



山东大学
SHANDONG UNIVERSITY

软件工程与实践

网上书店——体系结构设计文档

板块	初稿	审阅修改
范围、引用文件、CSCI 级设计决策	庞鑫	王子瑞
体系结构、全局变量说明、CSCI 部件	王子瑞	鲍泽雨
执行概念、接口设计（接口标识、INTF01-03）	鲍泽雨	赵忆楠
接口设计（INTF04-09）、需求的优先顺序和关键程度	赵忆楠	刘惠国
CSCI 详细设计、需求的可追踪性	刘惠国	庞鑫

团队成员：

庞 鑫（202200130172） 王子瑞（202200130095）

鲍泽雨（202200130120） 刘惠国（202200130019）

赵忆楠（202200130180）

目录

1 引言	4
1.1 标识	4
1.2 系统概述	4
1.3 文档概述	4
2 引用文件	5
3 CSCI 级设计决策	5
3.1 CSCI 行为设计决策（用户视角功能行为）	5
3.1.1 输入/输出接口设计决策	5
3.1.2 系统行为设计决策	5
3.1.3 安全、保密性设计决策	6
3.2 软件配置项选择与设计决策	6
3.2.1 技术选型	6
3.2.2 数据模型设计	6
3.2.3 针对关键性需求的专门设计决策	6
3.2.4 与系统状态/运行方式的依赖性	7
3.3 其他 CSCI 级设计决策	7
3.3.1 灵活性与扩展性	7
3.3.2 可维护性与监控机制	8
4 CSCI 体系结构设计	8
4.1 体系结构	9
4.1.1 程序(模块)划分	9
4.1.2 程序(模块)层次结构关系	12
4.2 全局数据结构说明	13
4.2.1 常量	13
4.2.2 变量	14
4.2.3 数据结构	14
4.3 CSCI 部件	16
4.3.1 软件配置项	16
4.3.2 静态结构关系	16
4.3.3 配置项用途与需求分配	17
4.3.4 开发状态说明	18
4.3.5 资源使用说明	18
4.4 执行概念	18
4.4.1 执行控制流	18
4.4.2 数据流	19
4.4.3 动态控制序列	19
4.4.4 资源管理	20

4.4.5 动态特性设计	20
4.4.6 状态转换	20
4.4.7 并发执行与异常处理	21
4.5 接口设计	22
4.5.1 接口标识	22
4.5.2 INTF-01 用户界面接口	22
4.5.3 INTF-02 支付系统接口	23
4.5.4 INTF-03 库存同步接口	24
4.5.5 INTF-04 物流状态通知接口	25
4.5.6 INTF-05 邮件与短信推送接口	26
4.5.7 INTF-06 数据库硬件接口	26
4.5.8 INTF-07 用户服务与订单接口	27
4.5.9 INTF-08 库存服务与订单接口	28
4.5.10 INTF-09 支付服务与订单接口	29
4.5.11 需求的优先顺序和关键程度	29
5 CSCI 详细设计	30
6 需求的可追踪性	31
6.1 从软件配置项到 CSCI 需求的可追踪性	31
6.2 从 CSCI 需求到软件配置项的可追踪性	33

1 引言

1.1 标识

系统名称：书海在线（BookOcean）网上书店系统

系统标识号：BOS-2025-PRJ001

软件名称：BookOcean 网上书店平台

软件缩略名称：BOS（BookOcean System）

软件版本号：v1.1.0（主版本.次版本.修订号）

发行号：Release-2025-Q2（按年度季度发行）

适用文档版本：需求规格说明书 v1.1、设计文档 v1.0

支付网关接口：PG-API v2.3

物流追踪系统：LTS v1.5

预计发布日期：2025 年 5 月 2 日

1.2 系统概述

网上书店系统旨在为图书销售业务提供数字化、智能化的线上购书解决方案，通过构建一个集图书展示、在线交易、库存管理、客户服务等功能于一体的电子商务平台，突破传统实体书店的地域限制，扩大市场覆盖范围。该系统的建设不仅有助于优化运营流程、提高库存管理效率，还能提升用户购物体验，推动书店实现线上线下业务的协同发展，增强市场竞争力。

1.3 文档概述

无论采取什么样的方法来定义需求，我们都必须使用一组文档来记录结果。我们和客户在整个开发过程和维护过程中都会查阅这些文档。本文档旨在明确“网上书店”的功能性需求、非功能性需求以及相关约束条件，为开发团队、测试团队、产品经理及其他利益相关者提供清晰、完整的软件需求定义，确保最终交付的系统符合业务目标和用户期望。

本文件属于内部资料，未经明确授权，禁止向外部人员或未授权方披露、复制或传播。

文档版本更新将通知授权用户，如有疑问或建议，请联系 18272892200。

2 引用文件

(1) GB T-8567-2006 计算机软件文档编制规范

(2) 《掌握需求过程(第3版)》

3 CSCI 级设计决策

3.1 CSCI 行为设计决策（用户视角功能行为）

3.1.1 输入/输出接口设计决策

用户界面（INTF-01）：所有核心功能必须通过 Web 前端提供人机交互，包括注册、登录、浏览图书、下单、支付、查看订单等，采用响应式设计以支持多终端访问。

支付接口（INTF-02）：调用支付宝/微信等支付网关，使用 RESTful 接口方式并配合 OAuth 2.0 鉴权，实现支付确认和支付状态异步通知。

数据库接口（INTF-06）：底层数据交互通过 MySQL 驱动，配合 JPA 框架自动完成对象关系映射，保障数据一致性。

3.1.2 系统行为设计决策

行为响应机制：

注册/登录采用 JWT 认证，支持 Token 刷新机制，实现前后端分离下的会话管理。

图书搜索支持 Elasticsearch 模糊查询，提供关键词自动补全和中英文语义匹配。

订单 15 分钟内未支付自动取消，并释放库存。

所有用户操作（如留言、退货）均记录时间戳和操作人，实现行为可追踪。

性能指标：首页加载时间 $\leq 1s$ ；搜索响应时间 $\leq 1s$ ；并发处理能力 ≥ 1000 用户注册登录请求/s。

3.1.3 安全、保密性设计决策

隐私数据脱敏：用户手机号、地址等个人数据仅限授权可见，展示时自动脱敏处理。

传输安全，系统统一通过 HTTPS 进行通信，防止中间人攻击。

数据加密，敏感信息如密码、支付信息采用 AES-256 加密。

访问控制，基于 RBAC 模型实现权限隔离，防止越权访问。

操作审计，所有关键操作生成审计日志，并强制加密存储，支持合规审计。

3.2 软件配置项选择与设计决策

3.2.1 技术选型

后端框架采用 Spring Boot 3.0，支持微服务架构与高并发处理。

前端框架采用 Vue 3 + Element UI，配合 Pinia 做状态管理。

缓存中间件采用 Redis 用于存储会话数据和热点图书信息。

搜索引擎使用 Elasticsearch 用于支持模糊检索和推荐算法。

数据库使用 MySQL 8.0，事务型关系数据库，满足 ACID 需求。

接口协议使用统一采用 RESTful API，接口安全通过 JWT + HTTPS 保障。

3.2.2 数据模型设计

所有主数据表（如用户、订单、图书）以 UUID 作为主键。

支持软删除机制，保留删除记录以供审计。

所有金额字段均使用 Decimal(10,2) 表示，防止浮点误差。

3.2.3 针对关键性需求的专门设计决策

安全性上设置，登录错误 5 次锁定账号 30 分钟；前端密码加密传输，后台存储哈希。

保密性设置上，订单、留言、后台操作等均需记录操作人、时间及详细行为。

私密性设置上，未登录用户无法访问用户数据、订单信息等敏感功能。

3.2.4 与系统状态/运行方式的依赖性

紧急状态下，以最大化系统安全性，防止数据篡改或泄露为目标。系统切换至只读模式，所有写入操作（如用户下单、数据修改）被禁止，仅允许查询类操作；依赖项为权限控制模块：动态关闭写入权限；日志监控系统，记录攻击来源及异常请求，触发告警；前端提示用户操作时显示“系统紧急维护中，仅可浏览”。自动隔离措施，可疑 IP 或账号被临时封禁，依赖实时风控系统的检测规则。触发条件是安全防护系统（如 WAF、IDS）检测到 DDoS 攻击、SQL 注入等威胁。

降级状态，出现在资源不足或部分服务故障，目标是保障核心业务流程可用，牺牲非关键功能。关闭推荐系统、广告等非核心功能。依赖项如下所示，服务熔断机制：自动切断对推荐系统 API 的调用；负载均衡器，将流量导向核心服务（如支付、商品详情页）。简化页面渲染，前端降级为静态内容，依赖 CDN 缓存或备用模板；限流措施，当交易系统压力过大时，触发排队机制（如秒杀场景）；触发条件，CPU/内存使用率超过阈值、第三方服务（如推荐引擎）超时。

维护状态，是在计划内升级或修复。为了平衡用户体验与后台管理需求。前台显示维护页，用户访问时返回 503 状态码及友好提示页，依赖运维人员预设的维护开关；后台允许数据维护操作，仅限管理员通过特定入口（如 VPN 或内网 IP）访问后台系统；依赖项，权限白名单：仅允许运维账号登录。数据库代理，主库禁止应用连接，仅开放给管理工具；触发条件，人工手动激活维护模式，或自动化工具检测到需紧急修复的漏洞。

3.3 其他 CSCI 级设计决策

3.3.1 灵活性与扩展性

模块化设计，预留扩展点，如以下所示。

会员体系扩展，支持多级会员（如普通/VIP/黑钻）、积分兑换、订阅制等，未来可无缝接入第三方权益平台（如航空里程、电商优惠）。

图书租赁模块，预留租赁计费规则接口（按天/周/月）、库存动态分配策略（防止租赁与销售冲突）。

电子书阅读扩展，支持 EPUB/PDF 格式解析，未来可扩展听书功能，依赖统一的媒体文件存储服务（如 AWS S3）。

核心配置动态化，库存阈值，根据商品热度自动调整（如畅销书阈值提高），或由运营人员在管理后台手动修改；支付超时时间，区分支付方式（支付宝/信用卡）设置不同超时策略，避免因第三方接口延迟导致订单失效；动态生效，通过配置中心（如 Nacos）实时推送至业务逻辑，无需重启服务。

接口版本化管理，语义化版本控制（如 `/api/v1/books`、`/api/v2/books`），

旧版本保留至少 3 个迭代周期。兼容性策略是新增字段时，旧接口返回默认值；废弃字段标记为 deprecated，前端逐步迁移；通过 Swagger 文档自动同步变更至开发者。

3.3.2 可维护性与监控机制

热部署，基于 Spring Boot DevTools 或 JRebel，修改代码后自动热加载，减少重启时间。

接口 Mock，前端通过 YApi 或 Postman Mock Server 模拟后端响应。

后端使用 WireMock 隔离依赖服务，实现并行开发。

采用实时监控与告警，使用 Prometheus + Grafana 监控方案，服务状态包含 CPU/内存、线程池队列、GC 频率等内容；接口指标包含 QPS、平均响应时间（P99/P95）、错误率（4xx/5xx）；业务数据包含每日订单量、库存周转率，通过自定义 Exporter 采集。

告警规则，自动触发企业微信/钉钉通知（如“支付接口错误率>1%持续 5 分钟”）。

自动化压力测试，定期执行场景，接口压测通过 JMeter 模拟高并发下单、查询，检测数据库连接池瓶颈；数据库压测使用 SysBench 评估索引效率，识别慢查询。报告自动对比历史数据（如 TPS 下降超过 10%标红预警）生成。

4 CSCI 体系结构设计

本章应分条描述 CSCI 体系结构设计。如果设计的部分或全部依赖于系统状态或方式，则应指出这种依赖性。如果设计信息在多条中出现，则可只描述一次，而在其他条引用。应给出或引用为理解这些设计所需的设计约定。

结合 Vue3 前端 + Spring Boot 后端（包含 Controller、Service、Mapper、Entity 分层）的技术栈，这种项目属于典型的前后端分离架构，后端采用分层架构（MVC）。

特点：

- （1）前后端完全解耦，可独立开发、部署。
- （2）前端负责渲染和用户交互，后端专注业务逻辑和数据持久化。

4.1 体系结构

4.1.1 程序(模块)划分

本项目典型的 Spring Boot + Vue3 前后端分离项目，于是程序模块的划分，分为以下前端和后端两个模块。

对于后端模块：

层级	模块名称	标识符	功能描述
Controller	BookController	BookController.java	处理书籍相关API请求（如获取书籍的ISBN）
Service	BookService	BookServiceImpl.java	书籍业务逻辑（获取所有书籍）
Mapper	BookMapper	BookMapper.xml	书籍表SQL操作（MyBatis）
Entity	Book	Book.java	书籍表映射的实体
Controller	UserController	UserController.java	处理用户相关API请求（如登录，注册等）
Service	UserService	UserServiceImpl.java	用户业务逻辑（增加新用户等）
Mapper	UserMapper	UserMapper.xml	用户表SQL操作（MyBatis）
Entity	User	User.java	用户表映射的实体

层级	模块名称	标识符	功能描述
Controller	SalesController	SalesController.java	处理售货相关API请求(如接收用户订单等)
Service	SalesService	SalesServiceImpl.java	售货业务逻辑(修改库存等)
Mapper	SalesMapper	SalesMapper.xml	售货单表SQL操作(MyBatis)
Entity	SalesInvoice	SalesInvoice.java	售货单映射的实体
Controller	PurchaseController	PurchaseController.java	处理进货相关API请求(如店家进货等)
Service	PurchaseService	PurchaseServiceImpl.java	进货业务逻辑(修改库存等)
Mapper	PurchaseMapper	PurchaseMapper.xml	进货单表SQL操作(MyBatis)
Entity	PurchaseInvoice	PurchaseInvoice.java	进货单映射的实体
Controller	MessageController	MessageController.java	处理留言板相关API请求(如用户留言等)
Service	MessageService	MessageServiceImpl.java	留言板业务逻辑(店家回复等)
Mapper	MessageMapper	MessageMapper.xml	留言板表SQL操作(MyBatis)

层级	模块名称	标识符	功能描述
Entity	MessageBoard	MessageBoard. java	留言板映射的实体

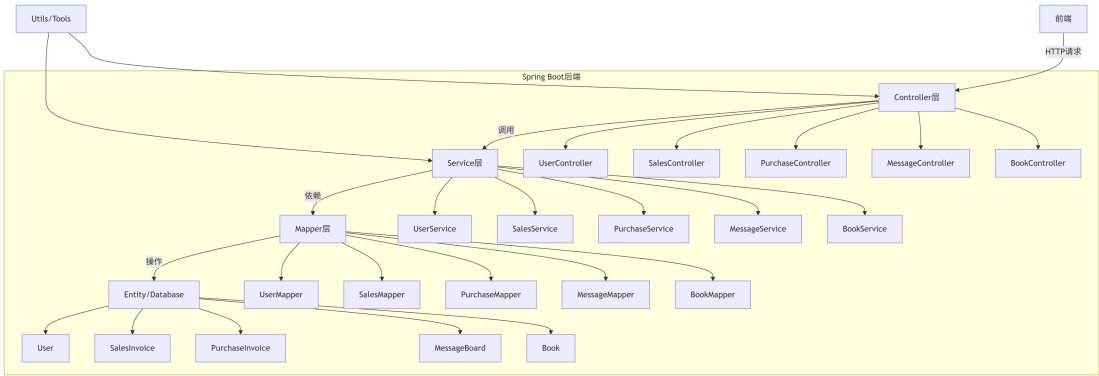
对于前端模块：

模块类型	模块名称	标识符	功能描述	源文件实例
page	About	About. vue	团队相关信息页面	src/pages/About/About. vue
page	Books	Books. vue	展示图书列表、管理员权限验证、添加图书功能	src/pages/Books/Books. vue
page	Home	Home. vue	网上书店首页	src/pages/Home/Home. vue
page	Login	Login. vue	登录页面	src/pages/Login/Login. vue
page	Orders	Orders. vue	订单管理页面	src/pages/Orders/Orders. vue
page	Search	Search. vue	书籍搜索页面	src/pages/Search/Search. vue
page	Statistics	Statistics. vue	图书销售数据统计页面	src/pages/Statistics/Statistics. vue
page	Users	Users. vue	用户管理页面	src/pages/Statistics/Statistics. vue
page	Mys	Mys. vue	个人中心页面	src/pages/Mys/Mys. vue
component	Book	Book. vue	书籍信息展示与管理组件	src/components/Book/Book. vue

component	Navbar	Navbar.vue	响应式导航栏组件	src/components/Navbar/Navbar.vue
component	Order	Order.vue	订单详情展示组件	src/components/Order/Order.vue
component	ShopCart	ShopCart.vue	购物车侧边栏组件	src/components/Shopcart/Shopcart.vue
api	ajax	ajax.js	提供了一个统一的ajax请求函数，支持GET和POST方法	src/api/ajax.js
api	index	index.js	定义并导出了多个与后端API交互的请求函数	src/api/index.js
router	index	index.js	Vue路由配置文件	src/router/index.js
store	modules	Books.js	管理图书数据状态	src/store/modules/Books.js
store	modules	Orders.js	管理订单状态	src/store/modules/Orders.js
store	modules	Person.js	管理用户状态	src/store/modules/Person.js
store	modules	ShopCart.js	管理购物车状态	src/store/modules/ShopCarts.js
store	index	index.js	创建并导出一个Vuex 集中式状态管理实例	src/store/index.js

4.1.2 程序(模块)层次结构关系

系统层次结构如下：



调用关系如下：

调用方	被调用方	场景示例
UserController	UserService	用户注册时调用
SalesController	SalesService	用户下单后修改库存
PurchaseService	PurchaseMapper	店家进货时插入进货单记录
BookService	BookMapper	获取所有书籍列表
MessageService	MessageMapper	店家回复用户留言
SalesService	BookMapper	销售完成后减少书籍库存

4.2 全局数据结构说明

4.2.1 常量

常量名	类型	值	用途	定义位置
API_TIMEOUT_MS	int	30000	API请求超时时间	src/config/constants.js

常量名	类型	值	用途	定义位置
MAX_UPLOAD_SIZE_MB	int	10	页面显示书籍个数	FileUpload.vue
MAX_LOGIN_ATTEMPTS	int	5	最大登录失败次数	UserService.java
MIN_BOOK_STOCK	int	10	库存预警	SalesService.java

4.2.2 变量

变量名	类型	用途	定义位置
currentInventoryStatus	Map<Long, Integer>	缓存当前库存状态(书籍ID → 数量)	PurchaseServiceImpl.java
activeSessions	ConcurrentHashMap<String, LocalDateTime>	记录活跃用户会话	SessionManager.java

4.2.3 数据结构

包括数据结构名称，功能说明，具体数据结构说明(定义、注释、取值…)等。

1、书籍表 book

字段名	类型	描述
name	string	书籍ID
isbn	string	国际标准书号
stock	INT	当前库存量

字段名	类型	描述
author	String	作者
price	Float	价格
version	String	版本
pages	INT	页数

2、用户表 user

字段名	类型	描述
account	String	用户ID
password	String	用户密码
username	String	用户名
isManager	Boolean	是否是管理员
mail	String	邮箱
money	Float	用户剩余金额

3、留言板 MessageBoard

字段名	类型	描述
id	String	用户留言ID
account	String	留言用户账号

字段名	类型	描述
usermessage	String	用户留言内容
managerReply	String	店主回复内容
mail	String	邮箱
money	Float	用户剩余金额

4.3 CSCI 部件

4.3.1 软件配置项

本节对构成网上书店系统（OBS）CSCI 的全部软件配置项（Software Configuration Item, SCI）进行标识、结构说明、用途描述、开发状态说明及资源需求分配，并指出其软件存放位置。编号采用 SCI-XX 格式统一管理。

编号	名称	类型	开发状态	所在模块
SCI-01	UserService	Java类	新开发	用户服务模块
SCI-02	OrderService	Java类	新开发	订单服务模块
SCI-03	InventoryService	Java类	新开发	库存服务模块
SCI-04	PaymentService	Java类	新开发	支付服务模块
SCI-05	BookRepository	数据访问类	新开发	数据访问模块
SCI-06	OrderDatabase	数据库	新开发	订单数据库
SCI-07	NotificationUtil	工具类	重用	通知服务模块

4.3.2 静态结构关系

组成关系，OBS 系统（在线书店系统，假设）的计算机软件组件（CSCI）由多个逻辑模块组成，这些模块进一步包含多个功能类与数据库实体。以下是各模块及其组成部分的具体描述：

用户服务模块，包括 UserService（SCI-01），负责处理用户的注册、登录、信息修改等操作；

用户相关数据访问类，用于执行与用户信息相关的数据库操作；

订单服务模块，包含 OrderService（SCI-02），管理订单的创建、查询和更新等功能；

订单数据库（SCI-06），存储所有订单相关信息，如订单号、订购书籍详情、订单状态等；

库存服务模块，包括 InventoryService（SCI-03），提供对图书库存的操作，如添加新书、更新库存数量等；

图书库存实体类，代表具体的库存记录，包括书籍 ID、库存数量等属性；

支付服务模块，包含 PaymentService（SCI-04），处理支付流程，包括但不限于信用卡支付、在线支付平台集成等；

第三方支付接口类，实现与外部支付网关的交互，确保支付过程的安全性和有效性；

工具模块，包括 NotificationUtil（SCI-07），提供发送通知的服务，例如邮件或短信通知用户订单状态更新等信息。

交互关系，在动态视图中，交互关系可以通过序列图或流程图进行补充说明。这有助于理解不同模块和服务之间如何协作完成特定任务，例如一个完整的购物流程可能涉及到用户服务、订单服务、库存服务以及支付服务等多个模块间的交互。通过序列图可以详细描绘出从用户提交订单到最终支付成功整个过程中涉及的所有步骤和参与对象；而流程图则更适合用来概述业务流程的整体流向，强调各个决策点和循环结构。

4.3.3 配置项用途与需求分配

编号	主要功能/用途	满足的CSCI需求编号
SCI-01	处理用户注册、登录、权限验证	REQ-USR-001--005
SCI-02	管理订单创建、查询、状态变更	REQ-ORD-001--006
SCI-03	维护图书库存、同步外部系统库存	REQ-INV-001--004
SCI-04	集成支付宝/微信支付接口	REQ-PAY-001--004
SCI-05	提供图书与用户的数据访问接口	REQ-DB-001--005
SCI-06	持久化订单数据及交易记录	REQ-DB-010--011
SCI-07	支持邮件、短信通知发送	REQ-NOTI-001--003

4.3.4 开发状态说明

编号	状态类型	来源说明
SCI-01--04	新开发	OBS项目内首次实现
SCI-05	新开发	数据访问层基于JPA封装实现
SCI-06	新开发	MySQL数据库创建新实例
SCI-07	重用	公共开源库

4.3.5 资源使用说明

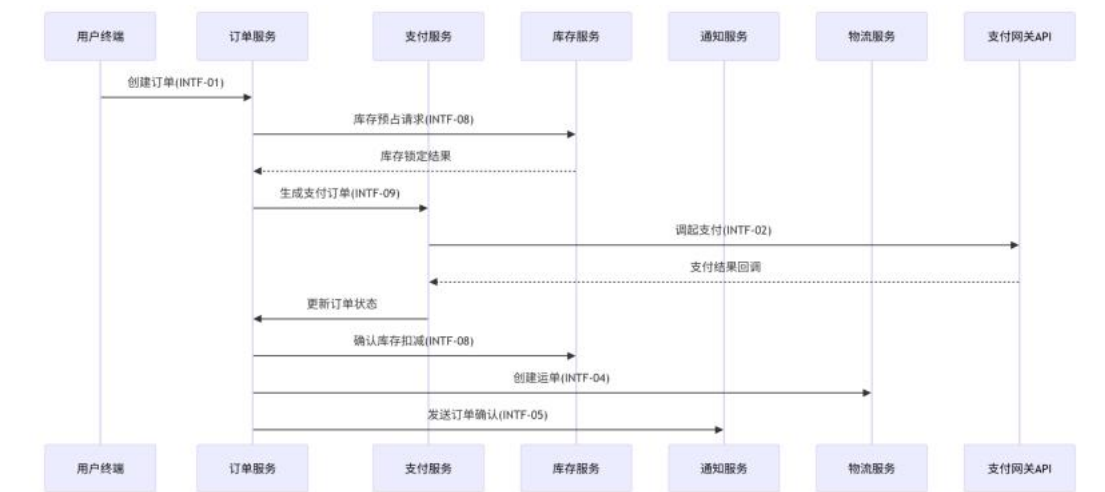
以典型和最坏情况为假设，资源分配如下表所示。

配置项	资源类型	使用情况说明	单位	假设条件
SCI-02	内存	最大使用 100MB，平均 30MB	MB	并发下单峰值
SCI-04	网络带宽	HTTPS通信，每秒最多 100 请求	请求/秒	并发支付
SCI-06	存储	数据库最大存储预计 20GB	GB	保留 6 个月订单数据
SCI-01~05	CPU	平均使用率 10%以下	CPU %	高峰期并发访问
SCI-07	网络IO	调用第三方通知API接口	请求/天	每日通知高峰

4.4 执行概念

4.4.1 执行控制流

CSCI 之间的执行遵循主控-从属模式，主控模块负责调度任务，从属模块执行具体功能。核心交互方法如下图所示。

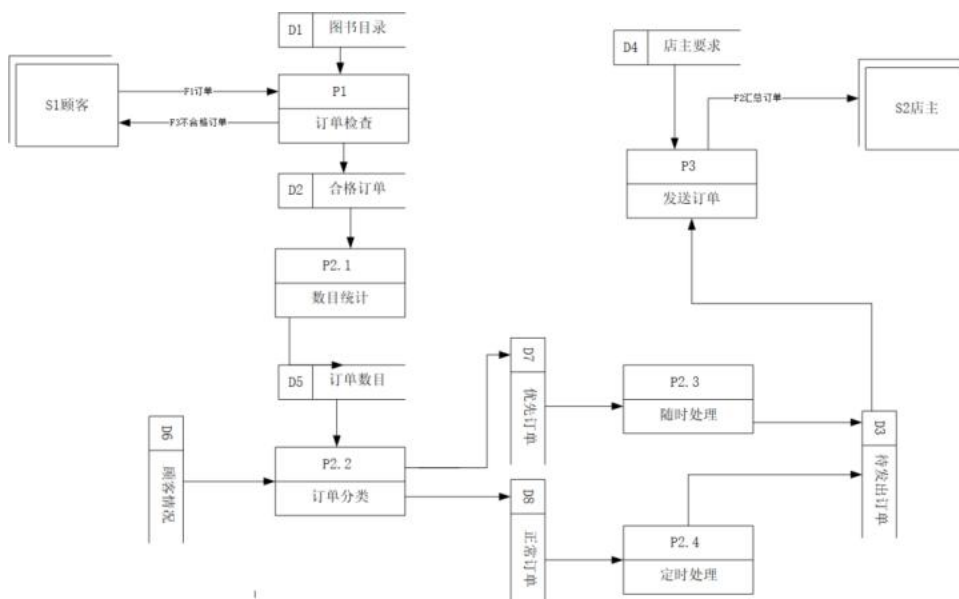


4.4.2 数据流

数据通过消息队列在 CSCI 之间传递，输入数据经处理后输出至下一模块。

示例：数据流图展示数据流向。

本实验采取数据流图来表示本实验的数据流程。



4.4.3 动态控制序列

本系统涉及多种数据流模式，根据业务需求采用不同的传输机制。

高优先级流（关键业务实时处理）：支付状态变更（INTF-09 → INTF-07）：采用同步 RPC 调用，确保支付结果实时生效。设计熔断机制，接口超时（阈值：2s）自动降级，触发补偿流程。

批量处理流（高吞吐量容忍延迟）：库存同步（INTF-03）通过消息队列（Kafka/RabbitMQ）积累批次数据，定时触发批量更新，通过 Saga 模式保证库存与订单状态的最终一致。

实时流（低延迟事件推送）：物流状态推送（INTF-04）基于 WebSocket 长连接实现服务端主动推送，客户端订阅特定订单 ID。

4.4.4 资源管理

系统运行时动态管理关键资源以优化性能。

动态创建，每个订单生成独立事务上下文（TxID），确保支付与库存操作的 ACID 特性。

连接池管理，数据库接口（INTF-06）限制最大连接数（默认 50），避免资源耗尽。

内存分配，支付结果处理采用弹性缓冲区，根据流量动态扩展（初始 1MB，峰值可扩容至 10MB）。

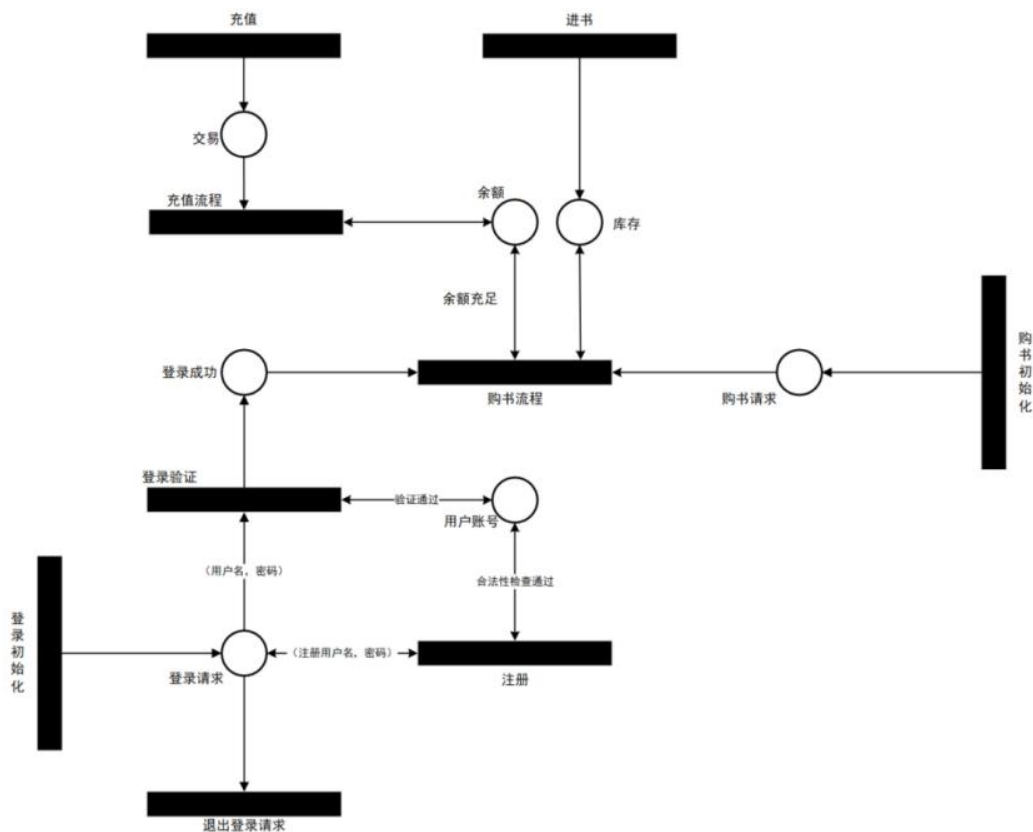
4.4.5 动态特性设计

事务一致性，支付交易与库存扣减通过分布式事务（2PC）保证强一致性。订单服务作为 Saga 协调器，处理异常场景的补偿操作（如支付失败后库存回滚）。异步非阻塞架构，通知服务（INTF-05）采用事件驱动（EventBridge），与核心业务逻辑解耦。

容错机制，采用熔断降级外部接口超时或失败时，自动切换至备用通道或本地模拟；采用重试策略，消息队列消费失败时，按指数退避规则重试（最多 3 次）。

4.4.6 状态转换

状态转换关键路径，本实验处理路程使用 petri 网来表示，如下：



4.4.7 并发执行与异常处理

采用多线程机制，关键资源通过互斥锁（Mutex）保护，避免竞争条件。并发执行，库存预占（INTF-08）与风控检查并行执行，支付结果通知（INTF-02）与物流创建（INTF-04）异步处理，其机制使用代码表示如下。

```

try:
    process_payment(INTF-09)
    update_order_status(INTF-07)
except PaymentTimeout:
    trigger_compensation(INTF-08) # 库存释放
    notify_user(INTF-05)
except InventoryException:
    rollback_order(INTF-07)
  
```

接口状态影响，开发中接口（INTF-03/07/08/09）采用模拟服务处理。检测到异常时，系统记录错误日志并尝试恢复，若失败则进入安全模式。

4.5 接口设计

本章详细描述为实现网上书店系统（OBS）及其各子系统、配置项接口所需的全部需求。本章内容按照接口类型进行分类，所有需求均具备项目唯一标识符，具备可测性和可追踪性。

4.5.1 接口标识

下表标识本系统所有接口，并指明接口实体、版本、状态、约束说明与文档引用。各接口以项目唯一标识符编码：INTF-XX。

编号	接口名称	接口实体	状态
INTF-01	用户界面接口	终端用户	固定
INTF-02	支付系统接口	支付网关API	外部系统
INTF-03	库存同步接口	后端库存服务	正在开发
INTF-04	物流状态通知接口	快递服务商API	外部系统
INTF-05	邮件与短信推送接口	通知服务平台	外部系统
INTF-06	数据库硬件接口	MySQL数据库	固定
INTF-07	用户服务与订单接口	用户服务、订单服务	正在开发
INTF-08	库存服务与订单接口	库存服务、订单服务	正在开发
INTF-09	支付服务与订单接口	支付服务、订单服务	正在开发

4.5.2 INTF-01 用户界面接口

接口类型，图形用户界面（Web）

优先级，高，所有业务功能必须支持终端用户交互，该部分直接决定用户使用体验。

接口函数表格如下。

项目唯一标识符	名称	返回数据类型	长度/格式	约束	来源与接收者
UI-001	获取用户名	String	最大 20 字符	必填，唯一；可更新	用户输入 → 系统验证
UI-002	获取密码	String	hash 后 128 字符	前端加密，后端验证	用户输入 → 系统校验

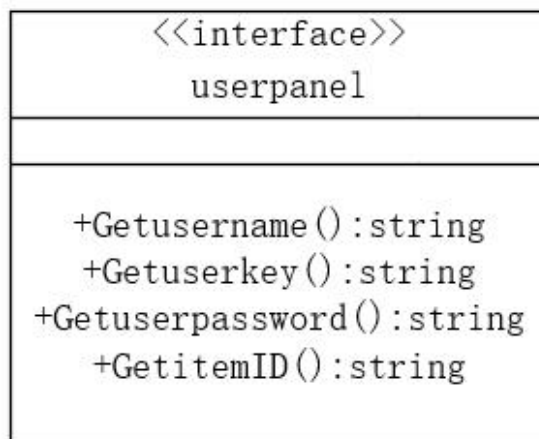
UI-003	获取搜索关键词	String	≤100	可模糊查询,敏感词过滤	用户 → 系统
UI-004	获取商品 ID	String	BK-xxxxxx	不可更改,系统唯一标识符	系统 → 用户界面展示

数据集合体, 页面布局信息 (字体、图标、配色、响应式样式), 登录页面、搜索页、商品详情页、结算页等组件集合。

显示特性, 支持 Chrome、Safari、Edge 浏览器; 移动端自适应布局; 页面加载不超过 1 秒, 按钮响应时间<100ms

安全性, HTTPS 通信; CSRF、XSS 防护; 登录接口限制 5 次错误尝试。

接口图如下:



4.5.3 INTF-02 支付系统接口

接口类型, RESTful API, 基于 OAuth 2.0 鉴权

优先级, 高, 直接关系到账务准确性。

接口函数如下

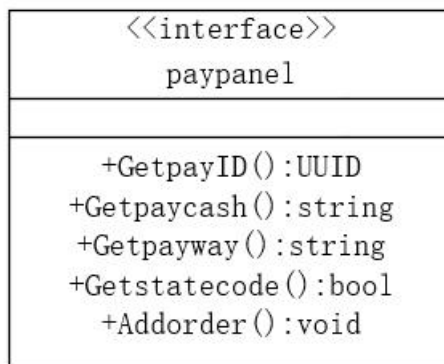
项目唯一标识符	名称	数据类型	范围	精度	来源/接收者
PAY-001	获取订单号	UUID	36 字符	唯一	系统 ↔ 支付平台

PAY-002	获取金额	String	0.01~9999.99	小数点后2位	系统 → 平台
PAY-003	获取渠道	String	ALIPAY, WECHAT	-	用户选择
PAY-004	获取状态码	布尔型	SUCCESS/FAIL	-	平台 → 系统
PAY-005	增加订单	void	-	-	系统→物流

通信特性，HTTPS，最大支持并发 100 qps，响应时间 < 1s，失败重试 3 次，幂等性要求强制执行，消息体 JSON 格式，UTF-8 编码。

协议特性，API 版本控制，签名机制 HMAC-SHA256，状态同步采用轮询+Webhook 双机制。

接口图如下：



4.5.4 INTF-03 库存同步接口

类型，后台 REST 服务接口。

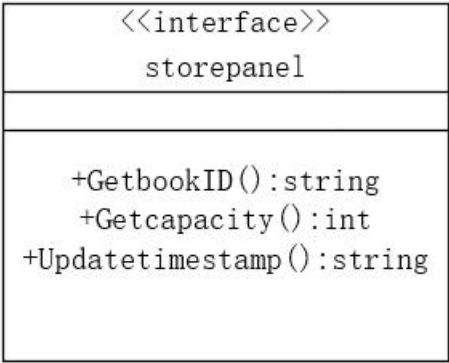
优先级，中，影响库存准确性与下单可用性。

数据结构如表中所示。

项目唯一标识符	字段名称	类型	说明
INV-001	获取图书 ID	String	唯一标识，格式 BK-xxxxxx
INV-002	获取可用库存量	Int	0~9999
INV-003	更新时间戳	String	ISO 8601 格式

通信与协议要求，接口调用方式 POST/PUT/GET，支持增量与全量更新，数据传输加密，访问需 Token 授权，日同步频率大于等于 10 次。

接口图如下：



4.5.5 INTF-04 物流状态通知接口

类型 Webhook 机制，异步消息推送

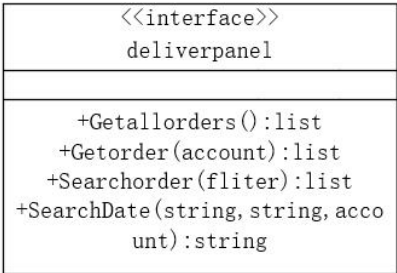
优先级，中，消息体结构

消息体结构，订单号、快递单号、状态枚举（运输中、派送中、签收）

通信特性，POST 方式推送到配置 Webhook URL；超时重试机制（最多 3 次）；传输延迟容忍上限：5 秒内

项目唯一标识符	字段名称	类型	说明
DEL-001	获取所有订单	List	获取所有的订单
DEL-002	获取指定用户订单	List	获取指定用户订单
DEL-003	获取过滤订单	List	获取过滤订单
DEL-004	筛选订单	List	获取特定日期特定用户的订单

接口图如下：



4.5.6 INTF-05 邮件与短信推送接口

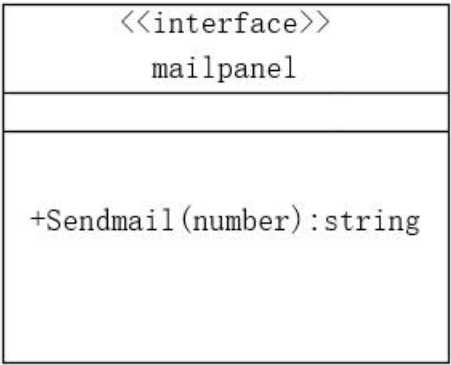
类型，REST API（对接第三方服务，如 SendCloud、阿里云短信）。

数据元素集合体，模板 ID、接收手机号/邮箱、变量参数映射表。

约束，每用户每日限制：10 条短信、20 封邮件；短信字符限制小于等于 70 汉字/条。

项目唯一标识符	字段名称	类型	说明
MAIL-001	发送邮件	String	向特定邮件发送信息

接口图如下：



4.5.7 INTF-06 数据库硬件接口

类型，与 MySQL 8.0 数据库的连接接口

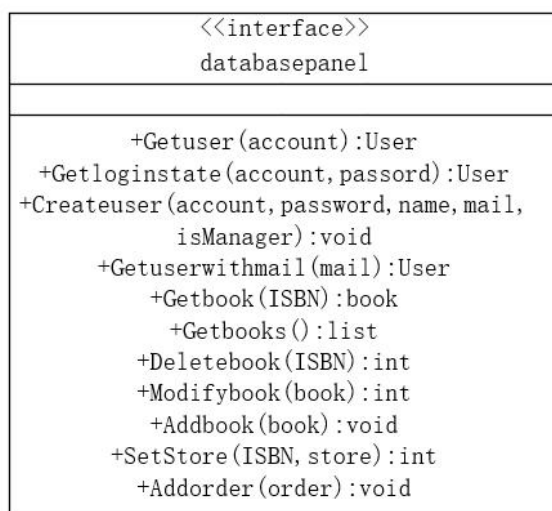
通信方式，内网 TCP 连接（端口 3306）；链接使用连接池，最大并发连接数 200；

安全特性，数据库账户按最小权限原则分配；连接加密（SSL）；访问日志审计每次操作行为。

项目唯一标识符	字段名称	类型	说明
DATA-001	获取指定用户	User	根据账号获取指定用户
DATA-002	获取登录状态	User	根据用户名获取登录状态
DATA-003	创建用户	-	用户注册
DATA-004	根据邮箱获取用户	User	用户邮箱登录
DATA-005	获取图书	Book	根据 ISBN 获取图书

DATA-006	获取所有图书	List	获取所有图书
DATA-007	删除某图书	int	删除某图书
DATA-008	修正图书	int	修正图书信息
DATA-009	增加图书	void	增加某图书
DATA-010	更新库存	int	更新某一图书的库存
DATA-011	增加订单	void	增加一个订单

接口图如下：



4.5.8 INTF-07 用户服务与订单接口

接口用途为验证用户权限并获取用户的订单历史信息，确保只有授权用户才能访问其订单详情。

接口数据格式设计如下，请求（订单服务 → 用户服务）Json 语言表示如下，其中变量含义 user_id 用户唯一标识符；request_type 请求类型，本例中为验证访问权限。

```

{
  "user_id": "U1001",
  "request_type": "verify_access"
}

```

响应 Json 语言表示如下，其中变量含义 is_valid 布尔值，表示用户是否具有访问权限；user_role 用户角色，可用于定制化展示或优惠。

```
{
    "is_valid": true,
    "user_role": "VIP"
}
```

接口性能需要达到调用延迟小于等于 200ms，保证快速响应用户体验，失败重试机制，在网络故障等情况下，自动重试 3 次以提高成功率。

4.5.9 INTF-08 库存服务与订单接口

接口用途为实现订单创建时的实时库存扣减和资源锁定，确保销售的商品库存准确无误。

接口数据格式设计如下，请求（订单服务 → 库存服务）Json 语言表示如下，其中变量含义 book_id 图书唯一标识符；quantity 需要预留或减少的数量；operation 操作类型，如“reserve”表示预留给订单使用。

```
{
    "book_id": "B205",
    "quantity": 2,
    "operation": "reserve"
}
```

响应 Json 语言表示如下，其中变量含义 success 操作是否成功；remaining_stock：当前库存剩余量。

```
{
    "success": true,
    "remaining_stock": 15
}
```

接口需要设计异常处理，当库存不足以满足请求时，返回错误信息 Json 语言表示如下，此响应指示当前操作无法完成，需要用户调整购买数量或选择其他商品。

```
{
```

```
"error": "INSUFFICIENT_STOCK"

}
```

4.5.10 INTF-09 支付服务与订单接口

接口用途为处理来自订单服务的支付请求，并通过回调通知订单状态更新情况，确保交易顺利完成。

接口通信协议设计如下，同步调用，使用 HTTP REST API 进行即时通信，适用于直接获取结果的情况。异步通知，利用 Webhook 机制，在支付完成后主动通知订单服务最新的支付状态，适合于长时间运行的任务。

支付请求 Json 语言表示如下，其中变量含义 order_id 订单编号，用于关联具体的订单记录；amount 支付金额；currency 货币单位，默认为人民币 (CNY)。

```
{

  "order_id": "ORD-20240501-001",

  "amount": 59.99,

  "currency": "CNY"

}
```

支付回调通知 Json 语言表示如下，其中变量含义 order_id 订单编号；status 支付状态，如 "paid" 表示支付成功；transaction_id 第三方支付平台生成的交易 ID，用于追踪和核对。

```
{

  "order_id": "ORD-20240501-001",

  "status": "paid",

  "transaction_id": "TXN-1234567890"

}
```

4.5.11 需求的优先顺序和关键程度

以下接口需求被评定为关键级别：

项目唯一标识符	接口名称	关键性质
INTF-01	用户界面接口	用户体验直接影响留存

INTF-02	支付系统接口	财务安全性要求极高
INTF-06	数据库接口	数据一致性/完整性依赖数据库

除以上关键接口外，其余接口按正常优先级处理。系统整体默认对所有接口施加安全、加密和错误容忍机制要求。

5 CSCI 详细设计

在整个系统中，UserService（SCI-01）是用于处理用户登录、注册和权限验证的核心服务。其实现采用 Spring Boot 框架，并结合 JWT 认证机制实现前后端分离架构下的安全会话管理。密码使用 BCrypt 加密算法进行哈希存储，保障用户敏感信息的机密性。用户接口受限于最大登录失败次数 5 次的安全策略，超限后将自动锁定账户 30 分钟。此外，所有接口调用必须经过统一认证网关验证，未授权请求将被拒绝访问。该模块使用 Java 语言开发，符合 CSCI 整体技术栈一致性。其输入输出包括用户名、密码、邮箱、权限信息等，响应结果为认证状态、会话 Token 或失败提示。在逻辑实现上，该配置项在系统初始化阶段即被加载，并在用户发起身份验证请求时被调用。成功认证后，系统将控制权移交给订单模块，继续后续的操作流程。对于异常情况，例如密码错误、用户不存在等，该模块均提供详尽的错误信息返回，并生成安全审计日志，供后期追踪与分析使用。

OrderService（SCI-02）作为用户购物流程的核心处理模块，承担订单的创建、查询、状态更新等职责。设计上，系统支持 15 分钟订单未支付自动取消机制，以释放库存资源，防止资源浪费。该模块通过 REST 接口与 InventoryService 和 PaymentService 模块进行同步与异步交互，使用 JPA 访问订单数据库，确保事务性与一致性。其数据输入主要包括书籍 ID、购买数量和用户信息，输出则包括订单编号、订单状态、支付接口信息等。在操作流程上，用户下单请求到达后，系统首先校验订单合法性，并在本地创建订单草稿，随后同步请求库存扣减，最后生成支付请求交由支付服务处理。整个操作过程响应时间设计控制在 300ms 以内，以保障良好的用户体验。当发生支付异常或库存不足等情况时，系统自动触发补偿机制，进行订单状态回滚和库存恢复处理，确保系统数据一致性。

InventoryService（SCI-03）负责图书库存的管理与维护，其设计支持高频读取与中低频写入的操作模式，采用缓存机制提升读取效率。库存数据的实时更新通过 Kafka 消息队列触发批量处理，并在高优先级请求中支持同步处理模式，以满足订单扣减等对一致性要求较高的场景。系统每日至少进行 10 次库存数据同步，所有库存数量低于阈值（默认 10 本）时会触发预警通知。库存服务接收来自订单服务的图书 ID 和操作数量请求，处理后返回是否成功以及剩余库存量。库存接口（参见 INTF-03 与 INTF-08）支持“reserve”、“release”等多种操作类型，以适配不同的业务场景。在设计逻辑上，系统接收到扣减请求后，会首先判断库存是否充足，若充足则进行数量变更并写入数据库；若不足则返回 INSUFFICIENT_STOCK 错误信息。对于并发场景，系统通过锁机制或原子操作保障库存数据不被竞争修改，确保准确性。

PaymentService (SCI-04) 用于处理订单的支付流程，并与第三方支付平台（如支付宝、微信）集成。其设计采用 OAuth 2.0 进行接口鉴权，数据传输采用 HTTPS 协议并支持签名校验机制（HMAC-SHA256），保障交易过程的完整性与安全性。支付模块既支持同步请求也支持 Webhook 形式的异步通知，以适应不同场景的需求。在发起支付时，系统通过接口向第三方平台发送订单编号、支付金额及支付渠道等数据，并记录返回的支付请求状态。支付成功后，第三方系统通过回调接口通知平台支付结果，系统再根据该结果更新订单状态，并触发后续的发货流程。整个支付过程必须在 1 秒内完成初始响应，并保证幂等性，即重复请求不应导致重复扣款或生成多个支付记录。对于支付超时、支付失败或中断的情况，系统将记录详细的错误日志并通过 NotificationUtil 发送提示信息通知用户。

BookRepository (SCI-05) 作为数据访问模块，封装了对图书数据库的所有增删查改操作。其底层使用 JPA 技术自动生成 SQL 语句，支持分页查询、模糊搜索以及按分类筛选等功能。该模块尤其在系统前台书籍展示、后台图书管理以及搜索功能中承担关键角色。输入数据包括关键词、ISBN 等，输出为图书对象列表，支持高并发只读访问。模块设计上，热书数据会被缓存至 Redis 中，减少数据库读取压力。该模块响应速度设计为查询在 300ms 内返回，并对无数据情况提供空结果提示；若数据库连接异常，则返回错误码并上报系统管理员。

OrderDatabase (SCI-06) 是系统中所有订单数据的存储载体，采用 MySQL 8.0 作为支撑数据库，使用 InnoDB 引擎以支持高并发事务处理。数据库表结构支持 UUID 主键、状态字段、时间戳记录，并为支付、取消、发货等各类状态设置明确标识。该数据库与 OrderService 进行高频数据交互，存储操作采用事务机制，确保订单数据的原子性、一致性、隔离性和持久性（ACID）。数据表还支持软删除功能，以便于后期审计和恢复。在数据更新过程中，若遇数据库主键冲突或约束校验失败，系统将中断当前操作并回滚所有变更，避免脏数据写入。

最后，NotificationUtil (SCI-07) 为整个系统提供统一的消息通知能力，负责将系统事件以短信或邮件形式传达给用户。该模块通过 REST API 对接 SendCloud 或阿里云短信平台，支持模板化发送，内容动态填充。每个用户每日短信与邮件发送量受限于 10 条和 20 封，以防止滥发。该模块采用异步调用机制，不阻塞主业务流程；若发生发送失败，会进行 3 次指数退避重试，仍失败则记录至系统日志供后续查看。通知服务的触发通常由订单状态变更、支付成功、库存预警等事件驱动，具有良好的可扩展性与解耦性。

6 需求的可追踪性

6.1 从软件配置项到 CSCI 需求的可追踪性

SCI-01 UserService:

UserService 模块承担用户注册、登录、权限验证与个人信息管理等核心业务功能，涵盖了前端用户交互与后端身份校验的完整流程。在需求分配阶段，本

配置项对应的 CSCI 需求编号集中在 REQ-USR-001 到 REQ-USR-005, 这五个需求编号具体描述了用户相关的功能性需求, 例如用户注册信息校验、登录状态维护、密码加密存储与验证、用户权限角色分配, 以及用户基本资料修改与查询等。通过将这五项需求统一分配给 SCI-01, 设计团队保证了身份认证与用户管理逻辑的一致性和完整性。

SCI-02 OrderService:

OrderService 模块负责全流程的订单管理, 包括订单创建、查询、更新与状态维护等功能, 用以支撑购物流程中的订单生成与后续状态演进。在需求分配表中, SCI-02 对应的需求编号为 REQ-ORD-001 到 REQ-ORD-006。这六项需求涵盖了从下单时的商品选取、价格计算、订单号生成到支付后状态变更, 并最终提供订单历史记录查询的所有功能性需求。通过赋予 OrderService 这六项需求, 团队确保系统能够符合业务对订单生命周期的完整支撑。

SCI-03 InventoryService:

InventoryService 模块主要负责图书库存的维护与同步, 包括库存扣减、库存回补、外部库存同步以及库存预警等功能。在需求分配中, SCI-03 对应 REQ-INV-001 到 REQ-INV-004。具体而言, 这四项需求涉及库存初始录入与维护、实时库存扣减逻辑、库存不足时的告警机制, 以及与外部供应链系统对接获取增量库存更新。通过将这四项需求集中到 InventoryService, 使得系统能够在订单创建或退货时实时更新库存, 并保证库存数据的准确性与一致性。

SCI-04 PaymentService:

PaymentService 模块承担所有与支付相关的功能, 包括调用支付宝、微信等第三方支付网关、处理支付回调, 以及在支付失败时进行异常处理与多渠道重试等。在需求分配表中, SCI-04 对应 REQ-PAY-001 到 REQ-PAY-004。这四项需求覆盖了支付订单号生成、支付金额校验、支付渠道选择与调用, 以及支付结果状态同步与回调通知。配置 PaymentService 满足了系统对“可靠、安全、高可用”支付流程的设计要求。

SCI-05 BookRepository:

BookRepository 作为数据访问层中的核心组件, 负责所有与图书和用户信息相关的数据持久化操作, 包括增删改查、分页查询与模糊匹配等功能。在需求分配表中, SCI-05 对应 REQ-DB-001 到 REQ-DB-005。这五项需求涉及图书表与用户表的基础增删改查、基于 ISBN 或用户账号的条件查询、以及数据库访问效率等非功能性需求, 通过将这些需求分配给 BookRepository, 保证了数据访问层的统一性与可维护性。

SCI-06 OrderDatabase:

OrderDatabase 即系统中专门用于存储订单信息的 MySQL 实例, 负责创建订

单表、持久化订单与交易记录，并提供事务回滚与恢复机制等。在需求分配中，SCI-06 对应 REQ-DB-010 和 REQ-DB-011。这两项需求分别描述了订单数据持久化要求与订单表结构设计约束（如字段规范、索引策略等），通过明确将它们分配给 OrderDatabase，确保了数据库层在存储性能与数据一致性方面满足系统需求。

SCI-07 NotificationUtil:

NotificationUtil 模块封装了邮件与短信通知服务，提供异步发送与模板动态渲染功能，用以在订单状态更新、库存预警或用户活动时向用户或管理员发送告警与提示信息。依据需求分配表，SCI-07 对应 REQ-NOTI-001 到 REQ-NOTI-003。这三项需求具体说明了通知触发条件、消息模板管理与发送限流策略，通过将这些需求赋予 NotificationUtil，系统能够灵活地对外发送通知并保障日均通知量的可控性。

6.2 从 CSCI 需求到软件配置项的可追踪性

本节从需求角度出发，按照需求编号对所有 CSCI 需求所分配的 SCI 进行汇总与说明。若后续某项需求发生变更，可通过本节快速定位受影响的配置项，便于进行相应修改与回归测试。

REQ-USR-001 至 REQ-USR-005（用户相关功能需求）：

这五项需求全部由 SCI-01 UserService 负责实现。具体而言，当需求文档中对用户注册、登录、权限校验、用户角色分配、用户资料管理等功能进行定义时，对应的实现逻辑均在 UserService 模块中完成，包括与数据库交互、JWT 令牌生成与刷新机制、权限判断与会话管理等操作。因此，在任何用户模块相关需求发生变更（如新增密码复杂度校验），都应首先定位到 SCI-01 进行相应调整。

REQ-ORD-001 至 REQ-ORD-006（订单管理相关功能需求）：

OrderService（SCI-02）承担所有与订单创建、查询、更新与生命周期管理相关的需求。无论是业务需求层面对订单号生成规则的修改、订单状态流转逻辑的调整，还是对历史订单查询条件的扩展，都需要在 SCI-02 中进行代码更新。因此，SCI-02 需要保留良好的可追溯记录，以便随时对应到需求文档中的具体 REX-ORD-序号。

REQ-INV-001 至 REQ-INV-004（库存管理相关需求）：

InventoryService（SCI-03）是库存逻辑的唯一承担者。所有与库存扣减、库存回补、库存同步与预警机制有关的需求，都在该配置项中得到实现与验证。当业务需求定义新的库存阈值计算方式或引入新的外部库存同步接口时，都会在 SCI-03 中同步更新。

REQ-PAY-001 至 REQ-PAY-004（支付管理相关需求）：

PaymentService（SCI-04）实现了系统与支付宝、微信等第三方支付网关的集成接口、支付回调的处理流程以及支付异常重试。任何支付流程或安全策略（如回调验签方式、支付超时处理时限）的需求变更，都应映射到 SCI-04 的对应代码或配置中。

REQ-DB-001 至 REQ-DB-005（基础数据访问需求）：

BookRepository（SCI-05）负责图书与用户数据访问逻辑，包括 DAO 层的增删改查、事务管理与分页查询。对数据库访问效率或索引策略等需求的变更，需要在 SCI-05 中进行针对性的改造与优化。

REQ-DB-010 至 REQ-DB-011（订单数据库存储需求）：

OrderDatabase（SCI-06）负责数据库层面的表结构设计与存储过程编写，包括订单表与交易表的字段定义、索引建立与分区策略等。若需求在订单表字段或存储方式上发生更改，需直接修改 SCI-06 相关脚本或模式定义。

REQ-NOTI-001 至 REQ-NOTI-003（通知服务相关需求）：

NotificationUtil（SCI-07）针对异步邮件与短信推送机制进行了封装，包括模板管理与限流策略。当通知业务场景或第三方服务变更（如引入新的短信厂商或调整每日发送上限）时，需要在 SCI-07 中增删相应的模板与发送策略。