
可行性分析(研究)报告(FAR)

目录

可行性分析(研究)报告(FAR).....	1
1 引言	3
1.1 标识	3
1.2 背景	3
1.3 项目概述	3
1.4 文档概述	4
2 引用文件	4
3 可行性分析的前提	4
3.1 项目的要求	4
3.2 项目的目标	5
3.3 项目的环境、条件、假定和限制	5
3.4 进行可行性分析的方法	5
4 可选的方案	6
4.1 可选择的方案	6
4.1.1 可选择的系统方案 1	6
4.1.2 可选择的系统方案 2	6
4.1.3 可选择的系统方案 3	6
4.2 原有方案的优缺点、局限性及存在的问题	7
4.2.1 方案 1	7
4.2.2 方案 2	7
4.2.3 方案 3	7
4.2.4 总体对比	7
4.3 可重用的系统，与要求之间的差距	8
4.3.1 Spring Boot 配置	8
4.3.2 Vue.js 前端	8
4.3.3 Spring Security 和 JWT 配置	8
4.4 选择最终方案的准则	8
5 所建议的系统	9
5.1 对所建议的系统的说明	9
5.2 数据流程和处理流程	9
5.3 与原系统的比较(若有原系统)	9
5.4 影响(或要求)	9
5.4.1 设备	9
5.4.2 软件	9
5.4.3 运行	9
5.4.4 开发	10
5.4.5 环境	10
5.4.6 经费	10

5.5 局限性	10
6 经济可行性(成本---效益分析)	10
6.1 投资	10
6.1.1 基本建设投资	10
6.1.2 其他一次性投资	11
6.1.3 非一次性投资	11
6.1.4 运营成本（长期投资）	11
6.2 预期的经济效益	11
6.2.1 一次性收益	11
6.2.2 非一次性收益	12
6.2.3 不可定量的收益	12
6.2.4 收益/投资比	12
6.2.5 投资回收周期	12
6.3 市场预测	12
7 技术可行性(技术风险评价)	13
8 法律可行性	13
9 用户使用可行性	14
10 其他与项目有关的问题	15
11 注解	16
附录	16

1 引言

1.1 标识

标识号：网上书店项目可行化分析报告
缩略词：OSBP（Online Bookstore Project）
版本号：1.0
发行号：2024-03-10

1.2 背景

当消费者对在线购物的需求增加，并且越来越多的人选择通过互联网购买商品时，提出网上书店项目可能是为了满足这种市场趋势和需求。随着互联网和电子商务技术的不断发展，建立一个网上书店变得更加容易和实用。例如，安全支付系统、电子书阅读器的普及等技术进步可以为网上书店的实施提供支持。与此同时，传统书店面临着来自线上竞争对手的竞争压力。为了保持竞争力，传统书店可能会考虑建立自己的网上渠道，或者新兴企业可能会看到这个市场的机会并提出网上书店项目。网上书店可以跨越地域和国界，向全球范围内的消费者提供服务。这种全球化的趋势以及人们对更广泛选择的需求可能会促使提出网上书店项目。

要求：用户的注册，登录；用户的分级浏览或图书选购；店主对进书，售书，库存，账目，客户的管理；以及网站的日常维护(比如,网上书店简介;网上书店信息发布;客户留言及对客户留言的反馈)。一般客户可以浏览网上书店内容,欲购书的客户需注册取得唯一的用户名成为会员，会员登录后便可以购书一本或多本。

目标：扩大销售渠道，增加销售量和利润；提升品牌知名度和市场份额；满足消费者对便捷购书方式的需求，提高用户满意度。

实现环境：

限制条件：资金方面，建立和维护一个网上书店需要一定的资金投入，包括网站开发、库存管理、营销推广等方面的费用。技术方面，可能会面临技术挑战，如网站性能优化、安全性保障等方面的问题。法律法规方面，需要遵守相关的电子商务法律法规，如消费者权益保护、个人信息保护等。

1.3 项目概述

本文档适用于网上书店项目（Online Bookstore Project），旨在提出并描述一个在线书店的建设计划。该项目的主要目标是创建一个安全、易用、可靠的在线购书平台，以满足消费者对便捷购书方式的需求，并且扩大销售渠道，增加销售量和利润。

该项目涉及的应用将包括：

电子商务平台：用于展示书籍、管理库存、处理订单和支付等功能的网站或应用程序。

安全支付系统：用于处理各种支付方式，包括信用卡、支付宝、微信支付等。

客户服务系统：用于提供客户支持和售后服务的系统，包括在线聊天、电子邮件支持等。

书籍管理系统： 用于管理书籍信息、库存、价格等的后台管理系统。

数据分析工具： 用于分析销售数据、用户行为等，以指导营销策略和库存管理。

这些软件的一般特性包括：

安全性： 提供安全的支付系统和用户数据保护，确保用户信息的机密性和交易的安全性。

易用性： 提供简洁清晰的用户界面，使用户能够轻松浏览、搜索和购买书籍。

可靠性： 保证系统的稳定性和可用性，避免系统故障和服务中断对用户体验的影响。

灵活性： 能够支持多种支付方式、多种书籍格式（如纸质书、电子书等）以及不同地区的配送方式。

扩展性： 能够支持未来业务增长和功能扩展，如新增书籍类别、优惠活动等。

总之，本文档描述了一个网上书店项目的建设计划，涉及到多个软件系统的开发和集成，旨在提供一个安全、便捷、可靠的在线购书平台。

本项目由软件工程实践小组开发。

1.4 文档概述

本文档的用途是提出和描述一个网上书店项目的建设计划，包括项目的背景、目标、实施计划以及涉及的软件系统和功能需求等方面。内容涵盖了项目提出的条件、要求、目标、实现环境和限制条件，以及软件系统的一般特性和功能描述。

文档中涉及用户数据、客户信息等个人隐私内容，需要严格遵守相关法律法规，确保用户数据的安全和隐私保护。

文档中包含了软件系统的设计方案、技术方案等知识产权内容，需要保护这些知识产权，防止未经授权的使用或复制。

2 引用文件

【本章应列出本文档引用的所有文档的编号、标题、修订版本和日期，本章也应标识不能通过正常的供货渠道获得的所有文档的来源。】

3 可行性分析的前提

3.1 项目的要求

创建一个网上书店项目，基本要求如下：

- （1）用户的注册，登录；
- （2）用户的分级浏览或图书选购；
- （3）店主对进书，售书，库存，账目，客户的管理；以及网站的日常维护(比如,网上书店简介;网上书店信息发布;客户留言及对客户留言的反馈)。
- （4）一般客户可以浏览网上书店内容,欲购书的客户需注册取得唯一的用户名成为会员，会员登录后便可以购书一本或多本。

也可以根据实际情况拓展网上书店的功能。

3.2 项目的目标

在实际应用中店主应该能够对网上书店的书籍的添加与下架、书籍信息，售书订单、库存、客户信息进行查询和管理。客户注册一个用户名之后，便可以浏览系统书店内的书籍信息，选定之后购书一本或多本。店主在客户确认网上订书单后，及时处理订单，修改物流信息等。同时，系统要求能够提供两级图书目录和三层信息（一级目录是图书的基本分类目录；其下是图书的二级目录，它对应于某基本分类之下的书名及其简要信息；当客户点击二级目录中的某本书之后，系统应当显示详细介绍该书的文字与图形信息），店主则是可以对进书、售书、库存、账目以及客户进行管理

3.3 项目的环境、条件、假定和限制

运行环境及条件：

服务器端：

操作系统：WINDOWS，LINUX/UNIX

客户端：

操作系统：任何 pc 平台；

浏览器：Internet Explorer

假定：本项目于本学期结束前完成。

限制：网上书店系统面向全球互联网用户，进行相关交易前必须注册为本网络平台用户，才能拥有相应权限；应用本平台必须保证电脑或其它访问本平台的软件有杀毒防毒能力，对于因客户中毒而产生的交易损失本平台不负任何责任；平台用户用妥善保管好个人密码，一旦发现丢失，需尽快提供相关证明交于服人员进行相关帐户锁定与资料的修改。

3.4 进行可行性分析的方法

市场可行性分析：调查目标市场的规模，分析目标市场的人口规模、购书需求、消费习惯等因素，不难发现用户的网上消费需求逐年增长，大多线下书店的经营模式转变为线上+线下的模式，故网上书店是被用户市场所需求的。

技术可行性分析：使用 SpringBoot 搭建项目基本的框架和 Mysql 数据库管理数据等的技术已经较为成熟。

经济可行性分析：网上书店主要以书籍、杂志及音像制品等商品为主，网站可提供书籍音像制品的片段供读者浏览，使消费者了解其内容后再订购，避免过多的退货问题。与传统书店相比，网上购书的市场规模和经营模式没有限制，突破了地理位置，经营面积，营业时间的局限。是一直种全新的销售渠道。从营业面积地理位置方面节省大量的资金。服务器 24 小时提供网络服务则消费者随时都能登陆网站购物，大大增加书店的销售额度。无论从商家还是从消费者的角度看网络书店都具有良好的发展前景。

法律可行性分析：符合法律法规。

4 可选的方案

总的开发流程就是根据系统需求进行总的设计，前端给出页面设计之后，数据库设计跟上，然后就是后端的开发，然后进行数据的链接以及前后端联调，并进行部署，系统就搭建完成。

4.1 可选的方案

4.1.1 可选的系统方案 1

我们想使用 SpringBoot 搭建项目基本的后端框架，快速构建应用程序，可以简化项目配置和搭建流程，提高开发效率，Spring Security 和 JWT：用于实现用户认证和权限控制，确保系统安全性，实现对数据库数据的操作的时候使用 Mybatis；

关于前端的设计，采取最原始的 jquery+html 以及 css 来实现界面设计或者 vue.js，前者更原始，后者效率更高，可以利用 Vue 脚手架搭建前端项目框架。Axios 来实现前端和后端数据的交互。并且结合 Vue 中特有的功能，可以实现对前端页面权限的配置(即决定哪些页面是可以不登录进行访问的，哪些页面需要某种权限才能访问)，通过集成 Markdown 编辑器插件，可以方便用户对图书信息进行文本描述，提升用户体验和信息展示的灵活性。

使用 MySQL 数据库管理系统数据。对于登录和密码，我们考虑选用 SpringSecurity 和 JWT 实现了用户的登录和权限控制以及密码加密等。此外我们考虑做一些数据库连接的优化，数据库的监控的操作，这可能会用到 Druid。

4.1.2 可选的系统方案 2

总体后端设计模式为 MVC，进行数据库相关操作，包括连接数据库、查询、更新等。使用 Java 类表示实体，与数据库表映射。JSP 页面负责展示数据，与用户进行交互。通过 HTML、CSS、JavaScript 实现页面布局和交互效果。处理用户请求，调用相关 Model 进行数据操作，然后将结果传递给 View。使用 Java Servlet 或其他框架作为控制器。

前端考虑使用 HTML 定义页面结构，CSS 负责样式布局 and 美化，确保页面清晰、美观且响应式。用 JavaScript 实现前端页面的动态交互和验证。JSP 与 Java 后端进行交互。

数据库设计使用 MySQL 数据库，设计用户表、图书表、订单表等。确保数据库范式设计，关联各个实体，保证数据一致性。使用 JDBC 实现 Java 与 MySQL 数据库的连接，执行 SQL 语句进行数据操作。

使用 Tomcat 作为 Java Web 应用的服务器，部署前后端代码，确保应用正常运行。

4.1.3 可选的系统方案 3

还可以使用 Python 的 Django 框架或 Node.js 等来开发后端功能，如用户管理、订单处理等。Django 提供了强大的 ORM 工具，可以简化数据库操作。Django 还包括用户认证系统、管理后台等，加速开发进程。

前端的开发，仍旧使用 HTML、CSS 和 JavaScript 构建网站前端，选择 Vue.js 前端框架实现用户界面和交互，提升用户体验。

数据库的搭建，使用 MongoDB 等 NoSQL 数据库来存储用户信息、订单数据等。更适用于灵活的数据模型和大量非结构化的数据的存储。

用 Nginx 或 Apache 服务器来托管网站，有更高的安全性。

4.2 原有方案的优缺点、局限性及存在的问题

4.2.1 方案 1

使用 Spring Boot 能够快速搭建后端框架，简化配置，提高开发效率。Spring Security 和 JWT 的结合提供了强大的用户认证和权限控制，有助于确保系统的安全性。MyBatis 作为持久层框架，能够提供对数据库操作的细粒度控制。使用 Vue.js 和 Axios 实现前后端分离，提高系统的可维护性和扩展性。

但是使用 jQuery 和原始 HTML/CSS，可能在现代前端开发中显得较为传统，可能影响用户体验。基于 Vue 的前端权限配置相对基础，可能在复杂场景下存在限制。使用多个框架和技术，需要确保它们之间的协同工作，增加了集成和维护的难度

4.2.2 方案 2

经典的 MVC 设计模式，易于理解和维护。虽然使用 JSP，但可以通过 Servlet 或其他框架实现前后端分离，提高可维护性。使用 MySQL，采用范式设计，有助于保持数据一致性。

但是使用 JSP 和传统的 Java Servlet，技术相对较老，不如现代框架灵活和高效。基于 JSP 的前端交互可能相对受限，难以实现现代前端开发的一些高级特性。使用 Tomcat 作为服务器，有时可能不如其他更轻量级的服务器，影响性能。

4.2.3 方案 3

Django 和 Vue.js 能提供更现代、高效的开发体验。Django 的 ORM 工具简化了数据库操作，提高了开发效率。使用 Vue.js 实现前后端分离，提高系统的可维护性和扩展性。MongoDB 适用于非结构化数据存储，灵活性更高。

但是书店项目只是一个小规模项目，使用 NoSQL 数据库和 Django 等可能显得过于灵活，增加了开发的复杂性。使用 Nginx 或 Apache 等服务器可能需要更多的配置和维护工作。

4.2.4 总体对比

方案一和方案三采用了较为现代的技术栈，更适合大多数项目的要求，特别是对于中小型团队和快速开发需求。方案二的技术选择相对陈旧，不太推荐使用。前后端分离能够提高系统的可维护性和扩展性，让我们更倾向选择方案一或方案三。

4.3 可重用的系统，与要求之间的差距

综合三种方案，我们认为在开发的过程中可以重用下述系统：

4.3.1 Spring Boot 配置

首先，Spring Boot 提供了自动配置的特性，大大简化了项目的搭建和配置过程。通过约定大于配置的方式，可以快速启动一个基于 Spring 的应用程序。其次，Spring Boot 允许将应用程序拆分为多个模块，每个模块负责不同的功能。这种模块化的设计使得配置可以被重用，并且易于管理和维护。而且，Spring Boot 支持将配置外部化，可以通过属性文件、环境变量、命令行参数等方式进行配置。这使得配置可以根据不同环境进行灵活调整，提高了应用程序的可移植性和灵活性。

4.3.2 Vue.js 前端

由于 Vue.js 采用了组件化的开发模式，将界面拆分为多个独立的组件，使得前端开发更加模块化、可复用。而且 Vue.js 的语法简洁易懂，学习难度和时间较低，同时提供了丰富的指令和功能，能够满足复杂的前端开发需求。Vue.js 拥有庞大的生态系统，包括 Vue Router、Vuex 等插件和工具，可以方便地扩展和定制前端功能。

4.3.3 Spring Security 和 JWT 配置

Spring Security 是一个功能强大的身份验证和访问控制框架，可以轻松集成到 Spring Boot 应用程序中，提供了各种安全功能，包括身份验证、授权、攻击防护等。

JDBC 和 MySQL 配置：JDBC 是 Java 数据库连接的标准接口，提供了对关系型数据库的访问能力，具有良好的性能和稳定性。此外，JDBC 提供了抽象层，使得应用程序可以与不同数据库进行交互而无需修改大量代码，提高了应用程序的可移植性。

4.4 选择最终方案的准则

通过小组讨论之后，我们小组确定选择方案 1，我们选择方案的准则有以下几点：

- 学习困难程度更小；
- 在规定的期限内，开发能更完整顺利的完成；
- 系统搭建的效率更高；
- 系统的安全性更高和权限控制；
- 页面权限配置和用户体验更好；

5 所建议的系统

5.1 对所建议的系统的说明

项目旨在建立一个完整的在线书店系统，包括前端和后端。前端使用 Vue.js 框架开发，提供用户友好的界面和良好的交互体验。后端使用 Spring Boot 框架开发，处理业务逻辑、数据库交互等功能，并使用 mysql 数据库管理数据。系统具备用户注册、登录、浏览图书、购买图书等基本功能，并添加了购物车、地址管理、个人信息管理等模块，同时支持管理员管理图书、订单等操作，另外实现了图书分类管理、书单管理以及数据统计模块。

5.2 数据流程和处理流程

用户浏览图书：用户通过前端界面浏览图书，前端向后端发送请求，后端从数据库中获取图书信息返回给前端。

用户购买图书：用户选择要购买的图书并进行下单，前端向后端发送订单信息，后端处理订单逻辑并更新数据库，同时生成订单信息返回给前端。

管理员管理图书：管理员登录系统后，可以添加、删除、修改图书信息，前端发送相应的请求到后端，后端进行相应的数据库操作。

5.3 与原系统的比较(若有原系统)

无原系统。

5.4 影响(或要求)

5.4.1 设备

暂时不考虑。

5.4.2 软件

开发需要安装能够搭建 Spring Boot 和 Vue.js 框架的软件。

5.4.3 运行

系统调试一周，在开始必须对初始数据进行录入，包括所有的书目信息等，这个期间还需要接受新用户的注册。在调试期结束，系统通过检验后，将要优化该系统的性能，把一些发挥不良的组件重新开发，或者加入合适的补丁文件。系统稳定后的主要工作是加强该系统

安全性。

5.4.4 开发

开发需要具备 Vue.js 和 Spring Boot 等相关技术的开发经验，并搭建新的开发环境。

5.4.5 环境

开发需要搭建适当的开发环境，包括 IDE、数据库等工具。

5.4.6 经费

开发需要购买软件。

5.5 局限性

暂时不考虑。

6 经济可行性(成本---效益分析)

6.1 投资

包括基本建设投资(如开发环境、设备、软件和资料等)，其他一次性和非一次性投资(如技术管理费、培训费、管理费、人员工资、奖金和差旅费等)。

6.1.1 基本建设投资

(1) 开发环境：

软件开发工具和服务，如集成开发环境（IDE）、数据库、前端和后端框架等。这些通常有免费的开源选项，但某些专业版可能需要付费。

版本控制系统，如 Git，以及在线代码托管平台，如 GitHub 或 GitLab，可能涉及到团队协作的高级功能订阅费用。

(2) 设备：

开发者和设计师所需的计算机硬件。

服务器硬件或云服务订阅（如 AWS、Google Cloud Platform 或阿里云），用于托管网站和相关服务。

(3) 软件和资料：

操作系统、数据库管理系统、服务器软件等的授权费用。

需要购买的专业书籍、在线课程或其他教育资源来提升团队能力。

6.1.2 其他一次性投资

（1）技术管理费：

初始设立项目时，可能需要外部咨询或技术支持的费用。

（2）培训费：

对团队成员进行新技术、工具或流程培训的费用。

（3）管理费：

公司注册、法律顾问费、财务审计等初始管理费用。

6.1.3 非一次性投资

（1）人员工资：

包括开发人员、设计师、项目经理、运营和客户服务团队的工资。

（2）奖金：

根据业绩或特殊贡献给予员工的奖金。

（3）差旅费：

团队成员出差、市场推广活动、参加行业会议等所产生的费用。

6.1.4 运营成本（长期投资）

（1）服务器和带宽费用：

网站运行所需的服务器租赁或云服务费用，以及足够的网络带宽费用。

（2）维护和更新费用：

定期对网站进行维护、更新和安全检查所需的费用。

（3）市场营销费用：

包括 SEO、社交媒体营销、电子邮件营销和在线广告等费用。

（4）客户服务费用：

提供客户支持所需的人力和技术工具费用。

6.2 预期的经济效益

6.2.1 一次性收益

一次性收益通常指项目启动初期可以获得的收益，对于网上书店来说，可能较为有限。这类收益可能来源于：

特别促销活动，如开业大促，吸引首批用户的购买。

初始合作伙伴关系的建立，如与出版社的特别合作促销。

6.2.2 非一次性收益

书籍销售收入：这是最直接的收入来源，包括会员和非会员的购书收入。

广告收入：随着网站流量的增加，可以通过展示广告获取收益。

会员费：如果设立了高级会员制度，提供额外服务，可以收取会员费。

数据分析服务：对用户行为进行分析，可能向出版社或第三方提供市场分析报告，作为收入来源之一。

6.2.3 不可定量的收益

品牌价值：随着用户基数和知名度的增加，品牌价值提升。

客户忠诚度：优质的服务和用户体验可以增加客户忠诚度，促进口碑营销。

市场数据：用户行为数据是宝贵的资源，有助于产品调整和市场战略规划。

6.2.4 收益/投资比

收益/投资比是衡量项目投资效益的重要指标。计算 ROI 需要减去项目总成本（包括初期投资和运营成本）从总收入中，然后除以总成本。理想情况下，ROI 应该大于 1，表明项目盈利。

6.2.5 投资回收周期

投资回收周期是指项目从开始投入到收回成本所需的时间。这个指标对投资者来说非常重要，因为它直接影响资金的流动性和项目的风险评估。网上书店的投资回收周期受多种因素影响，包括但不限于市场接受度、运营效率、成本控制等。

6.3 市场预测

市场需求：随着互联网的普及和便捷性，越来越多的人选择在网上购书。尤其是在忙碌的现代生活中，人们更喜欢在家中通过网站购书，省去了传统书店的时间和精力消耗。

潜在用户群体：潜在用户群体包括有限时间购书的上班族、热爱阅读的学生群体、喜欢网购的年轻群体等。这些群体对于一个方便快捷的网上书店平台会有较高的接受度。

竞争情况：网上书店市场竞争激烈，除了传统实体书店外，还有许多知名的在线书店存在。因此，网上书店项目需要在服务质量、图书种类、价格竞争等方面找到自己的竞争优势。

市场增长潜力：随着数字化阅读的普及和人们对知识、文化需求的增加，网上书店市场具有较大的增长潜力。同时，随着物流和支付方式的不断完善，网上购书的便捷性也在提升，这将进一步促进市场的增长。

7 技术可行性(技术风险评价)

开发人员:

开发团队需要具备 Java、JavaScript 等相关技术的开发经验，以及对 Spring Boot、Vue.js 等技术栈的熟悉程度。

如果团队成员对这些技术已经有一定的了解和经验，将有助于项目的顺利进行和高效开发。

现有技术:

该系统开发需求不高，硬件环境和软件基础已经十分成熟和完善。

(1) 后端技术栈 (Spring Boot)

成熟稳定性: Spring Boot 作为一种轻量级的 Java 开发框架，已经在许多项目中得到了广泛应用，拥有庞大的社区支持和活跃的开发群体。

生态系统: Spring Boot 提供了丰富的生态系统和功能模块，包括安全性、数据访问、消息队列等，可以满足项目的各种需求。

灵活性和可扩展性: Spring Boot 提供了许多灵活的配置选项和扩展机制，使得开发团队能够根据项目需求进行定制开发和扩展。

(2) 前端技术栈 (Vue.js)

现代化框架: Vue.js 是一种轻量级、灵活的 JavaScript 框架，拥有简单易学的 API 和响应式的数据绑定机制，适合构建现代化的单页面应用。

生态系统: Vue.js 有着庞大的生态系统和丰富的第三方组件库，可以加速项目开发并提高开发效率。

性能优化: Vue.js 具有优秀的性能表现，包括虚拟 DOM、异步更新等机制，能够有效提升前端页面的加载速度和用户体验。

(3) 数据库技术 (MySQL)

稳定性和可靠性: MySQL 是一种成熟稳定的关系型数据库管理系统，被广泛应用于各种 Web 应用开发中。

兼容性和易用性: MySQL 支持多种操作系统和开发语言，并且具有良好的用户界面和文档支持，易于学习和使用。

技术风险:

(1) 隐私信息保护

系统需要对用户、管理员的个人信息进行保护，防止数据泄露。

(2) 系统性能和可靠性

系统需要长期运行，因此需要保证系统的可靠性和稳定性。

(3) 系统兼容性

系统需要保证能够在多种操作系统平台都能运行，因此需要充分测试。

8 法律可行性

政府，无论是中央政府还是地方政府，一般都用法律规定组织可以做什么，不可以做什么。例如：《合同法》，《消费者权益保护法》，《专利法》，《反不正当竞争法》等对所有企业的行为都做了限制。我们公司就设有自己的法律顾问，这是因为法规的影响不仅仅限

于时间和金钱，它还缩小了管理者可斟酌决定的范围，限制了可行方案的选择。

根据《中华人民共和国计算机软件保护条例》（1991年6月4日中华人民共和国国务院令 第84号发布）（修改后的新条例 2002年1月1日起施行）第九条可知：软件著作权人享有下列各项权利：

- (一)发表权，即决定软件是否公之于众的权利；
- (二)开发者身份权，即表明开发者身份的权利以及在其软件上署名的权利；
- (三)使用权，即在不损害社会公共利益的前提下，以复制、展示、发行、修改、翻译、注释等方式使用其软件的权利；
- (四)使用许可权和获得报酬权，即许可他人以本条第(三)项中规定的部分或者全部方式使用其软件的权利和由此而获得报酬的权利；
- (五)转让权，即向他人转让由本条第(三)项和第(四)项规定的使用权和使用许可权的权利。

第十二条 受他人委托开发的软件，其著作权的归属由委托者与受委托者签订书面协议约定，如无书面协议或者在协议中未作明确约定，其著作权属于受委托者。

第十八条 在软件著作权的保护期内，软件的著作权人或者其受让者有权许可他人行使本条例第九条第(三)项规定的使用权。著作权人或者其受让者许可他人行使使用权时，可以按协议收取费用。

另外，软件权利的使用许可应当根据我国有关法规以签订、执行书面合同的方式进行。被许可人应当在合同规定的方式、条件、范围和时间内行使使用权。合同中未明确规定为独占许可的，被许可的软件权利应当视为非独占的。

第二十二条 因课堂教学、科学研究、国家机关执行公务等非商业性目的需要对软件进行少量的复制，可以不经软件著作权人或者其合法受让者的同意，不向其支付报酬。但使用时应当说明该软件的名称、开发者，并且不得侵犯著作权人或者其合法受让者依本条例所享有的其他各项权利。该复制品使用完毕后应当妥善保管、收回或者销毁，不得用于其他目的或者向他人提供。

第三十条 除本条例第二十一条及第二十二条规定的情况外，有下列侵权行为的，应当根据情况，承担停止侵害、消除影响、公开赔礼道歉、赔偿损失等民事责任，并可以由国家软件著作权行政管理部门给予没收非法所得、罚款等行政处罚：

- (一)未经软件著作权人同意发表其软件作品；
- (二)将他人开发的软件当作自己的作品发表；
- (三)未经合作者同意，将与他人合作开发的软件当作自己单独完成的作品发表；
- (四)在他人开发的软件上署名或者涂改他人开发的软件上的署名；
- (五)未经软件著作权人或者其合法受让者的同意修改、翻译、注释其软件作品；
- (六)未经软件著作权人或者其合法受让者的同意复制或者部分复制其软件作品；
- (七)未经软件著作权人或者其合法受让者的同意向公众发行、展示其软件的复制品；
- (八)未经软件著作权人或者其合法受让者的同意向任何第三方办理其软件的许可使用或者转让事宜。

第三十三条 当事人不履行合同义务或者履行合法义务不符合约定条件的，应当依照民法通则有关规定承担民事责任。

9 用户使用可行性

该系统是一个电子商务网站，所以操作简易，用户购书不需要了解整个书店的工作流程，

只需要按照提示一步步操作即可完成购书等其他活动。

10 其他与项目有关的问题

在完成给定项目要求的前提下，小组参考其他网上书店的模式，可能会调整完善本次网上书店项目的各项功能，使得项目的应用效率更高。

未来网上书店项目可能会面临很多风险，这些风险涉及到技术、运营、市场等方面。下面是对可能存在的风险进行细化、分级以及应对预案的分析。

(1) 技术风险

① 网站安全漏洞

风险等级：高

可能原因：代码编写不规范、未经充分测试、第三方组件漏洞等。

应对预案：定期进行安全漏洞扫描和修补，采用安全加固措施，如加密传输、用户身份验证、防止 SQL 注入等；建立应急响应机制，及时应对可能的安全威胁。

② 数据泄露或丢失

风险等级：高

可能原因：管理员账号信息泄露导致用户信息、公司核心信息泄露；管理员操作失误导致信息被删除。

应对预案：对管理员登录添加多重验证；采用日志记录管理员操作，如果管理员操作失误可以及时回溯；对信息进行多份备份并分地存储。

③ 软件安全性

风险等级：高

可能原因：系统自身运行出错；黑客攻击，利用软件漏洞窃密或篡改信息

应对预案：设计并实现更完善的软件安全体系；对数据进行加密和备份。

④ 技术故障

风险等级：中

可能原因：硬件故障、软件错误、网络问题等。

应对预案：实施高可用性架构，使用负载均衡和多台服务器配置，定期备份数据并建立灾备系统，确保系统能够在故障情况下迅速恢复。

(2) 运营风险

① 物流配送延迟或损坏

风险等级：中

可能原因：天气影响、仓库管理不善等。

应对预案：选择可靠的物流合作伙伴，建立物流跟踪系统，实时监控货物运输情况。与物流公司签订明确的服务协议，对延迟或损坏情况进行赔偿处理。

② 商品库存管理不足或过剩

风险等级：中

可能原因：由于市场需求变化、对市场需求预测不准确、采购计划失误等。

应对预案：建立科学的库存管理系统，实施定期盘点和库存预警机制，根据历史销售数据和市场需求进行合理的库存规划和调整。

(3) 市场风险

① 竞争激烈导致市场份额下降

风险等级：高

可能分析：同行竞争加剧、用户需求变化等。

应对预案：进行市场调研，了解竞争对手的优势和劣势，制定差异化竞争策略，提升品牌影响力和用户体验。开展促销活动，提高品牌知名度和用户粘性。

② 法律法规变化导致合规风险

风险等级：中

可能分析：政策调整、法律法规更新、监管要求变化。

应对预案：密切关注相关法律法规的变化，保持良好的合规意识，及时调整业务流程和服务模式，确保网站运营合法合规。建立法律顾问团队，处理潜在的法律风险和纠纷。

11 注解

【本章应包含有助于理解本文档的一般信息(例如原理)。本章应包含为理解本文档需要的术语和定义，所有缩略语和它们在文档中的含义的字母序列表。】

附录

【附录可用来提供那些为便于文档维护而单独出版的信息(例如图表、分类数据)。为便于处理附录可单独装订成册。附录应按字母顺序(A, B 等)编排。】