图书管理系统项目文档

# 一、项目概述

该系统为一个多角色图书管理平台，支持借阅者端（UserSide）、员工端（EmployeeSide）和 Web 前端接入服务端（ServerSide），实现注册、借还书、图书管理、员工管理等功能。

# 二、系统架构总览（双通路架构）

Web端浏览器 用户/员工客户端  
 ↓ ↓  
 [Node 网关] [Java Socket 客户端]  
 ↓ ↓  
 HTTP/JSON Socket/JSON  
 ↓ ↓  
 [📡 ServerMain 服务端主线程]  
 ↓  
 RequestDispatcher  
 ↓  
 ┌────┬──────┬──────┬──────┐  
 │注册│图书管理│借还书│员工管理│  
 └────┴──────┴──────┴──────┘  
 ↓  
 MySQL + Druid连接池

# 三、模块功能划分

1. ServerSide 服务端  
- networkCommunicationService  
 - ServerMain.java: 启动Socket服务端  
 - RequestDispatcher.java: 分发请求到 service  
- tableService: 注册、借阅校验、库存变更、员工控制等  
- dao: BasicDAO 通用封装，支持事务/查询等  
- domain: 借阅者、员工、图书等实体类  
- protocol: Request、Response、OperationType  
  
2. UserSide / EmployeeSide  
- 使用 Socket 向 ServerMain 发送 JSON Request  
- 使用 JUnit 测试类模拟用户行为  
  
3. Web端（Node桥接 + HTML）  
- index.js：Node.js HTTP → Java Socket 转发  
- index.html：提供表单提交，fetch /api/socket 接口

# 四、数据通信格式（JSON）

// 请求 Request  
{  
 "operationType": "REGISTER",  
 "payload": {  
 "borrowerNumber": "202512345678",  
 "name": "测试用户",  
 "gender": "男",  
 "age": 21,  
 "registerDate": "2025-06-05",  
 "status": "正常"  
 }  
}  
// 响应 Response  
{  
 "success": true,  
 "message": "注册成功",  
 "data": null  
}

# 五、并发控制策略

| 场景 | 是否加锁 | 方案 |  
|--------------------|-----------|-------------------------------------------|  
| 同时借同一本书 | ✅ 是 | MySQL事务 + UPDATE 条件判断库存 |  
| 员工信息更新 | ✅ 是 | 乐观锁 version 字段 |  
| 用户注册编号冲突 | ✅ 是 | 唯一索引 + try-catch 捕获冲突 |  
| 查询图书列表 | ❌ 否 | 纯读操作无需并发控制 |

# 六、关键技术栈

| 层级 | 技术 |  
|------------|------------------------------------|  
| 服务端 | Java 15 / Socket / Gson / JDBC |  
| 数据库连接 | Druid 连接池 |  
| Web 接口 | Node.js / Express |  
| 前端 | 原生 HTML + JS |  
| 测试 | JUnit 4 |

# 七、项目启动顺序

1. 启动 ServerMain（监听9999）  
2. 启动 index.js（Node桥接服务，监听3000）  
3. 打开 index.html 或运行客户端测试类 @Test

# 八、补充说明

- 所有通信统一使用 JSON 格式，接口规范统一  
- RequestDispatcher 中用 gson.toJson + fromJson 规避类型丢失  
- 双通路架构既支持客户端 Socket，又支持 Web HTTP 请求  
  
如需与前端同学联调，只需确保 Node 服务与 Java 服务均启动。