logo：

徽标, 公司名称

描述已自动生成

### 总

### 体

### 设

### 计

**G02小组**

**二手书交易系统**

目录

[校园二手书交易网站总体设计方案 3](#_Toc25807)

[1. 设想供选择的方案 3](#_Toc19493)

[2. 选取合理的方案 4](#_Toc25459)

[3. 推荐最佳方案 8](#_Toc32133)

[4. 功能分解 8](#_Toc30604)

[5. 设计软件结构 9](#_Toc14389)

[6. 设计数据库 11](#_Toc5039)

[7. 制定测试计划 12](#_Toc27603)

[1. 测试策略 13](#_Toc6875)

[2. 测试方案 13](#_Toc16857)

[3. 预期的测试结果 14](#_Toc24912)

[4. 测试进度计划 15](#_Toc29571)

[5. 人员分工 16](#_Toc8530)

[6. 风险管理 16](#_Toc2077)

[8. 书写文档 16](#_Toc27337)

[9. 审查和复审 17](#_Toc1676)

### 

### **校园二手书交易网站总体设计方案**

#### **1. 设想供选择的方案**

在需求分析阶段，我们已经明确了校园二手书交易平台的主要功能需求，包括用户注册登录、书籍信息发布、搜索浏览、在线交流、订单管理等。基于这些需求，我们可以设想以下几个可能的实现方案：

****集中式架构****：所有服务运行在同一服务器上，适用于初期用户数量不多的情况。

****分布式架构****：将不同服务（如用户服务、书籍信息管理、支付接口等）部署在不同的服务器上，适合用户量大、并发访问高的场景。

****微服务架构****：将整个系统拆分成多个小型、独立的服务，每个服务负责一部分业务逻辑，通过API接口进行通信。这种架构具有很高的灵活性和可扩展性，但实现起来相对复杂。

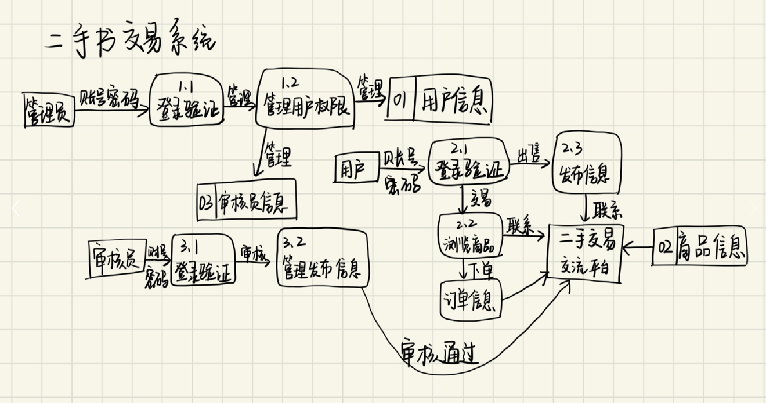
#### **2. 选取合理的方案**

经过初步评估，我们选择了以下三种方案进行深入分析：

* ****低成本方案****：采用集中式架构，使用开源软件和免费云服务，减少初期投入。
* ****中等成本方案****：采用分布式架构，部分关键服务（如支付、数据库）使用付费云服务，确保系统稳定性和安全性。
* ****高成本方案****：采用微服务架构，全面使用高性能云服务，支持大规模并发访问，具备高度的可扩展性和灵活性。

对于每个方案，我们将准备以下资料：

****系统流程图****：



****物理元素清单****：运行环境：推荐jdk1.8；开发工具：eclipse以及idea（推荐）、vscode、redis、node环境（16.X）；操作系统：windows 10 8G内存以上（其他windows以及macOS支持，但不推荐）；浏览器：Firefox(推荐)、Google Chrome(推荐)、Edge;数据库：MySQL8.0(推荐)及其他版本（支持，但容易异常尤其MySQL5.7（不含）以下版本）；数据库可视化工具：Navicat Premium 15（推荐）以及其他Navicat版本是否maven项目：是项目技术后端：Springboot、Mybatis、mysql前端：html、elemetui、vue、axios

****成本/效益分析****：

开发费用：约￥50,000（包括前后端开发、数据库设计、功能集成，人员工资一个人15小时每周，时薪90）

服务器及托管费用：约￥5,000（包含初期服务器采购、域名和云服务费用）

推广费用：约￥10,000（包含线上推广、校园宣传材料）  
**长期成本**：包括系统维护、升级、日常运营、技术支持等。

年度维护费用：约￥30,000（包括服务器费用和技术人员支持）

### 收益预测

收入来源主要包括以下几个方面：

**交易佣金**：从每笔交易中收取一定比例的佣金（例如3% - 5%）。

**广告收入**：通过在平台展示广告获取收益。

**增值服务**：为用户提供VIP服务，优先展示书籍，提升搜索排名。

****进度计划****：任务名称 工期 开始时间 完成时间

软件定义阶段 35 个工作日? 2024年10月1日 2024年11月18日

项目计划书设计 10 个工作日? 2024年10月1日 2024年10月14日

文档撰写 10 个工作日? 2024年10月1日 2024年10月14日

 PPT制作 10 个工作日? 2024年10月1日 2024年10月14日

项目数据调查 10 个工作日? 2024年10月1日 2024年10月14日

可行性研究 5 个工作日? 2024年10月15日 2024年10月21日

数据流程图 5 个工作日? 2024年10月15日 2024年10月21日

系统流程图 5 个工作日? 2024年10月15日 2024年10月21日

文档撰写 5 个工作日? 2024年10月15日 2024年10月21日

需求分析 5 个工作日? 2024年10月22日 2024年10月28日

sRS设计 5 个工作日 2024年10月22日 2024年10月28日

ppt制作 5 个工作日 2024年10月22日 2024年10月28日

E-R图制作，PPT内容总结 5 个工作日 2024年10月22日 2024年10月28日

总体设计 5 个工作日 2024年10月29日 2024年11月4日

详细设计 5 个工作日 2024年11月5日 2024年11月11日

项目设计评审 5 个工作日 2024年11月12日 2024年11月18日

项目实现阶段 25 个工作日? 2024年11月19日 2024年12月23日

程序编码 15 个工作日 2024年11月19日 2024年12月9日

实现 + 翻转1 5 个工作日 2024年11月19日 2024年11月25日

维护 + 翻转2 5 个工作日 2024年11月26日 2024年12月2日

软件测试 7 个工作日 2024年12月9日 2024年12月17日

项目总结维护 5 个工作日 2024年12月17日 2024年12月23日

#### **3. 推荐最佳方案**

综合考虑技术可行性、成本效益和用户体验，我们推荐低成本方案。该方案既能满足当前用户需求，又具有一定的扩展性，能够应对未来用户增长带来的挑战。我们将提交该方案给用户和技术专家进行审查，确保其符合实际需求并具备可行性。

#### **4. 功能分解**

为了实现校园二手书交易平台的各项功能，我们需要对系统进行功能分解：

****用户管理模块****：负责用户注册、登录、个人信息管理等功能。

****书籍信息发布模块****：允许用户发布书籍信息，包括书名、作者、价格、图片等。

****搜索浏览模块****：提供强大的搜索功能，支持关键词搜索和分类浏览。

****在线交流模块****：用户可以通过站内消息或即时聊天工具进行交流。

****订单管理模块****：处理用户的购买请求，生成订单并跟踪订单状态。

****支付模块****：集成第三方支付接口，支持多种支付方式。

****评价反馈模块****：用户可以对交易进行评价，提高平台的透明度和信任度。

#### **5. 设计软件结构**

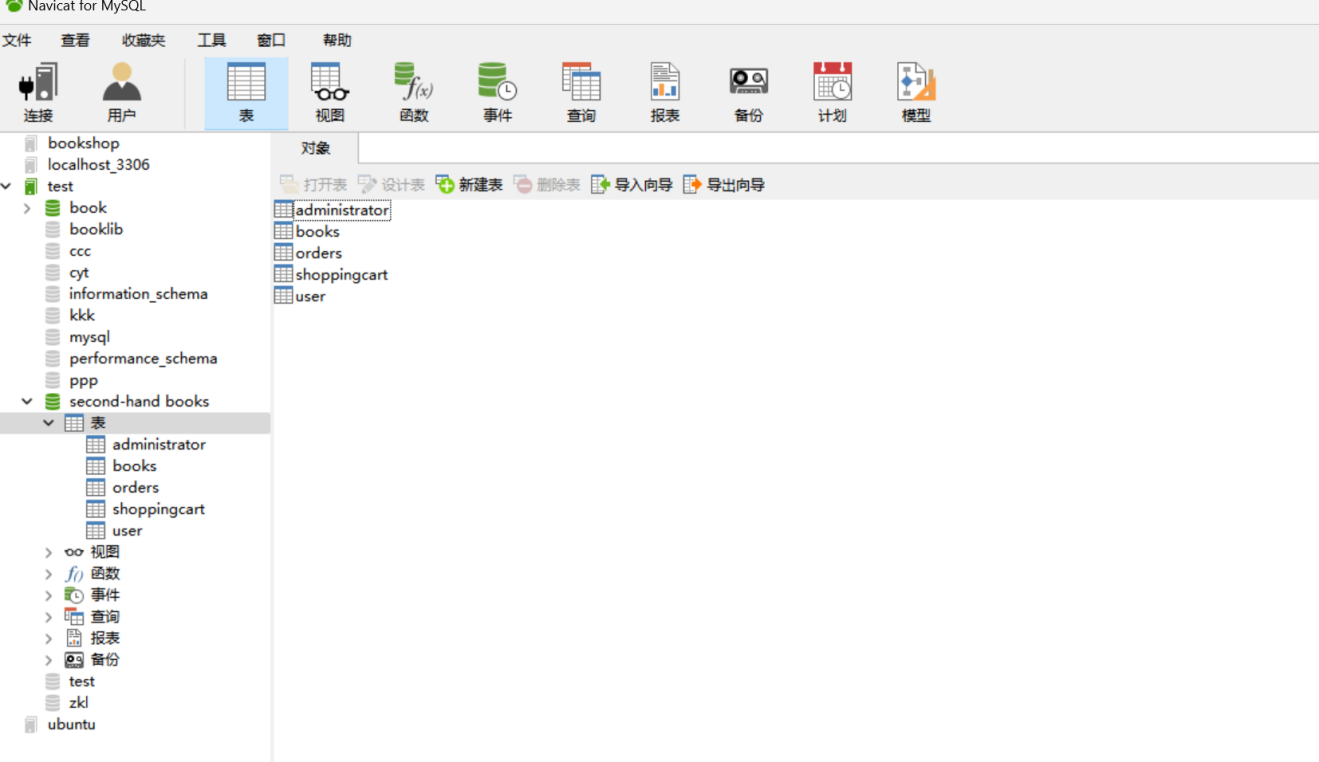
我们将采用模块化设计，每个模块负责一个特定的子功能。模块之间通过明确定义的接口进行通信，确保系统的松耦合和高内聚。主要模块结构如下：

* ****用户管理模块****
  + 用户注册
  + 用户登录
  + 个人信息管理
* ****书籍信息发布模块****
  + 发布书籍信息
  + 编辑书籍信息
  + 删除书籍信息
* ****搜索浏览模块****
  + 关键词搜索
  + 分类浏览
  + 搜索结果排序
* ****在线交流模块****
  + 站内消息
  + 即时聊天
* ****订单管理模块****
  + 创建订单
  + 查看订单状态
  + 取消订单
* ****支付模块****
  + 支付请求
  + 支付确认
  + 支付失败处理
* ****评价反馈模块****
  + 发布评价
  + 查看评价
  + 回复评价

#### **6. 设计数据库**

根据需求分析阶段确定的数据需求，我们将设计以下数据库表：

* ****用户表****：存储用户基本信息，如用户名、密码、邮箱、手机号等。
* ****书籍信息表****：存储书籍详细信息，如书名、作者、价格、图片、描述等。
* ****订单表****：记录用户的订单信息，包括订单号、买家ID、卖家ID、书籍ID、订单状态等。
* ****消息表****：存储用户之间的交流记录，包括发送者ID、接收者ID、消息内容、发送时间等。
* ****评价表****：记录用户对交易的评价信息，包括评价者ID、被评价者ID、评价内容、评分等。



#### **7. 制定测试计划**

#### 1. 测试策略

##### 1.1 目的

* 本测试计划旨在确保二手书交易网站的各项功能和性能符合设计规范和用户需求，通过系统化的测试过程发现并修复潜在的问题，提高软件的质量和稳定性。

##### 1.2 范围

* 本测试计划涵盖二手书交易网站的所有核心功能模块，包括但不限于用户注册与登录、书籍信息发布与管理、搜索与筛选、在线聊天与留言、交易管理和个人账户设置等。

##### 1.3 方法

* + **单元测试**：针对每个功能模块的内部逻辑进行测试，确保每个模块独立运行正常。
  + **集成测试**：测试不同模块之间的接口和交互，确保整个系统协同工作。
  + **系统测试**：模拟真实使用环境，对整个系统进行全面测试，包括功能性、性能、安全性等方面。
  + **验收测试**：邀请部分用户参与测试，收集反馈意见，确保软件满足用户需求。

#### 2. 测试方案

##### 2.1 测试环境

* + 硬件环境：
    - 处理器：Intel Core i5 或更高
    - 内存：8GB 或更高
    - 存储：256GB SSD 或更高
    - 显示器：1080p 分辨率
    - 网络：100Mbps 以上宽带
  + 软件环境：
    - 操作系统：Windows 10, macOS Catalina, Android 5.0+, iOS 9.0+
    - 浏览器：Chrome 80+, Firefox 70+, Safari 13+, Edge 80+
    - 支持软件：MySQL 5.7, Apache 2.4, Node.js 12.0+

##### 2.2 测试工具

* + **自动化测试工具**：Selenium, JMeter
  + **性能测试工具**：LoadRunner, Gatling
  + **缺陷管理工具**：Jira, Bugzilla

##### 2.3 测试类型

* + **功能测试**：验证系统功能是否按预期工作。
  + **性能测试**：评估系统在高并发情况下的表现。
  + **安全测试**：检查系统的安全性和数据保护机制。
  + **兼容性测试**：确保系统在不同浏览器和设备上的兼容性。
  + **用户体验测试**：评估用户界面的友好性和易用性。

#### 3. 预期的测试结果

##### 3.1 功能测试

* + **用户注册与登录**：验证用户注册、登录、注销等功能的正确性。
  + **书籍信息发布与管理**：确保用户可以成功发布、编辑和删除书籍信息。
  + **搜索与筛选**：验证搜索功能的准确性和筛选条件的有效性。
  + **在线聊天与留言**：确保用户可以顺利进行文字和图片的交流。
  + **交易管理**：测试交易流程的完整性，包括下单、支付、确认收货等环节。
  + **个人账户设置**：验证用户可以修改个人信息、查看交易记录等。

##### 3.2 性能测试

* + **响应时间**：所有页面加载时间不超过3秒。
  + **并发用户**：系统可以稳定支持1000个并发用户。
  + **负载测试**：在高负载下，系统性能下降不超过20%。

##### 3.3 安全测试

* + **数据加密**：确保所有敏感数据（如密码、支付信息）均经过加密处理。
  + **防注入攻击**：防止SQL注入和XSS攻击。
  + **权限控制**：确保只有授权用户可以访问和修改数据。

##### 3.4 兼容性测试

* + **浏览器兼容性**：确保系统在主流浏览器上正常运行。
  + **设备兼容性**：确保系统在不同设备上（如PC、手机、平板）正常运行。

##### 3.5 用户体验测试

* + **界面友好性**：确保用户界面简洁明了，易于操作。
  + **响应速度**：确保用户操作后的响应时间短。
  + **错误提示**：确保系统在出错时能够提供清晰的错误提示。

#### 4. 测试进度计划

##### 4.1 准备阶段

* + **时间**：2024年11月10日 - 2024年11月15日
  + 任务：
    - 制定详细的测试计划和测试用例。
    - 准备测试环境和测试工具。
    - 培训测试团队成员。

##### 4.2 执行阶段

* + **时间**：2024年11月16日 - 2024年11月30日
  + 任务：
    - 执行单元测试、集成测试、系统测试和验收测试。
    - 记录测试结果，提交缺陷报告。
    - 修复发现的问题，重新测试。

##### 4.3 回归测试

* + **时间**：2024年12月1日 - 2024年12月5日
  + 任务：
    - 对修复的问题进行回归测试，确保问题已解决且没有引入新的问题。
    - 最终确认系统功能和性能符合要求。

##### 4.4 总结阶段

* + **时间**：2024年12月6日 - 2024年12月10日
  + 任务：
    - 编写测试总结报告，分析测试结果。
    - 提交最终测试报告，评估软件质量。
    - 准备上线前的最后准备工作。

#### 5. 人员分工

**项目经理**：负责整体测试计划的制定和协调。莫睿廷

**测试经理**：负责测试方案的实施和测试团队的管理。宋世博

**开发工程师**：负责修复测试中发现的问题。徐炳凯

**运维工程师**：负责测试环境的搭建和维护。徐炳凯，宋世博

#### 6. 风险管理

* + 风险识别：
    - 测试环境搭建延迟。
    - 测试工具兼容性问题。
    - 测试人员技能不足。
    - 发现重大缺陷导致测试延期。
  + 应对措施：
    - 提前准备备用测试环境。
    - 选择成熟的测试工具并进行预测试。
    - 提供充分的培训和技术支持。
    - 制定详细的缺陷管理流程，确保及时修复问题。

#### **8. 书写文档**

为了记录总体设计的结果，我们将编写以下文档：

* ****系统说明书****：详细描述系统的整体架构、模块划分和功能实现。
* ****用户手册****：提供用户操作指南，帮助用户快速上手使用平台。
* ****测试计划****：包括测试策略、测试方案、预期的测试结果和测试进度计划。
* ****详细的实现计划****：制定详细的开发时间表，分配任务和资源。
* ****数据库设计结果****：详细描述数据库表结构和关系。

#### **9. 审查和复审**

在完成上述设计工作后，我们将对总体设计结果进行严格的技术审查，确保设计方案的合理性和可行性。审查通过后，我们将提交给客户从管理角度进行复审，确保方案符合客户的实际需求和期望。通过审查后，我们将进入下一阶段的结构设计。

客户代表：软工2203班王家瑞，复审结果：功能符合需求，也保证了用户隐私。结构设计合理