate" class="org.mybatis.spring.SqlSessionTemplate">
    <constructor-arg index="0" ref="sessionFactory" />
</bean>

<!-- 设定 Mybatis dao 包路径 -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
    <property name="basePackage" value="com.bookshop.dao"/>
    <property name="sqlSessionFactoryBeanName" value="sessionFactory" />
</bean>

```

2.2 采用 MySQL 数据库进行数据管理

数据库是计算机应用系统中的一种专门管理数据资源的系统。数据有多种形式，如文字、数码、符号、图形、图像以及声音等。数据是所有计算机系统所要处理的对象。人们所熟知的一种处理办法是制作文件，即将处理过程编成程序文件，将所涉及的数据按程序要求组织成数据文件，用程序文件来调用。数据文件与程序文件保持着一定的对应关系。在计算机应用迅速发展情况下，这种文件式方法便显出不足。比如，它使得数据通用性差，不便于移植，在不同文件中存储大量重复信息、浪费存储空间、更新不便等。数据库系统便能解决上述问题。数据库系统不从具体的应用程序出发，而是立足于数据本身的管理，它将所有数据保存在数据库中，进行科学的组织，并借助于数据库管理系统，以它为中介，与各种应用程序或应用系统接口，使之能方便地使用数据库中的数据。

现在市面有多种主流数据库，如 Oracle 数据库，Microsoft SQL Server 数据库，MySQL 数据库，与前两种相比，MySQL 非常容易学习和掌握，因为是关系数据库管理系统，将数据保存不同的表中，增加了速度并提高了灵活性，同时 MySQL 是开源软件，可以免费使用，扩展性好，如果需要，还可以更改其源代码或进行二次开发。它支持标准的 SQL 语言，可以在各种平台上运行，如 UNIX，Linux，Windows 等，故不管在小型还是大型应用程序中，都是理想的选择。

2.3 采用 jQuery 完成前端开发

jQuery 是继 prototype 之后又一个优秀的轻量级 JavaScript 框架。其宗旨是——“Write Less, Do More”，写更少的代码，做更多的事情。它是一个快速和简洁的 JavaScript 库，可以简化 HTML 文档元素的遍历，事件处理，动画和 Ajax 交互以实现快速 Web 开发，它被设计用来改变编写 JavaScript 脚本的方式。文档非常丰富，因为其轻量级的特性，文档并不复杂，随着新版本的发布，可以很快被翻译成多种语言，这也为 jQuery 的流行提供了条件。其最有特色的语法特点就是与 CSS 语法相似的选择器，并且它支持 CSS1 到 CSS3 的几乎所有选择器，并兼容所有主流浏览器，这为快速访问 DOM 提供了方便，加之其简单易学，已为开发网站等复杂度较低的 Web 应用程序的首选 JavaScript 库。

2.4 采用 Tomcat 服务器部署系统

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，且技术先进，性能稳定，是开发和调试 jsp 程序的首选。

2.5 采用 IDEA 完成程序集成开发

IDEA 集成开发工具是被公认为最好的 java 开发工具之一，尤其在代码自动提示、重构、动态语法检测，代码检查，JavaEE 支持、各类版本工具等方面的功能可以说是超常的，可以有效提高代码规范度，提交开发效率等。

3 系统分析

3.1 可行性分析

可行性分析是任何一项系统研究投入力量之前必须进行的一项工作，这对于保证资源的合理使用、避免浪费是十分必要的，也是系统设计一旦开始以后能顺利进行的必要保证。

3.1.1 技术可行性

本系统采用 SSM 框架，开源的 My SQL 数据库引擎，jQuery 框架，Tomcat 服务器，IDEA 集成开发工具进行设计开发。通过对以上技术的学习了解和实际开发经验，本系统可以采用这些技术实现，同时本系统的开发对计算机的要求也不是太高，具有使用的普遍性，因此技术上是可行的。

3.1.2 经济可行性

通过对图书进行线上销售，同时针对销售信息进行数字化的管理和统计分析，提高书店运营效率，同时能够节约一定的人力成本，从而提高书店经济效益。本系统的开发所使用的软件都是开源的，免费的，能够节省一大笔开发资金，投入不大，因此在经济上是可行的。

3.2 需求分析

需求分析，是开发人员经过深入细致的调研和分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求表述转化为完整的需求定义，从而确定系统必须做什么的过程。

是软件计划阶段的重要活动，也是软件生存周期中的一个重要环节。目标是把用户对待开发软件提出的“要求”或“需要”进行分析与整理，确认后形成描述完整、清晰与规范的文档，确定软件需要实现哪些功能，完成哪些工作，本文将从业务需求，用户需求，功能需求，性能需求四个方面来进行分析。

3.2.1 业务需求

业务需求主要是描述为什么要开发这个系统，即希望达到的目标。

设计一款针对中小型综合性实体书店的图书销售信息管理系统，打通网上销售渠道，增加跟消费者之间的粘合度，帮助实体书店进入线上市场，给实体书店引流，创造更多的经济效益；建立销售信息资料库，数字化管理，提高销售管理效率，同时支持根据各种维度分析销售情况，方便书店确定下一步的运营方向，提高经济收益，促进书店进一步发展。

3.2.2 用户需求

用户需求描述的是用户要求系统必须能完成的任务，即描述了用户能使用系统来做些什么。本节主要从用户角度出发，明确图书销售信息系统的用户类型，以及各类用户需要系统实现的具体功能。

图书销售信息管理系统的使用用户主要有两大类：消费者用户，书店人员。

消费者用户：游客，普通用户，VIP 用户；

书店人员：采购人员，营销人员，业务经理，店长；其中采购人员为商品管理员角色，营销人员对应销售角色，业务经理对应销售经理角色，店长对应超级管理员角色。

本文将针对两大类角色来进行用户需求分析。

消费者用户—游客，即未注册或未登录的用户。在进入商城后首先有对商城有一个初步的了解需求，也就是说要可以浏览网站基本信息，了解网站最新动态；进入商城就是有购买需求，故必须可以查看网站所有商品，以及商品详情，也可以对感兴趣商品通过关键字或者目录种类查询商品，可以直接商品添加至购物车。但是不能生成订单，如果他们想进一步使用网站的功能就必须登录或者注册。游客用例图如图 3-1 所示。

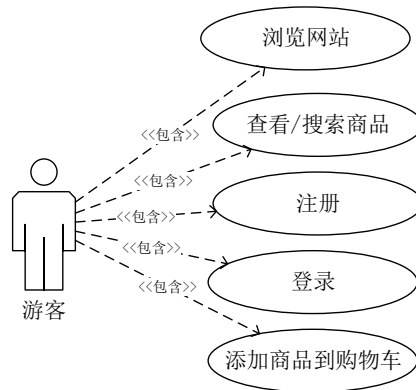


图 3-1 游客用例图

消费者用户—普通用户，即在网站注册过，拥有网站账号的人。如果用户未登录，则跟游客一样的访问权限，当用户登录商城后，除去游客的需求外，会有购买商品，查看订单，维护个人信息，优惠活动，和售后服务的需求。故需要能直接进行购买商品；可以查看个人所有的订单信息，如对已完成的订单进行评价，删除订单，取消未发货订单等；可以查看个人信息，修改个人信息，包括姓名，邮箱，生日，收货地址等； 优惠活动的需求是必然的，也是吸引用户消费的一种手段，同时优良的售后服务，会让用户感觉到被重视，对于维护客户有至关重要的作用。用户用例图如图 3-2 所示。

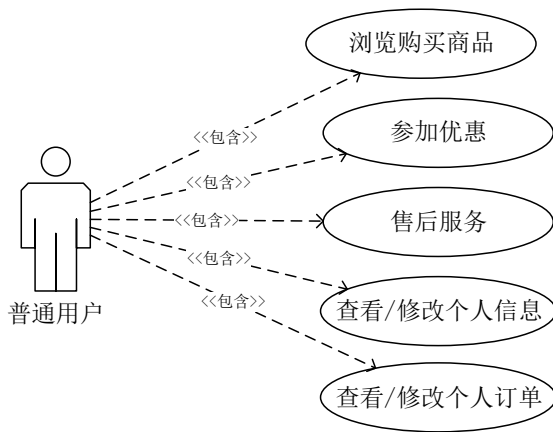


图 3-2 普通用户用例图

消费者用户—VIP 用户，即普通用户一年内在网站消费达到 1000 元以上的客户，并已升级为VIP，此类客户属于网站的优质用户，为了维护优质用户持续消费，并推动普通用户转换为优质用户，VIP 用户，可在购买图书时默认享受 8 折优惠，VIP 有效期 1 年。VIP 用户用例图如 3-3 所示。

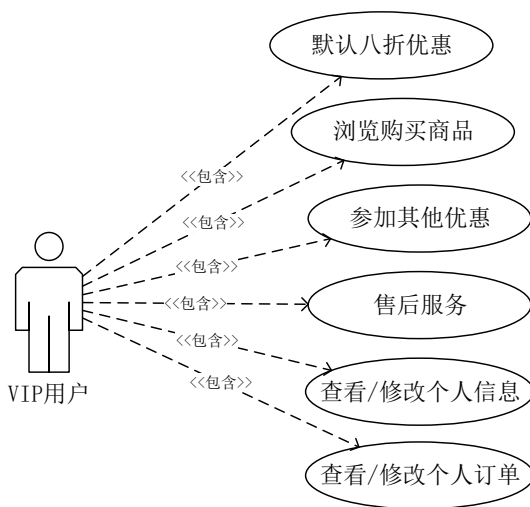


图 3-3 VIP 用户用例图

超级管理员，即图书销售管理系统的最高管理员。拥有后台管理系统所有权限，用例图如图 3-4 所示。

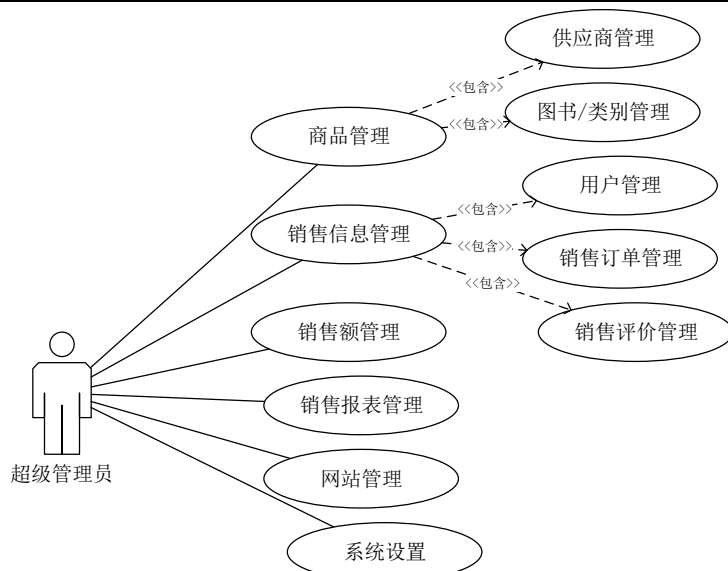


图 3-4 超级管理员用例图

商品管理员，主要对供应商和图书信息，和图书类别进行管理，可以添加，修改供应商信息，当跟供应商不再续约时，可禁用供应商；可查看售卖的图书信息、根据特定条件搜索图书，修改图书信息，来新图书时要上架新图书，下架已有图书，补充图书库存；可以管理图书分类，查看、搜索图书类别条目，添加新类别，删除已有类别，商品管理员用例图如 3-5 所示。

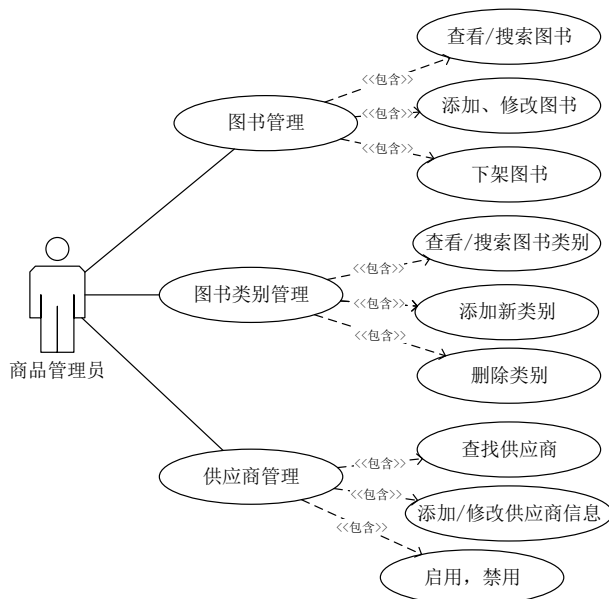


图 3-5 商品管理员用例图

销售，主要负责管理网站消费者用户，销售订单和销售评价，可以查看网站所有的消费者用户，能根据条件搜索消费者用户，可以直接添加新消费者用户，修正已有用户信息，直接将用户置为无效状态，同时可直接修改用户的 VIP 状态；可以对销售订单情况跟进，维护所有状态的订单，协助用户操作订单。可以查看所有订单信息，对未发货的订单发货后进行标记；对于交易未结束的订单可以进行取消订单的操作，可以查看所有订单评价，处理评价，销售用例图如 3-6 所示

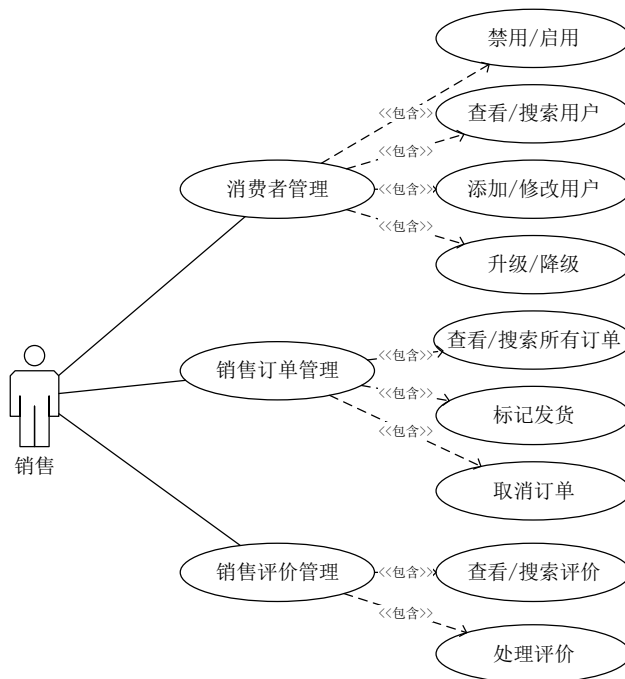


图 3-6 销售用例图

销售经理，销售经理除了拥有销售角色的权限外，可以进行销售额管理，和销售报表查询。其中销售额可以从时间维度（按日，月，季，年等），图书维度（包括图书名称，图书类别，供应商等）进行汇总；报表中可以查看图书销售排行榜（按时间，按图书类别，按供应商等），可以查看消费者分析（按年龄段，按购买月份，收货地址，分析各类图书购买情况），销售经理用例图如 3-7 所示

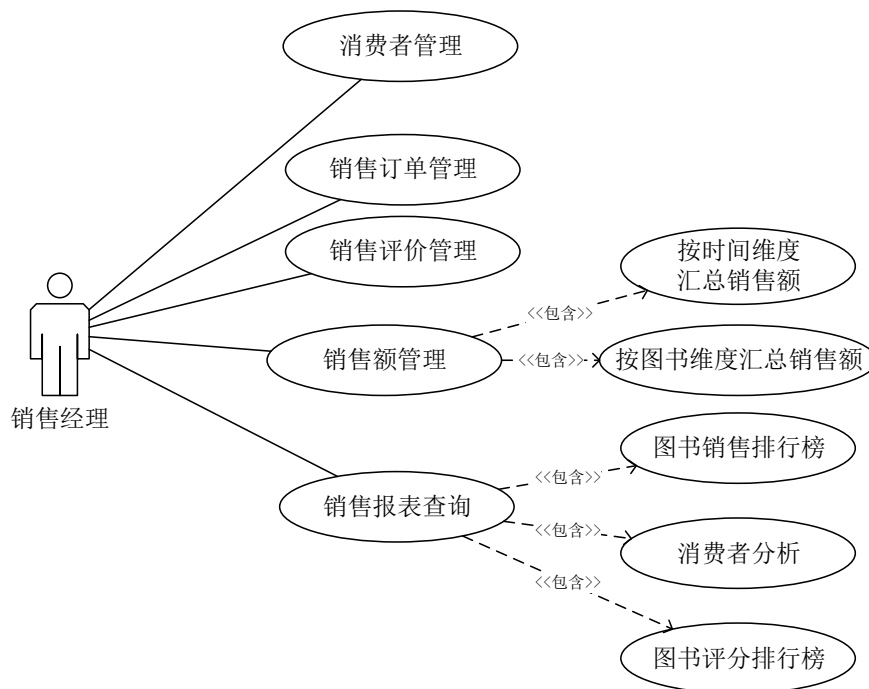


图 3-7 销售经理用例图

3.2.3 功能需求

功能需求是指规定开发人员必须在产品中实现的软件功能，用户利用这些功能来完成任务，满足业务需求，本章节通过对业务需求和用户需求的分析，详细的描述了系统具体要实现的功能。根据分析将功能需求分为两类：前端面向消费者用户的功能需求，后端面向书店人员的功能需求。

1. 前端面向消费者的功能需求

浏览网站：不管用户是否已登录，都可以在网站进行浏览，且能浏览到网站所有公开内容；

搜索/查看商品：不管用户是否登录，打开网站后，可以查看任意商品的基本信息和详情，同时也可以根据兴趣进行商品搜索；可将商品添加到购物车。

购物车管理：不管用户是否登录，都可以在购物车中，修改商品数量，或者删除商品，查看商品总金额。

注册：顾客要注册为网站的用户。注册时需要填写用户名，密码，邮箱，手机号，性别，出生日期，其中用户名，手机号，邮箱，密码必填，且手机号必须唯一。

登录：顾客注册成为商城用户后，可以登录网站，可以在网站进行购买商品，管理个人订单，维护个人信息，修改登录密码，管理收货地址。

购买：用户在购物车点击结算，或者在商品详情页面直接点击“立即购买”，要校验是否登录，如果未登录则跳到登录页面，如果登录跳转到订单详情页面提交订单，要校验是否有收货地址，如果没有提示添加收货地址，收货地址会自动进入收货地址管理，提交后进入支付页面，可放弃支付，可完成支付，支付完成购买完成。提交订单后，商品库存要响应减少，网站后台会进行发货和结算。

个人信息管理：用户登录后，可以查看个人信息，修改邮箱，手机号，性别，出生日期；也可修改密码；

个人收货地址管理：一个人可以有多个收货地址，地址可以从收货地址管理中添加，也可以在下单的时候直接录入；同时可以对自己的收货地址管理，进行增删改查。

个人订单管理：购买后会生成对应的订单，对与未支付的订单可进行再次支付，取消订单，删除订单；对于已支付的订单，可进行确认收货，售后申请，商品评价，删除订单。

2. 后端面向书店人员的功能需求

商品管理：权限角色有商品管理员，超级管理员，可以管理供应商，图书和图书类别，可以进行查看、修改、删除、查询商品信息等操作，并且可以改变商品上架下架状态，商品只有上架后，在前台才能被商城用户浏览。可以对商品的类别进行查看、修改、删除。可以查看，修改，禁用启用供应商，可以根据关键词对商品信息，商品类别，供应商进行检索。

消费者管理：权限角色有销售，销售经理，超级管理员。可以进行查看、修改、删除、查询商城用户的相关信息的操作，也可以改变用户的状态，对用户账号进行暂停或者启用；可以查询所有商城用户的积分信息，将商城用户升级为VIP或者将VIP用户降级，可以查看用户积分使用历史。可以通过关键词检索用户信息。

销售订单管理。权限角色有销售，销售经理，超级管理员。可以直接查看所有订单，对未处理的商品订单进行发货处理，当用户有需求要取消订单，也可以协助取消。

销售评价管理。权限角色有销售，销售经理，超级管理员。可以查看所有订单评价，处理评价，修改评价处理进度。

销售额管理。权限角色有销售经理，超级管理员。可以根据图书类别，图书书名，供应商，按照时间维度（日，周，月，季度，年）查看汇总销售额。显示内容要有图书基本信息，包含类别，书名，出版社，供应商，销售数量，成本价，销售价，成本汇总，销售金额汇总。

销售报表分析。权限角色有销售经理，超级管理员。可以查看销售图书销售排行榜，消费者分析，图书评分排行榜。销售图书排行榜，可按照时间维度（日，周，月，季度，年）进行整体排行，也可根据图书类别排行。消费者分析，可以根据消费者年龄段，购买月份，收货地址所属地区查看各类别图书销售情况。图书评分排行榜，可以根据图书类别查找图书评分排行榜。

网站管理。权限角色有超级管理员，可以查看、修改、发布公告，并可根据关键词检索公告。

系统设置。权限角色有超级管理员，管理后台系统所有账号，添加新账号，也可对账号信息进行修改，比如修改密码、修改角色，禁用账号等操作，也可以用户名为关键词进行检索。

3.2.4 性能需求

为了保持系统能够长期的，稳定的，高效的运行，系统要有以下几点。

1. 简单易用

图书销售系统面对的客户群体，可以是能够快速接收任何新事物年轻人，也可以在慢慢适应网络购物的老年人，所以就要求系统交互必须直观，且易操作。

2. 可快速扩展

系统开发中，必须考量系统的扩展性。在整体架构不变的情况下，能快速响应新需求。

3. 安全性

网上购物因会涉及到用户的隐私信息，例如电话，邮箱，地址，订单等等，因此网站必须具备保证数据安全的能力。

4. 系统设计

4.1 总体功能设计

4.1.1 总体功能结构

本系统采用前后台分开的设计。其中前端销售系统主要是面对消费者用户应用，通过浏览器的模式展示出来，让消费者用户可以进行购买。后台管理系统主要是针对书店人员，实现对商城各模块的管理和对销售数据的维护分析。系统的总体功能结构如图 4-1 所示：

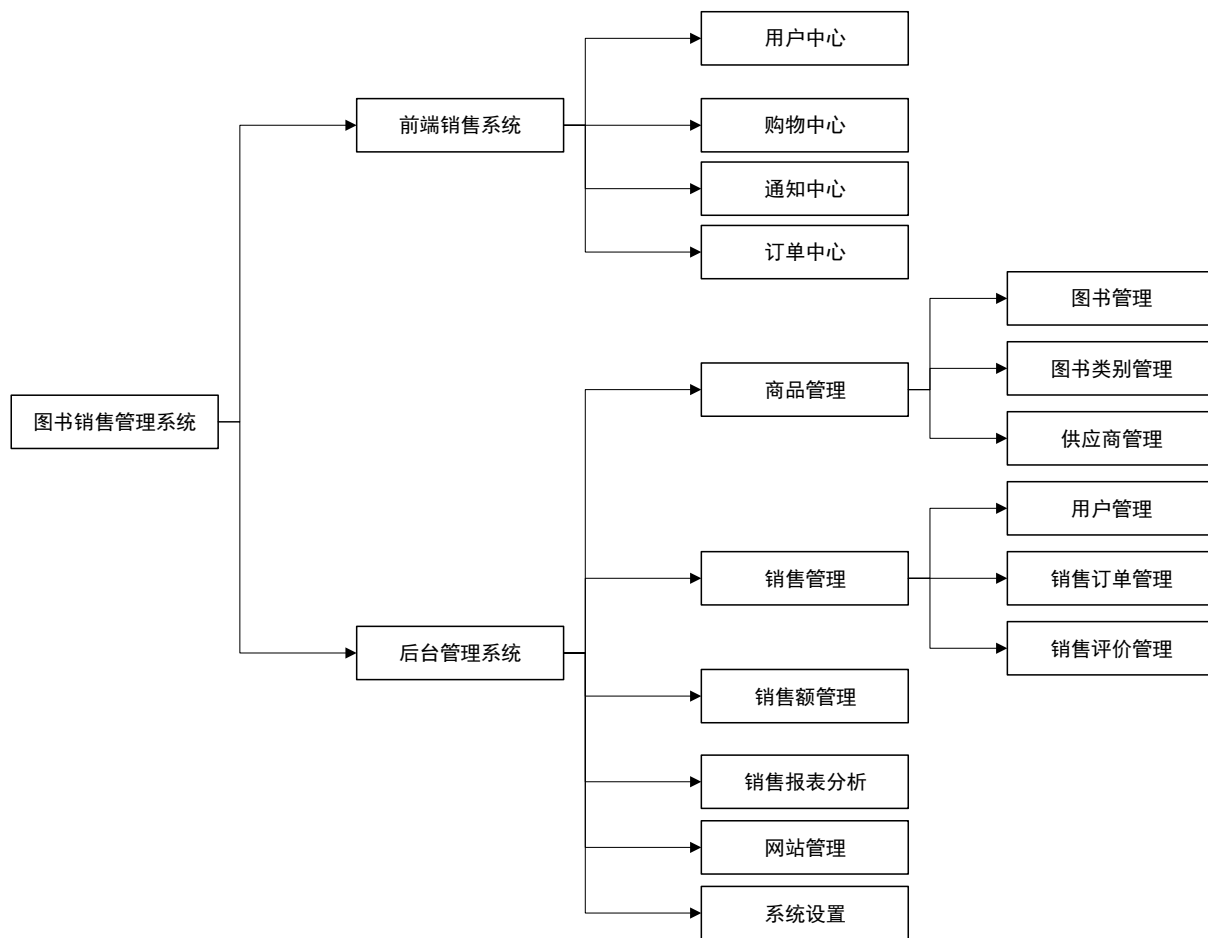


图 4-1 总体功能结构图

4.1.2 各模块功能

1. 前端销售系统—用户中心模块。该模块包含注册，登录，个人信息管理等功能。未注册游客可以通过注册页面进行注册，登录网站后，进入用户中心，可以查看个人基本信息，修改个人基本信息，修改密码；查看个人积分情况；管理个人收货地址，添加新的收货地址，修改或删除已有收货地址。

2. 前端销售系统—通知中心模块。该模块主要是在前端展示网站的最新通知，只要访问网站都可看到，不论访问人是游客还是用户。

3. 前端销售系统—购物中心模块。该模块包含了搜索商品，查看商品，购买商品，购物车管理，

商品评价功能。不管是游客还是用户，都可以根据关键字搜索商品，也可以根据分类查询商品，查看商品基本信息和商品详情，且都可以将商品添加至购物车。购物车管理包含查看购物车，修改购物车中商品数量，删除购物车商品，计算购物车中已选中商品总金额。购买商品提交时是否登录，如果未登录跳转到的登录页面，如果已登录则跳转到订单页面。

4. 前端销售系统—订单中心模块。该模块包含了订单管理和商品评价管理。订单管理主要是查看订—查看已有订单详情；取消订—对于还未支付的订单，如果不想继续购买时可取消订；修改订—对于还未出库的订单，可以修改收货地址；删除订—对于已完成或已取消的订—删除，订—支—对于未付款的订—，可进行支付。商品评价管理，对于已完成的订—可以进行商品评价，可以查看个人已有商品评价。

5. 后台管理系统—商品管理模块。该模块主要对商品，商品类别，供应商进行维护，可以上架新商品，下架已有商品，修改已有商品信息（商品销量不可修改），以及删除已有商品；可以添加，搜索商品类别，启用禁用商品类别；可以添加，修改，搜索供应商信息，启用禁用供应商。

6. 后台管理系统—销售管理模块。该模块主要包含用户管理，销售订单管理和销售评价管理，用户管理是对普通消费者用户的管理维护，可查看修改用户的基本信息，可启用/禁用账户；可查看用户收货地址信息；可以查看用户积分和积分扣减记录；可以进行VIP用户管理，将普通用户升级为VIP用户，或将用户降级为普通用户。销售订单管理主要是对所有订—进行管理，可以查看所有状态的订—，对于不同状态的订—，有不同的操作权限。销售评价管理是对商品评价进行管理，可查看所有用户的商品评价，搜索商品评价，跟进评价。

8. 后台管理系统—销售额管理模块。该模块主要是销售额进行管理，可以按照时间段，图书类别，图书名称，供应商等维度，进行销售额汇总计算。

9. 后台管理系统—销售报表模块。该模块主要包含图书销售排行榜和消费者分析。图书销售排行榜可根据时间维度，图书类别，供应商进行排行，消费者分析，可以根据消费者年龄段，下单时间，收货地址来分析图书销售情况。

10. 后台管理系统—网站管理模块。该模块主要是发布网站最新通知，管理通知，对所有的通知可以进行编辑，删除操作。

11. 后台管理系统—系统设置。该模块主要是对所有后台系统管理员进行维护，只有超级管理员有此模块权限，可以新增后台管理系统用户，修改已有人员的信息（包含角色），删除现有后台管理员等。

4.2 数据库设计

数据结构是一切系统设计的基础，如果数据结构本身设计存在问题，在之后的统开发过程中，可能要面临频繁返工问题，甚至到了后期的系统维护、功能扩展时，引发很多问题，例如数据记录不全，无法快速支持新需求等，严重时甚至要重新设计，重做大量已完成工作。故数据库设计是系统设计中非常重要的一个部分，设计出合理的数据库是一个十分关键的步骤。

4.2.1 数据逻辑设计

数据的逻辑设计首先要明确系统的实体，以及实体间的关系，然后根据实体关系进行数据库表的设计。为了将实体关系表现出来，选用E-R图来表示，E-R图非常方便，且具有较高的准确性，当下流行的概念模型的方法也是选用E-R图来表示。E-R图主要有三要素构成：实体，关系，属性，

对系统处理的数据进行分析，可以得出本系统共需要设计13个实体，分别是消费者用户，消费者用户收货地址，用户积分，积分扣减，管理员，通知，供应商，商品，商品类别，附件，商品评价，订—，订—详情，通过对各个实体之间的依赖关系进行符合逻辑的整合，得出系统实体E-R图如图4-2所示

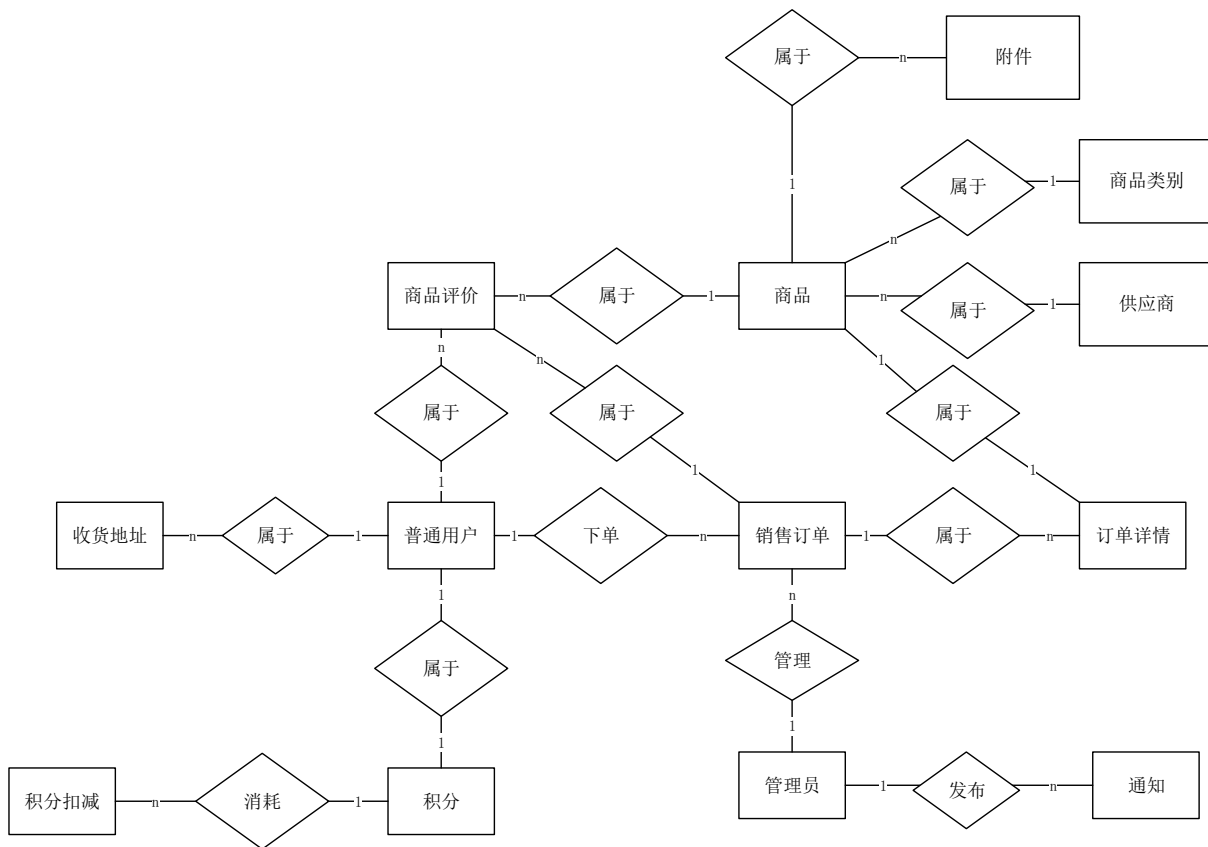


图 4-2 图书销售管理系统 E-R 图

系统主要的实体属性如下：

1. 消费者用户实体属性图。

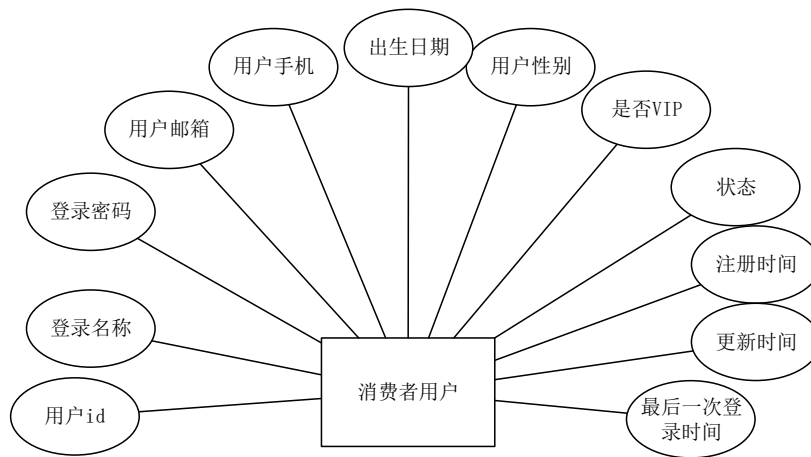


图 4-3 用户实体属性图

2. 消费者用户收货地址实体属性图

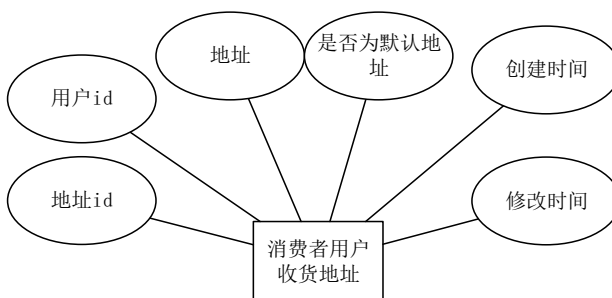


图 4-4 用户收货地址实体属性图

3. 消费者用户积分实体属性图

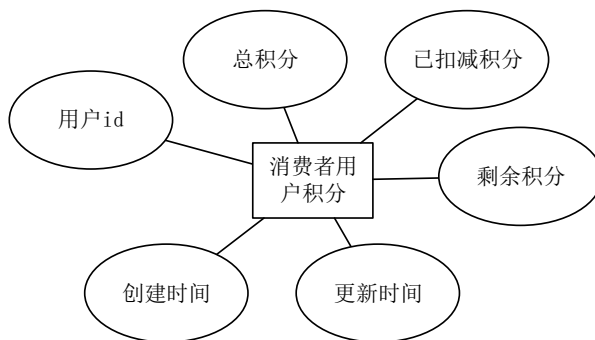


图 4-5 用户积分实体属性图

4. 消费者用户积分扣减实体属性图

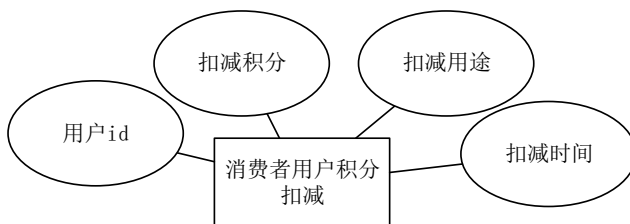


图 4-6 用户积分扣减实体属性图

5. 系统管理员实体属性图

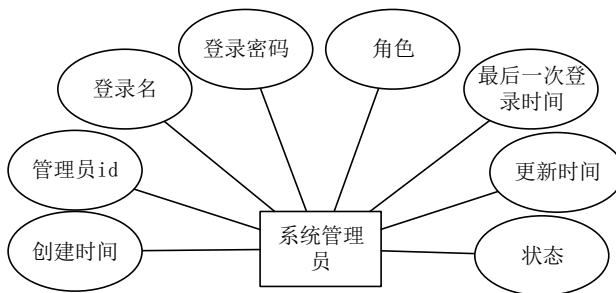


图 4-7 管理员实体属性图

6. 通知实体属性图

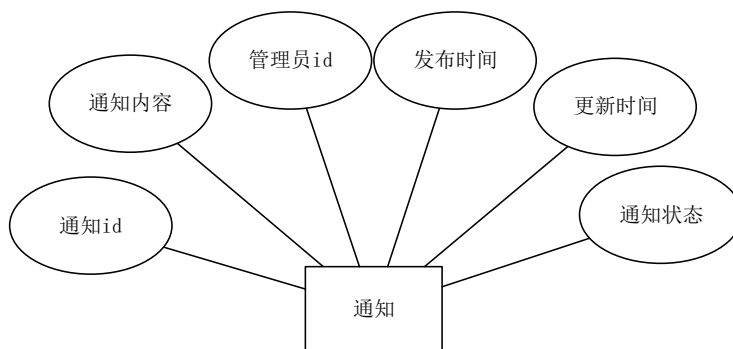


图 4-8 通知实体属性图

7. 供应商实体属性图

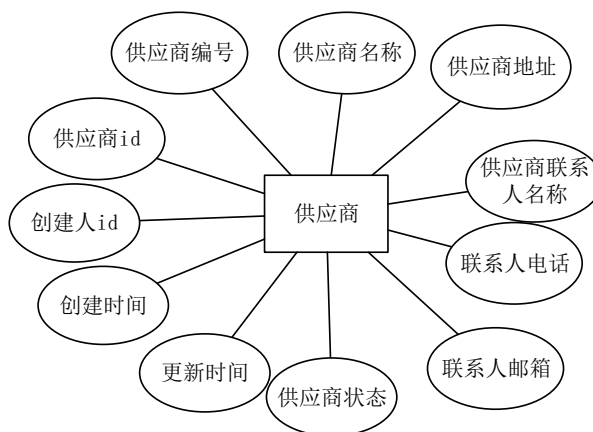


图 4-9 供应商实体属性图

8. 商品实体属性图

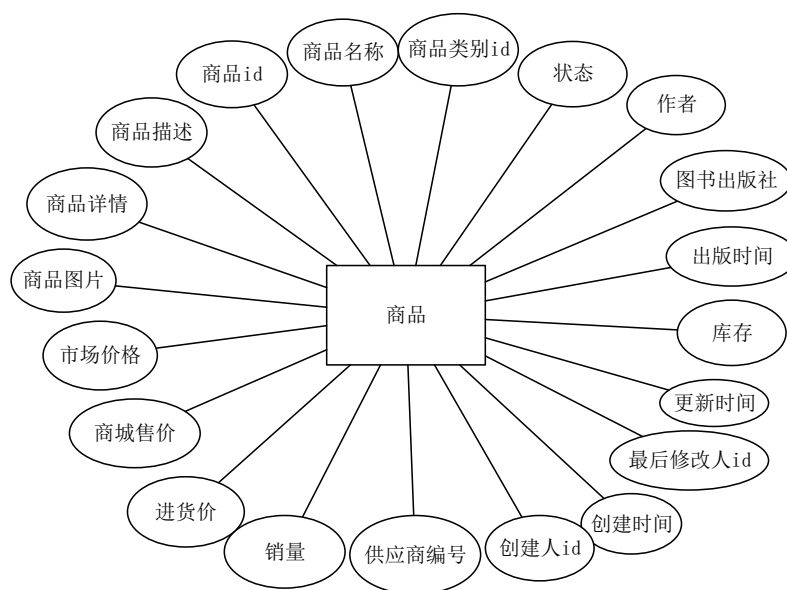


图 4-10 商品实体属性图

9. 商品类别实体属性图

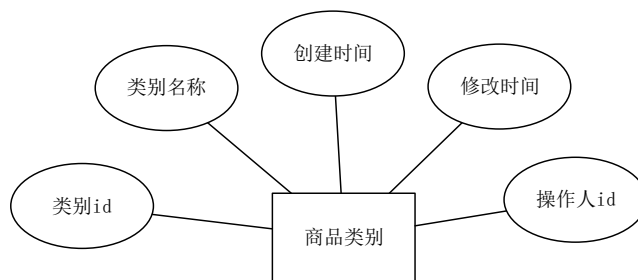


图 4-11 商品类别实体属性图

10. 商品评价实体属性图

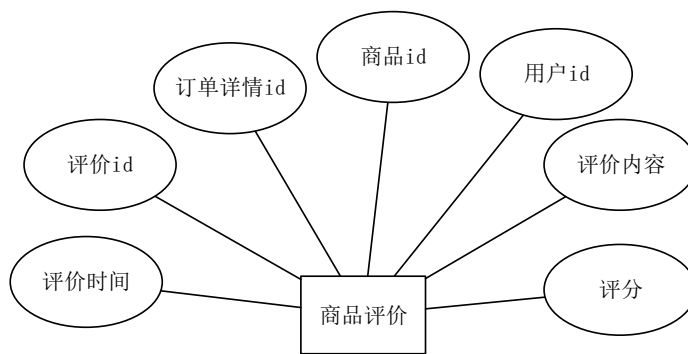


图 4-12 商品评价实体属性图

11. 订单实体属性图

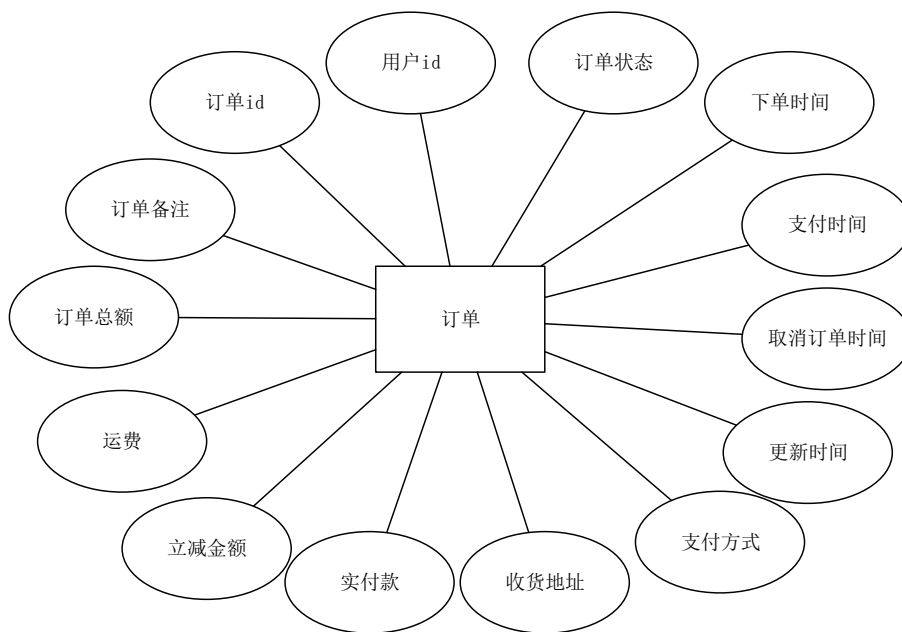


图 4-13 订单实体属性图

12. 订单详情实体属性图

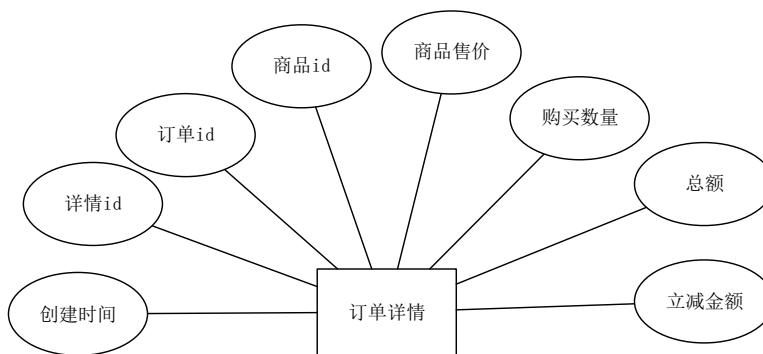


图 4-14 订单详情实体属性图

4.2.2 数据表设计

本系统的数据库中包含 13 张数据表。分别为：用户表，用户收货地址表，用户积分表，用户积分扣减记录表，管理员表，通知表，供应商表，商品表，商品类别表，商品评价表，订单表，订单详情表，附件表，下面是对各数据表的字段说明

用户表如表 4-1 所示

表 4-1 用户表 /t_user

列名	数据类型	是否为空	备注
----	------	------	----

userId	varchar(36)	NO	用户 id
userName	varchar(32)	NO	用户名称
password	varchar(8)	NO	用户密码
sex	varchar(4)	NO	性别
birthday	varchar(8)	YES	出生日期
email	varchar(32)	NO	用户邮箱
phoneNumber	varchar(20)	NO	用户电话
userStatus	tinyint	NO	用户状态
isVip	tinyint	NO	是否是 VIP
createTime	datetime	NO	创建时间
lastLoginTime	datetime	YES	最后一次登录时间
updateTime	datetime	YES	更新时间

用户收货地址表 如表 4-2 所示

表 4-2 用户收货地址表/t_user_delivery_address

列名	数据类型	是否为空	备注
addressId	varchar(36)	NO	地址 id
userId	varchar(36)	NO	用户 id
address	varchar(200)	NO	地址信息
defaultAddress	tinyint	YES	是否为默认收货地址
createTime	datetime	YES	创建时间
updateTime	datetime	YES	更新时间

用户积分表 如表 4-3 所示

表 4-3 用户积分表/ t_user_points

列名	数据类型	是否为空	备注
userId	varchar(36)	NO	用户 id
totalPoints	double	NO	总积分
deductPoints	double	YES	扣除积分
availablePoints	double	YES	剩余积分
createTime	datetime	YES	创建时间
updateTime	datetime	YES	更新时间

用户积分扣除表 如表 4-4 所示

表 4-4 用户积分扣除记录表/t_user_points_deduct_record

列名	数据类型	是否为空	备注
id	int(10)	NO	扣减 id
userId	varchar(36)	NO	用户 id

deductPoints	double	NO	扣除积分
source	tinyint	NO	扣除用途
isValid	tinyint	YES	是否有效
createTime	datetime	NO	扣除时间
updateTime	datetime	YES	更新时间

管理员表 如表 4-5 所示

表 4-5 管理员表/t_user_manager

列名	数据类型	是否为空	备注
managerId	varchar(36)	NO	管理员 id
managerName	varchar(32)	NO	管理员登录名
password	varchar(32)	NO	管理员密码
role	varchar(1)	NO	角色
lastLoginTime	datetime	YES	上次登陆日期
createTime	datetime	NO	创建日期
updateTime	datetime	YES	更新时间
status	tinyint	NO	状态

通知表 如表 4-6 所示

表 4-6 通知表/ t_notice

列名	数据类型	是否为空	备注
noticeId	int(10)	NO	通知 id
noticeContent	varchar(255)	NO	通知内容
operatorId	varchar(36)	NO	操作人员 id
noticeStatus	tinyint	NO	通知状态
createTime	datetime	NO	创建时间
updateTime	datetime	YES	更新时间

商品表 如表 4-7 所示

表 4-7 商品表/t_product

列名	数据类型	是否为空	备注
productId	varchar(36)	NO	商品 id
productName	varchar(200)	NO	商品名称
status	tinyint	NO	商品状态
author	varchar(100)	NO	作者
press	varchar(100)	NO	出版社
publicationTime	varchar(14)	NO	出版时间
categoryId	int(10)	NO	商品类别 id
description	varchar(500)	NO	商品描述
stock	int(10)	NO	商品库存
costPrice	decimal(14,2)	NO	进货价
marketPrice	decimal(14,2)	NO	市场价格
price	decimal(14,2)	NO	商城售价
salesvolume	int(10)	NO	销售量

iconUrl	varchar(100)	YES	商品图片
detailContent	longtext	NO	商品详情
createId	varchar(36)	YES	创建人 id
modifyId	varchar(36)	YES	最后一次更新人 id
createTime	datetime	NO	创建时间
updateTime	datetime	YES	更新时间

商品类别表 如表 4-8 所示

表 4-8 商品类别表/ t_product_category

列名	数据类型	是否为空	备注
categoryId	int(10)	NO	类别 id
categoryItem	varchar(32)	NO	类别名称
operatorId	varchar(36)	YES	操作人员 id
createTime	datetime	YES	创建时间
updateTime	datetime	YES	更新时间

商品评价表 如表 4-9 所示

表 4-9 商品评价表/ t_product_comment

列名	数据类型	是否为空	备注
commentId	int(10)	NO	评价 id
productId	varchar(36)	NO	商品 id
userId	varchar(36)	NO	用户 id
comment	varchar(1500)	YES	评价内容
score	tinyint	NO	评分
subscriptionId	varchar(50)	NO	订单 id
commentTime	datetime	NO	评价时间

附件表 如表 4-10 所示

表 4-10 附件表/ t_attachment

列名	数据类型	是否为空	备注
Id	int(10)	NO	附件 id
objectId	varchar(50)	YES	对象 id
fileType	tinyint	NO	附件类型
fileUrl	varchar(100)	NO	附件存储路 径
createTime	datetime	NO	创建时间
updateTime	datetime	YES	更新时间

订单表 如表 4-11 所示

表 4-11 订单表/ t_subscription

列名	数据类型	是否为空	备注
subscriptionId	varchar(36)	NO	订单 id
userId	varchar(36)	NO	用户 id
status	tinyint	NO	订单状态
subscriptionTime	datetime	NO	下单时间
unsubscribeTime	datetime	YES	取消订 单 时间

message	varchar(100)	YES	订单备注
payType	tinyint	NO	支付方式
payTime	datetime	NO	支付时间
address	varchar(100)	NO	收货地址
totalAmount	decimal(14,2)	NO	订单总额
freight	decimal(14,2)	NO	运费
totalDiscountAmount	decimal(14,2)	NO	优惠总金额
actualPayment	decimal(14,2)	NO	实付款
updateTime	datetime	YES	更新时间

订单详情表 如表 4-12 所示

表 4-12 订单详情表/ t_subscription_detail

列名	数据类型	是否为空	备注
detailId	varchar(36)	NO	订单详情 id
subscriptionId	varchar(36)	NO	订单 id
productId	varchar(36)	NO	商品 id
productPrice	decimal(14,2)	NO	商品单价
productCount	int(10)	NO	购买数量
productAmount	decimal(14,2)	NO	总金额
discountAmount	decimal(14,2)	YES	优惠金额
createTime	datetime	YES	创建时间

供应商表 如表 4-13 所示

表 4-13 供应商表 /t_supplier

列名	数据类型	是否为空	备注
supplierId	int(10)	NO	供应商 id
supplierCode	varchar(10)	NO	供应商编号
supplierName	varchar(32)	NO	供应商名称
supplieraddress	varchar(200)	NO	供应商地址
contractName	varchar(32)	NO	供应商联系人名称
contractPhone	varchar(20)	NO	供应商联系人电话
contractEemail	varchar(32)	YES	供应商联系人邮箱
supplierStatus	tinyint	NO	供应商状态
createId	varchar(36)	NO	创建人 id
createTime	datetime	NO	创建时间
updateTime	datetime	YES	更新时间

4.3 各模块流程详细设计

4.3.1 用户购买模块

用户购买指消费者用户从前端销售系统购买图书，包含查找选择商品，到购物车点击结算，生成订单，系统自动更新相对应的库存量和销售量，跳转支付，支付成功，购买完成。如果未登录则自动跳转到登录页面，让用户登录。用户购买流程如图 4-15 所示。

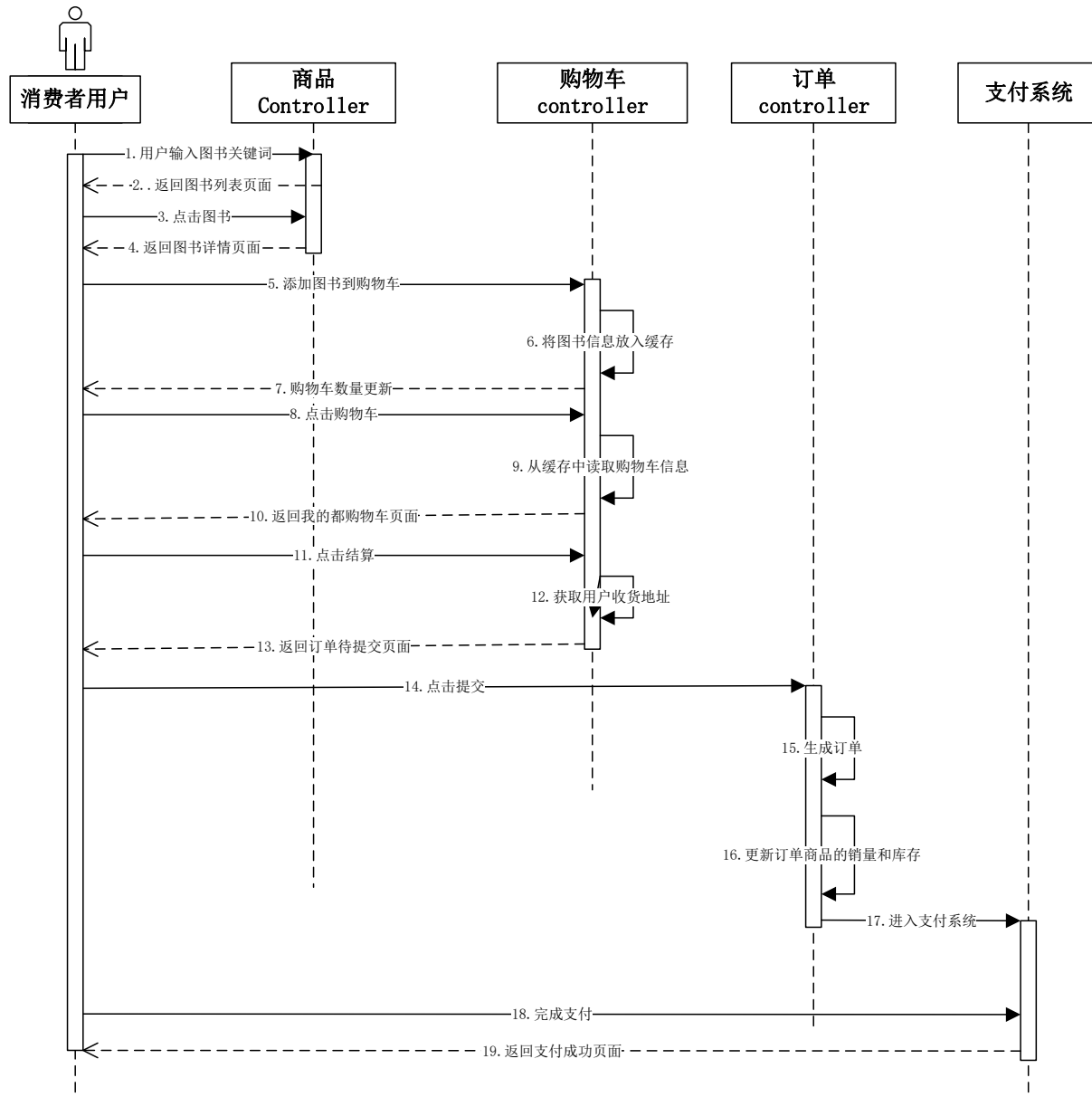


图 4-15 用户购买序列图

4.3.2 商品管理模块

商品管理模块在后台管理系统，主要由超级管理员和商品管理员操作，主要功能包含查看和搜索所有图书，要能支持上架或下架图书，包含添加、修改、搜索，启用、禁用供应商，商品上架流程如图 4-16 所示

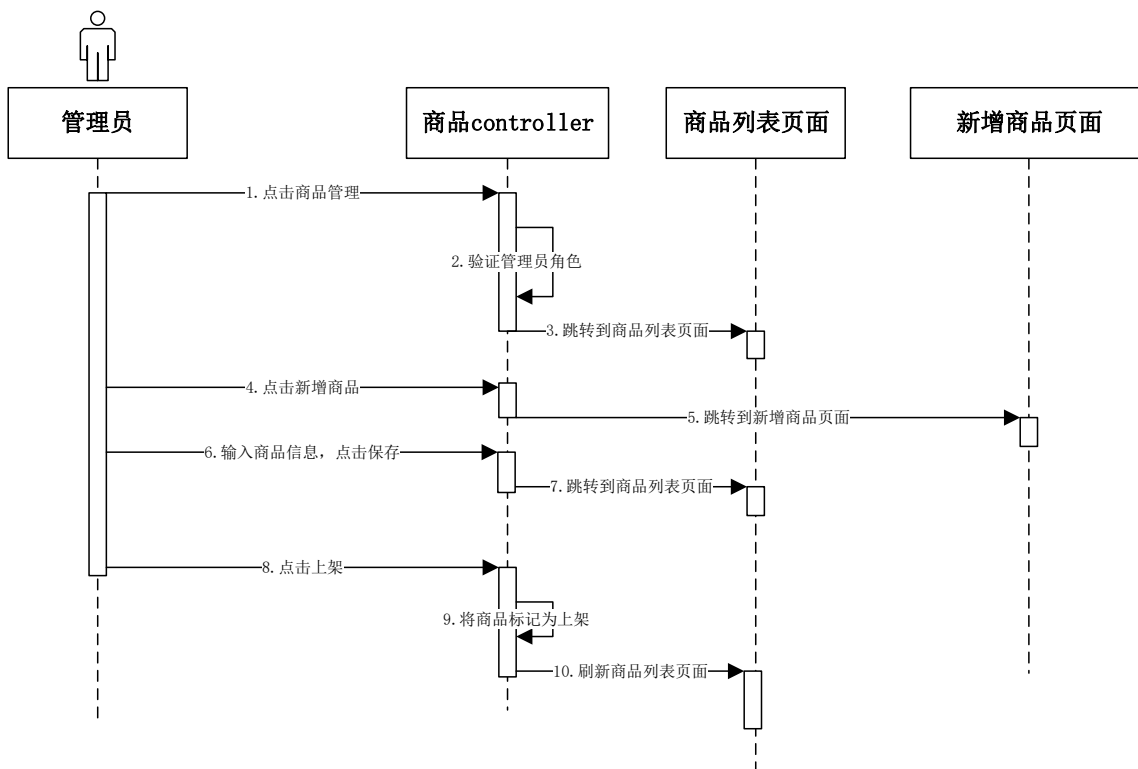


图 4-16 商品上架序列图

4.3.3 销售管理模块

销售管理模块在后台管理系统，主要由超级管理员和销售操作，主要功能包括查看，搜索网站所有订单，订单发货后在系统中标记已发货，未完成的订单可以取消，查看销售商品评价，查看，搜索消费者用户信息。其中订单发货流程如图 4-17 所示

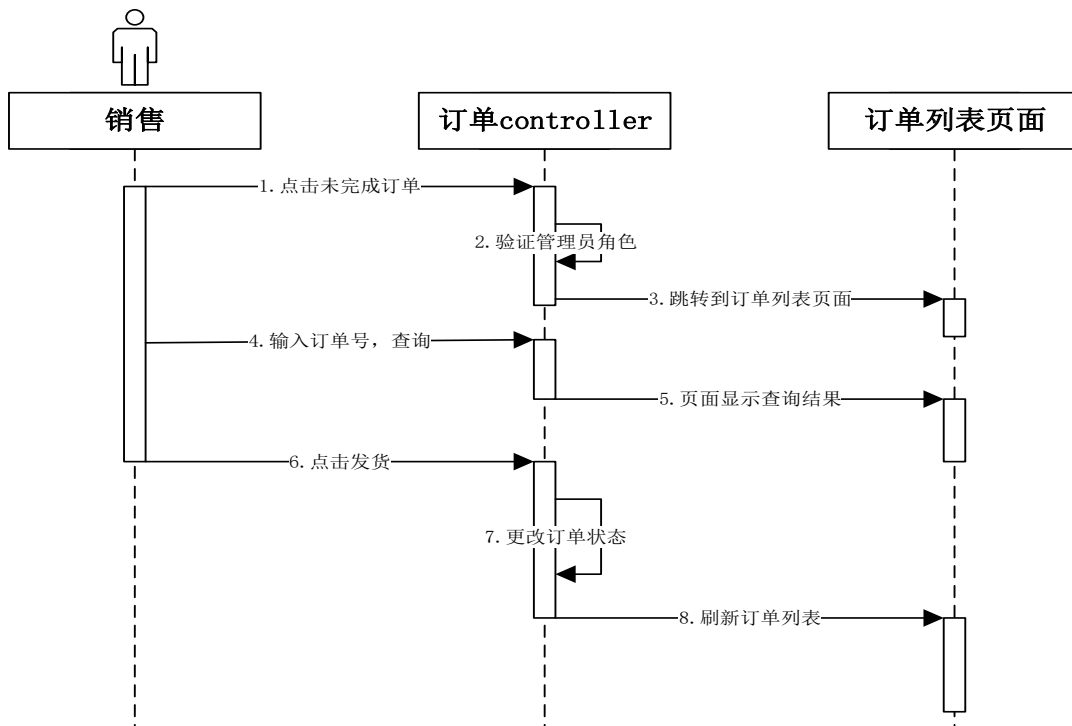


图 4-17 销售订单发货序列图

4.3.4 销售额管理模块

销售额管理模块在后台管理系统，主要由销售经理和超级管理员操作，主要功能是图书销售额汇总。默认显示当天全品类的销售额，同时可以根据

时间段（日，周，月，季度，年，或特定时间段），图书类别，供应商进行销售额查询汇总；其中图书销售额管理流程如图 4-18 所示

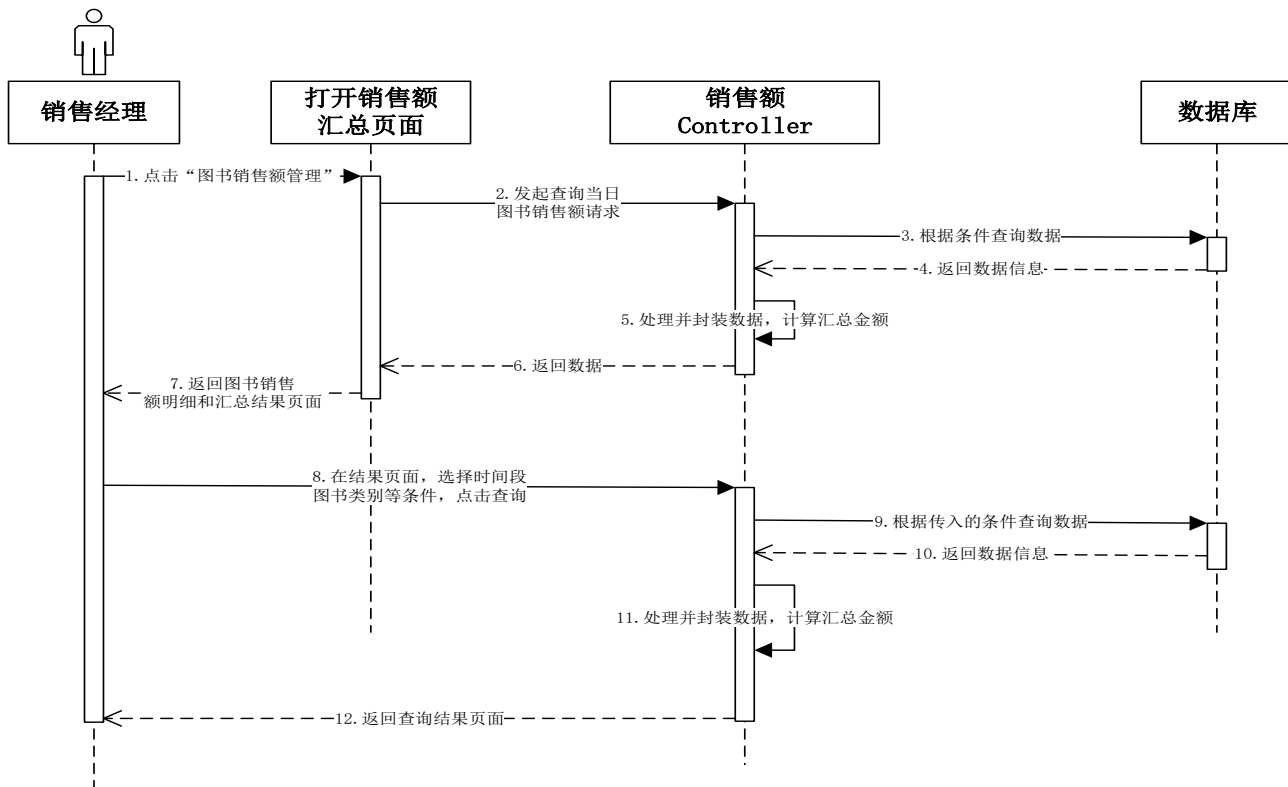


图 4-18 图书销售额管理榜序列图

4.3.5 销售报表模块

销售报表模块在后台管理系统，主要由销售经理和超级管理员操作，主要功能是图书销售排行榜，图书评分排行榜，消费者分析。销售排行榜默认显示当天全品类的排行，同时可以根据时间段（日，周，月，季度，年，或特定时间段），图书类别，供应商进行查询排行；图书评分排行榜默认显示近一月的全品类评分排行，同时可以根据时间段，图书类别进行查询排行；消费者分析按照十年为一个年龄段查询所有品类的销售数量，同时可以根据指定年龄段，指定品类，指定图书查询销售数量，其中图书销售排行榜流程如图 4-19 所示

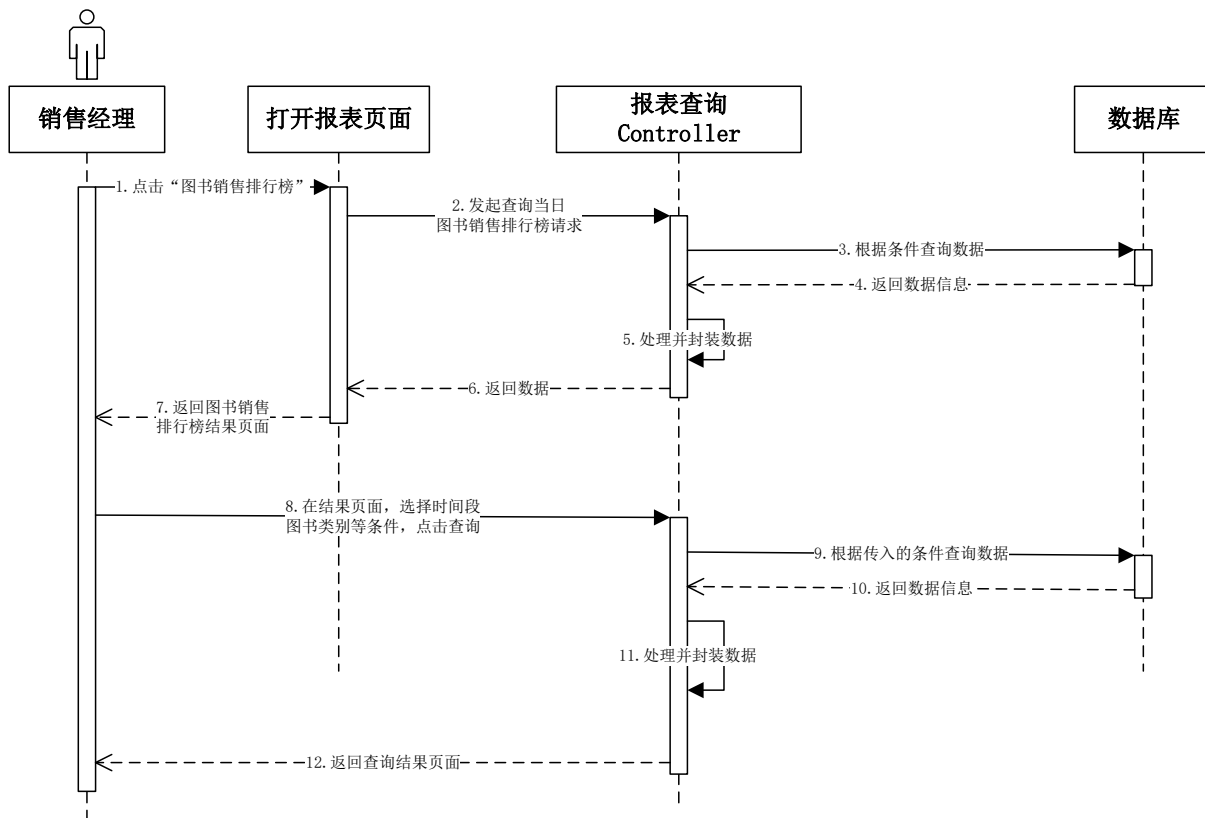


图 4-19 图书销售排行榜序列图

5. 系统实现

5.1 系统开发环境的搭建

环境搭建有两部分：硬件环境和软件环境。

硬件环境采用普通网络设备环境。

软件环境有以下几部分组成：

服务器：windows10，

JDK 版本：1.8

数据库：MySQL

集成开发工具：IDEA

Web 应用服务器：TOMCAT8.0

5.2 关键模块的实现

5.2.1 搜索商品模块的实现

网络购物时网站信息数据量过大，用户不可能一条一条去找想要购买的书籍，必须能够让用户根据自己的需求搜索想要的书籍。本系统支持两种搜索方式，一种是根据商品类别搜索出该类别商品，在进行查找，第二种是直接输入商品名称关键字进行模糊查询，检索之后系统会自动把按照要求搜索到的全部图书信息展现给消费者用户，搜索商品界面如图 5-1 所示。



图 5-1 搜索图书界面

关键实现如下：

```
@RequestMapping("searchProduct")
```

```
public ModelAndView toProductManage(HttpServletRequest request, ModelAndView
model) {
    // 获取用户名
    String username = request.getParameter("username");
    CommonService.setUserSession(request, model, username);
    // 商品名称关键字
    String condition = request.getParameter("condition");
    // 商品类别 id
    String categoryId = request.getParameter("categoryId");
    ProductFilter productFilter = new ProductFilter();
    productFilter.setProductName(condition);
    productFilter.setCategoryId(categoryId);
    // 根据条件查询已上线的产品
    productFilter.setStatus(ConstantCode.ProductStatus.ONLINE);
    List<ProductDetail> productList =
productManager.getProductList(productFilter);
    model.setViewName("productList");
    model.addObject("productList", productList);
    return model;
}
```

5.2.3 个人订单中心模块的实现

消费者用户选择好商品进行结算提交订单后，会生成对应的订单编号，在我的订单中，用户可以查看到自己名下所有的订单，可以对已经发货的订单进行确认收货的操作。我的订单界面如图 5-2 所示

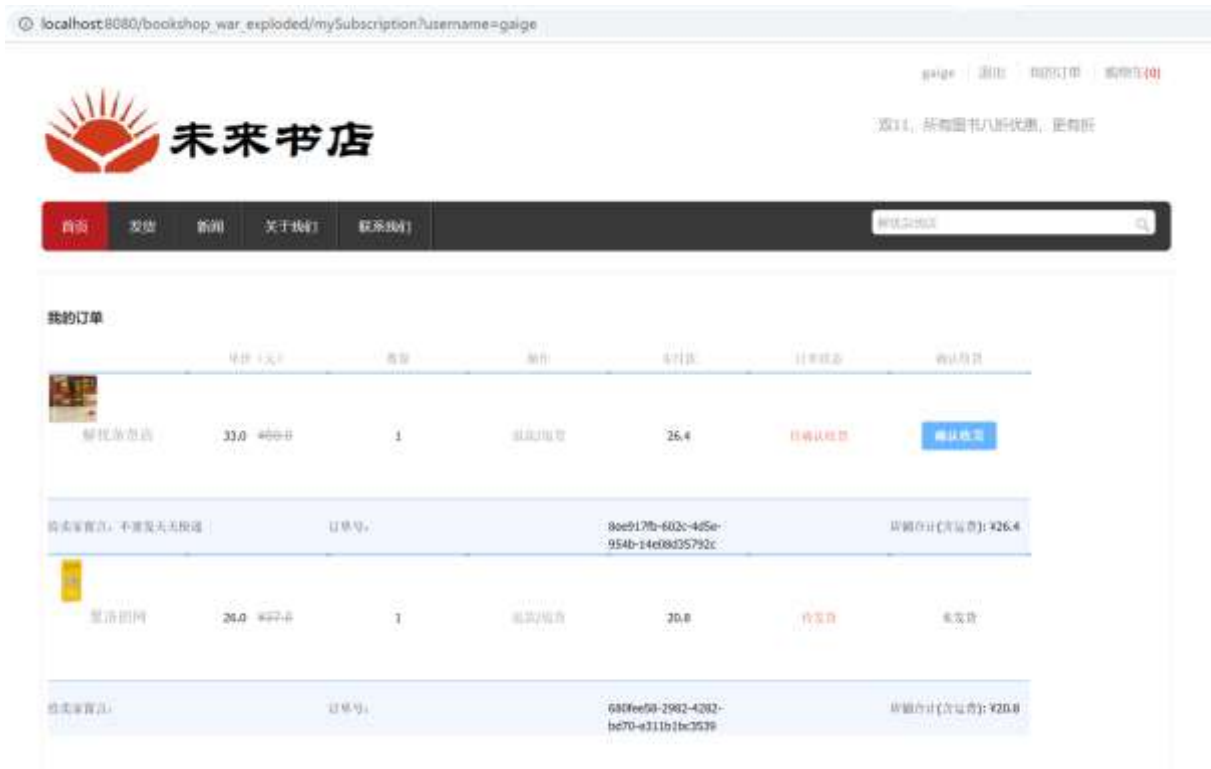


图 5-2 我的订单界面

我的订单具体实现如下：

```
@RequestMapping("mySubscription")
```

```
public ModelAndView mySubscription(HttpServletRequest request, ModelAndView model)
```

```
{
```

```
    // 获取用户名
```

```
    String username = request.getParameter("username");
```

```
    CommonService.setUserSession(request, model, username);
```

```
    // 查询用户信息
```

```
    UserDetail userInfo = userService.getUserDetail(username);
```

```
    // 查询用户订单信息
```

```
    List<SubscriptionDetail> subList = subscriptionService.getSubByUserId(userInfo.getUserId());
```

```
    if(!CollectionUtils.isEmpty(subList)) {
```

```
        List<String> prodIds = new ArrayList<>();//拿到所有产品 id
```

```
        for (SubscriptionDetail sub: subList) {
```

```
            prodIds.add(sub.getProductId());
```

```
        }
```

```
        // 查询产品信息
```

```
        Map<String, ProductRecord> productRecordMap = productService.getProductRecordByIds(prodIds);
```

```
        for (SubscriptionDetail subDetail : subList) {
```

```
            ProductRecord productRecord = productRecordMap.get(subDetail.getProductId());
```

```
            subDetail.setIconUrl(productRecord.getIconUrl());
```

```
            subDetail.setPrice(productRecord.getPrice());
```

```
            subDetail.setMarketPrice(productRecord.getMarketPrice());
```

```

        subDetail.setProductName(productRecord.getProductName());
    }
}
model.setViewName("mySubscription");
model.addObject("subList", subList);
return model;
}

```

5.2.3 商品管理模块的实现

商品管理模块是商品管理员和超级管理员才有的权限，登录后端的管理系统后，管理员可以管理商品，可根据关键字搜索商品，也可添加新商品和下架商品；也可以管理商品类别，添加新类别，删除已有类别。商品管理界面如图 5-3 所示。



图 5-3 商品管理界面

具体实现如下：

```

@RequestMapping("backstage_productManage")
public ModelAndView toProductManage(HttpServletRequest request, ModelAndView model)
{
    // 权限验证
    String username = (String) request.getSession().getAttribute("adminName");
    if (StringUtils.isBlank(username)) {
        model.addObject("message", "未登录或者登录超时");
        model.addObject("result", -1);
        model.setViewName("/backstage/operateResult");
        return model;
    }
    ManagerRecord record = managerService.getMangerRecordByName(username);
    String role = record.getRole();
    // 不是超级管理员或者商品管理员
    if (!role.equals("0") && !role.equals("2")) {
        model.addObject("message", "权限不足");
        model.setViewName("backstage/operateResult");
        model.addObject("result", -1);
        return model;
    }
    // 查询产品
    String productName = request.getParameter("productName");
    ProductFilter productFilter = new ProductFilter();

```

```

        productFilter.setProductName(productName);
        List<ProductDetail> productList =
productManager.getProductList(productFilter);
        model.setViewName("backstage/Product/productIndex");
        model.addObject("productList", productList);
        return model;
    }

```

5.2.4 销售管理模块的实现

销售管理模块是销售和超级管理员才有的权限，登录后端的管理系统后，管理员可以管理所有订单，订单评价和消费者用户，其中订单管理可根据订单编号搜索订单，订单后可将订单比较已发货，未完成的订单可以取消。订单管理界面如图 5-4 所示。



图 5-4 订单管理界面

具体实现如下：

```

@RequestMapping("backstage_toNoSubscription")
public ModelAndView toNoSubscription(HttpServletRequest request, ModelAndView
model) {
    // 登陆检测
    String adminName = (String) request.getSession().getAttribute("adminName");
    if (StringUtils.isBlank(adminName)) {
        model.addObject("message", "您未登录或者登录超时");
        model.addObject("result", -1);
        model.setViewName("/backstage/operateResult");
        return model;
    }
    // 权限控制
    ManagerRecord recorda = managerService.getMangerRecordByName(adminName);
    String role = recorda.getRole();
    // 不是超级管理员或者销售
    if (!role.equals("0") && !role.equals("5")) {
        model.addObject("message", "权限不足");
        model.setViewName("backstage/operateResult");
        model.addObject("result", -1);
        return model;
    }
    // 查询未完成的订单
    List<SubscriptionResult> subscriptionResultList =
subscriptionService.querySbuscriptionNoCompl();
    model.addObject("subscriptionResultList", subscriptionResultList);
    model.setViewName("/backstage/Subscription/index");
}

```

```

return model;
}

```

5.2.5 销售额管理模块的实现

销售额管理模块是销售经理和超级管理员才有的权限，登录后端的管理系统后，管理员可以查看根据时间段，商品类别，商品名称，供应商计算销售额。销售额管理模块界面如图 5-5 所示。



图 5-5 销售额管理界面

具体实现如下：

```

@RequestMapping("backstage_toSales")
public ModelAndView toSales(HttpServletRequest request, ModelAndView model) {
    // 登陆检测
    String adminName = (String) request.getSession().getAttribute("adminName");
    if (adminName == null) {
        model.addObject("message", "您未登录或者登录超时");
        model.addObject("result", -1);
        model.setViewName("/backstage/operateResult");
        return model;
    }
    // 权限控制
    ManagerRecord recorda = new ManagerRecord();
    recorda.setManagerName(adminName);
    recorda = managerService.getManagerRecordByName(recorda);
    String role = recorda.getRole();
    // 最高管理员或者销售经理
    if (!role.equals("0") && !role.equals("6")) {
        model.addObject("message", "权限不足");
        model.setViewName("backstage/operateResult");
        model.addObject("result", -1);
        return model;
    }
    String startTime = request.getParameter("startTime");//开始时间
    String endTime = request.getParameter("endTime");//结束时间
    String supplierId = request.getParameter("productCondition");//供应商 id
    String productCondition = request.getParameter("productCondition");// 商品
    名称关键字
    String categoryId = request.getParameter("categoryId");// 商品类别 id
    // 根据条件查询订单销售额信息
    SalesFilter filter = new SalesFilter();

```

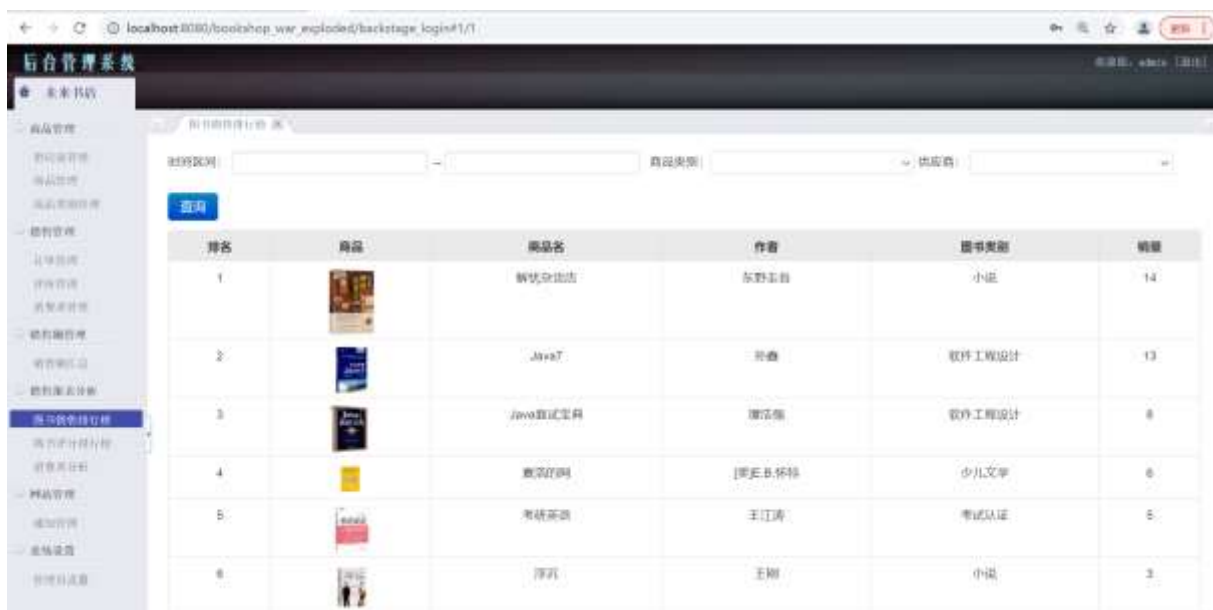
```

        filter.setStartTime(startTime);
        filter.setEndTime(endTime);
        filter.setSupplierId(supplierId);
        filter.setProductCondition(productCondition);
        filter.setCategoryId(categoryId);
        List<SubscriptionResult> subscriptionResultList =
subscriptionService.querySbuscriptionSales(filter);
        model.addObject("subscriptionResultList", subscriptionResultList);
        model.setViewName("/backstage/Sales/index");
        return model;
    }

```

5.2.6 销售报表分析模块的实现

销售报表分析模块是销售经理和超级管理员才有的权限，登录后端的管理系统后，管理员可以查看图书销售排行榜，图书评分排行榜，消费者分析。其中销售额管理模块界面如图 5-6 所示。



排名	商品	商品名	作者	图书类别	销量
1		新编Java	东野圭吾	小说	14
2		Java7	张鑫	软件工程设计	13
3		Java面试宝典	唐宏海	软件工程设计	8
4		童真乐园	[美] B. 怀特	少儿文学	8
5		考研英语	王江涛	考试认证	5
6		浮城	王朔	小说	3

图 5-6 销售报表分析界面

具体实现如下：

```

@RequestMapping("backstage_toCountRanking")
public ModelAndView toSales(HttpServletRequest request, ModelAndView model) {
    // 登陆检测
    String adminName = (String) request.getSession().getAttribute("adminName");
    if (adminName == null) {
        model.addObject("message", "您未登录或者登录超时");
        model.addObject("result", -1);
        model.setViewName("/backstage/operateResult");
        return model;
    }
    // 权限控制
    ManagerRecord recorda = new ManagerRecord();
    recorda.setManagerName(adminName);
    recorda = managerService.getMangerRecordByName(recorda);
}

```

```

String role = recorda.getRole();
// 最高管理员或者销售经理
if (!role.equals("0") && !role.equals("6")) {
    model.addObject("message", "权限不足");
    model.setViewName("backstage/operateResult");
    model.addObject("result", -1);
    return model;
}

String startTime = request.getParameter("startTime");//开始时间
String endTime = request.getParameter("endTime");//结束时间
String supplierId = request.getParameter("productCondition");//供应商 id
String categoryId = request.getParameter("categoryId");// 商品类别 id
// 根据条件查询产品销量
SubscriptionFilter filter = new SubscriptionFilter ();
filter.setStartTime(startTime);
filter.setEndTime(endTime);
filter.setSupplierId(supplierId);
filter.setCategoryId(categoryId);
List<ProductDetailResult> productDetailList =
productService.queryCountRanking (filter);
model.addObject("productDetailList ", productDetailList);
model.setViewName("/backstage/Report/index");
return model;
}
    
```

5.3 出现的问题及解决的办法

1) 在分析和设计阶段，因为把握不清销售系统和进销存系统的边界，以至于在分析和设计过程中总会出现方向上的偏离，在网上找了各种销售系统的论文，反复进行研究和对比，相对明确了边界。

2) 前端的开发过中，因为录入商品时，需要录入商品详情，为了方便录入要用到富文本编辑器，之前对于这块没有特别的接触，算是开发过程中的一个难点，所以在网上学习了 UEditor。

6. 系统测试

系统测试是根据用户的实际要求，检测所设计系统的各方面的性能情况。测试所要完成的目标有：检验系统是否与系统需求、用例需求相符合；准确、详细的记录系统测试情况，记录其优缺点，尤其是系统缺陷；确保软件系统测试活动及其结果能够及时通知相关的小组和个人。

6.1 测试环境搭建

在进行系统测试前，首先要对测试环境进行搭建。本系统进行测试的环境如表 6-1 所示。

表 6-1 测试环境

项目	内容
测试平台	window10
硬件设备	CPU : Intel(R) Core(TM) i5-4590 ;硬盘： 250G ; 内存： 8G
WEB 服务器	Tomcat8.0
JDK	1.8

6.2 部分功能测试

测试模块：用户选择商品添加至购物车，在购物车结算功能测试

测试用例如 表 6-2 所示

表 6-2 测试用例

用例 ID	测试步骤	预期结果	测试结果
1	已经登录的用户，在购物车点击结算	页面跳转到订单提交页面	与预期结果一致
2	未登录用户，在购物车点击结算	页面跳转到登录页面	与预期结果一致
3	在订单提交页面，未选择收货地址，点击提交	页面阻断，提示“还没添加收货地址”	与预期结果一致
4	点击“添加收货地址”	页面跳转到收货地址管理	与预期结果一致
5	添加收货地址成功，返回购物车，点击结算	页面跳转到订单提交页面，添加的默认地址默认选中	与预期结果一致
6	点击提交	页面成功跳转到支付页	与预期结果一致

测试模块：书店图书管理员添加新商品功能测试

测试用例如表 6-3 所示

表 6-3 测试用例 2

用例 ID	测试步骤	预期结果	测试结果
1	登录超时的管理员，点击“新增商品”	页面跳转到登录超时页面	与预期结果一致
2	在新增商品页面，未选择分类，点击保存	页面阻断，提示类别不能为空	与预期结果一致
3	在新增商品页面，编辑商品详情，添加图片，编辑文本	导入图片可正常显示，文本内容可正确编辑	与预期结果一致
4	点击“返回列表”	页面内容不会保存，页面跳转到商品列表页面	与预期结果一致
5	点击“保存”	页面跳转到商品列表页面，刚刚新增也正确显示在列表中	与预期结果一致

测试模块：书店后台管理系统销售管理测试

测试用例如表 6-4 所示

表 6-4 测试用例 3

用例 ID	测试步骤	预期结果	测试结果
1	点击“销售额汇总”	页面跳转到销售额汇总页面，显示当天所有图书销售订单销售明细和合计金额	与预期结果一致
2	在汇总页面，选择商品类别为“少儿文学”，点击查询	页面展示当天所有类别为“少儿文学”的销售订单明细和合计金额	与预期结果一致
3	在汇总页面，选择供应商为“天顶文化传播有限公司”，点击查询	页面展示当天类别为“少儿文学”，供应商为“天顶文化传播有限公司”的销售订单明细和合计金额	与预期结果一致
4	在汇总页面，选择时间段为“2021-10-1”到“2021-10-31”，点击查询	页面展示下单时间在“2021-10-1”到“2021-10-31”，类别为“少儿文	与预期结果一致

		学”，供应商为“天顶文化传播有限公司”的销售订单明细和合计金额	
5	在汇总页面，商品名称填写“java”，点击查询	页面展示下单时间在‘2021-10-1’到 2021-10-31，类别为“少儿文学”，供应商为“天顶文化传播有限公司”，商品名称中包含“java”的销售订单明细和合计金额	与预期结果一致

6.3 测试结果

测试中发现了一些程序的 BUG，通过代码优化，最终得以解决。系统响应时间短，可以带给消费者用户良好的购物体验，也可以给书店管理员带来便捷的管理。

7. 总结

本文主要对中小型综合性图书销售信息管理系统进行了较为深刻的研究和探讨，通过对本片论文的书写，让我对中小型实体书店的现状有了更深入的了解，开发的整个过程中，也对学过的知识进行了一次重新的梳理。前阶段的开发任务如基本的购物流程，商品管理，销售管理，销售额管理，销售报表分析，网站管理等已经完成，因当前另一种主流的推广方式就是微信，微信小程序也火爆来袭，本系统采用 SSM 架构，该架构可以很方便的对接微信小程序，故给书店开发微信小程序也准备好了基础；但由于时间和精力有限，系统仍然存在很多不足，一些问题有待解决，例如支付系统模块的开发，报表分析的图形化，技术框架上没有采用 vue 做成前后端分离等；在以后的工作的学习过程中将对存在的问题进行深入研究，不断的完善。

【引文注释】

注[1]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.

注[2]谢瀚力.基于 JSP 的网上书店系统的设计与实现初探[J].电脑迷,2018(02):81.

【参考文献】

[1] 蒋舒娟.后疫情时代的图书销售启示[J].市场观察,2020,(7).

[2] 李天庆.基于 SSM 框架的电子商城项目的设计与实现.山东大学.2019

[3] 朱沈宁.图书销售系统的设计与实现[D].东北大学,2016.

[4] 田田.稳中向好:基于京东《2019 图书阅读报告》的观察与分析[J].出版广角,2020, No. 365(11):39-41.

[5] 赵浩翔.基于 SSM 框架的网上商城系统的设计与实现[D].北京邮电大学,2019.

[6] 陈婷.“书店+”模式在实体书店设计中的应用与思考[D].山西大学,2019.

[7] 郭永虎.中国实体书店与国民阅读的发展蜕变[J].人民论坛,2019(34).

[8] 朱永新.全民阅读奠基未来[J].教育(周刊),2020,000(003):P.46-48.

[9] 黄建.大数据视角下线上图书营销因素的实证研究——以京东平台的图书评论为例[J].现代出版,2019, No. 120(02):64-67.

[10] 刘燕军.试论新时期编辑在图书线上销售中的价值体现[J].东西南北,2020, No. 556(08):184-185.