

# WEB工程阶段性报告

## 第一组

2025 春

汇报人：张讯启  
小组成员：郑 好  
张讯启  
王嘉亿  
任小珂  
陈宇航  
孙朝冉

# 目录

01

项目需求引入

02

Web应用建模  
(UWE方法)

03

Web应用架构设计

04

Web应用设计

05

总结与展望

# PART 01

## 项目介绍



# 项目背景与目标

## 项目背景

传统图书管理效率低下，手工登记易出错，借阅流程繁琐，需数字化升级。

图书馆资源利用率低，缺乏有效信息管理系统，影响读者体验。

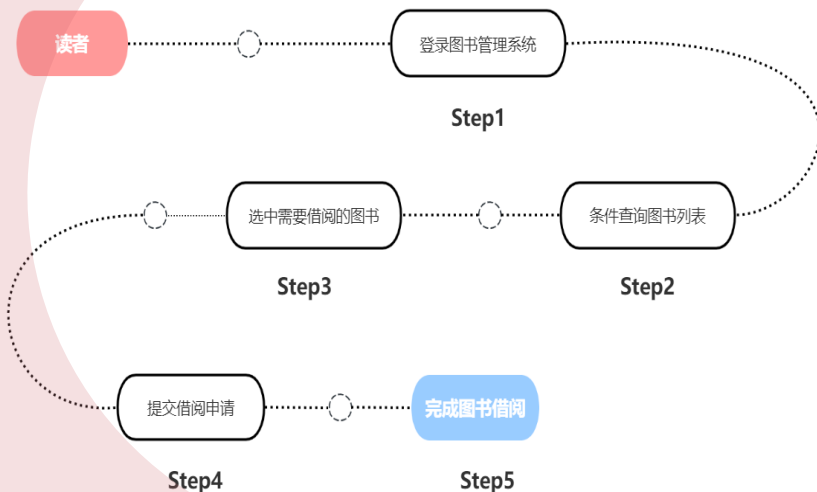
## 项目目标

实现图书分类、借阅、用户管理，提升管理效率和用户体验。

提供高效、安全的在线借阅体验，支持多终端访问。

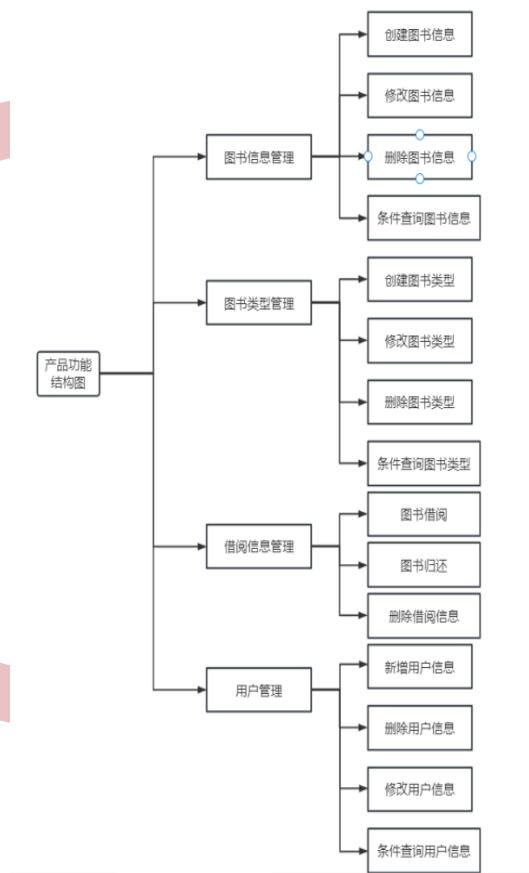
支持前后端分离开发，确保系统可扩展性和维护性。

产品总体业务流程图



# 核心功能需求

## 产品功能结构



### 01 用户管理

注册/登录/密码修改，支持邮箱或手机号注册，提供找回密码功能。

用户信息管理，包括个人信息查看、编辑和权限设置。

### 02 图书信息管理

增删改查图书信息，支持多字段搜索和条件分页查询。

图书分类管理，自定义分类体系，方便图书检索。

### 03 借阅管理

借阅/归还图书，实时更新库存状态，记录借阅历史。

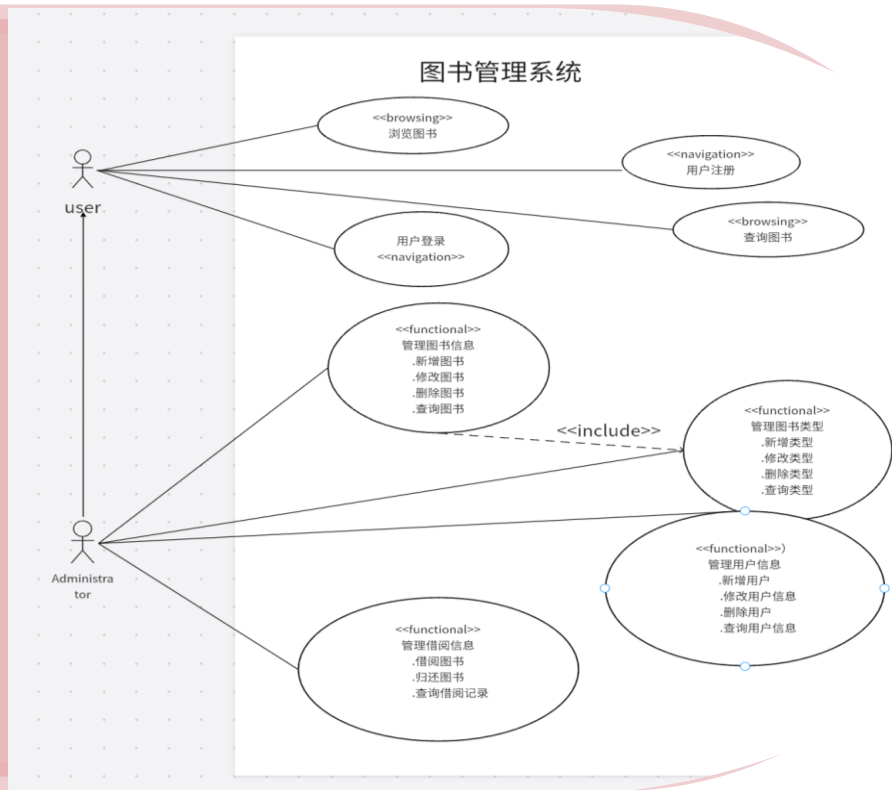
借阅记录删除，管理员可对异常记录进行清理。

## PART 02

# Web应用建模（UWE方法）



# 功能需求建模



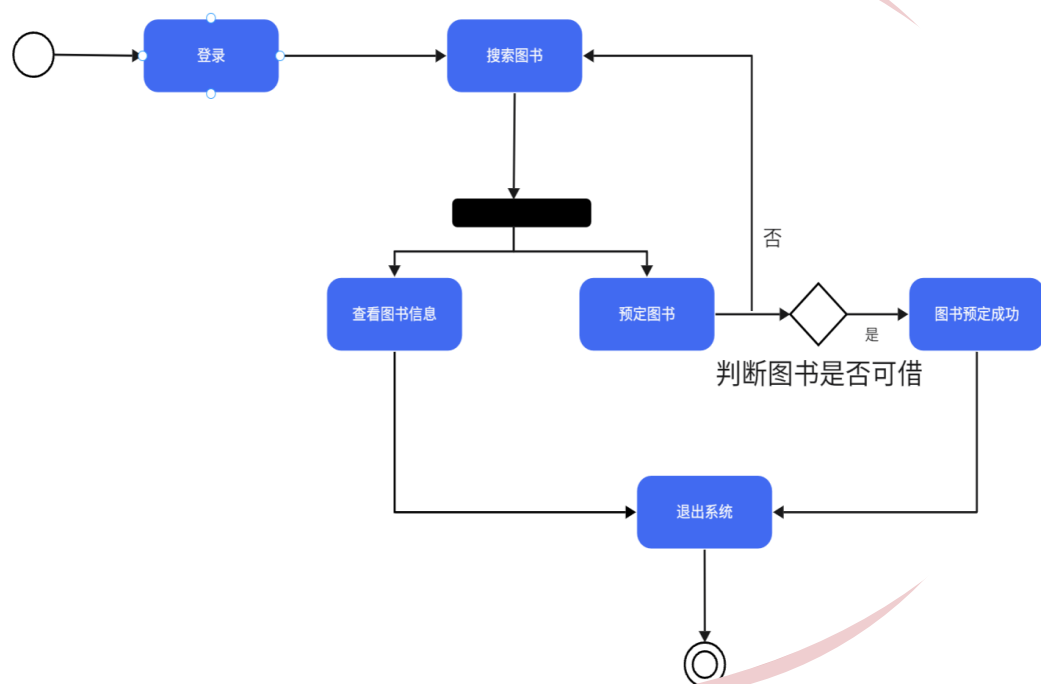
## 用例图

参与者：管理员、读者，明确各自权限和操作范围。

核心用例：图书管理、借阅操作，覆盖系统主要功能。

# 功能需求建模

用户借书活动图



## 活动图

借阅流程：查询→申请→审批→记录，清晰展示借阅步骤。

还书流程：提交申请→审核→更新库存，规范还书操作。



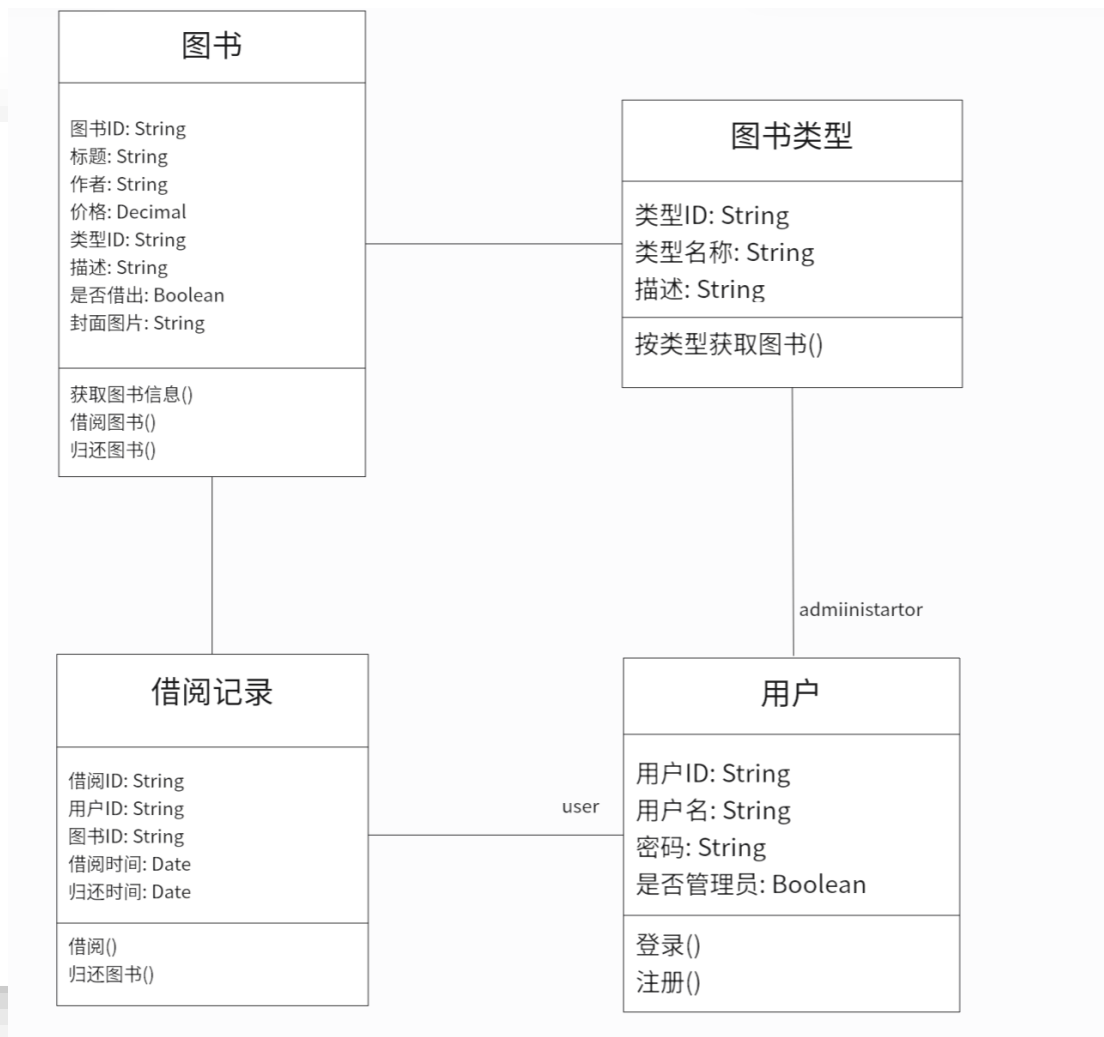
# 内容建模

## 01.

### 静态模型（类图）

关键实体：Book（图书）、User（用户）、Borrow（借阅记录）。

关系：借阅记录关联用户与图书，体现多对多关系。

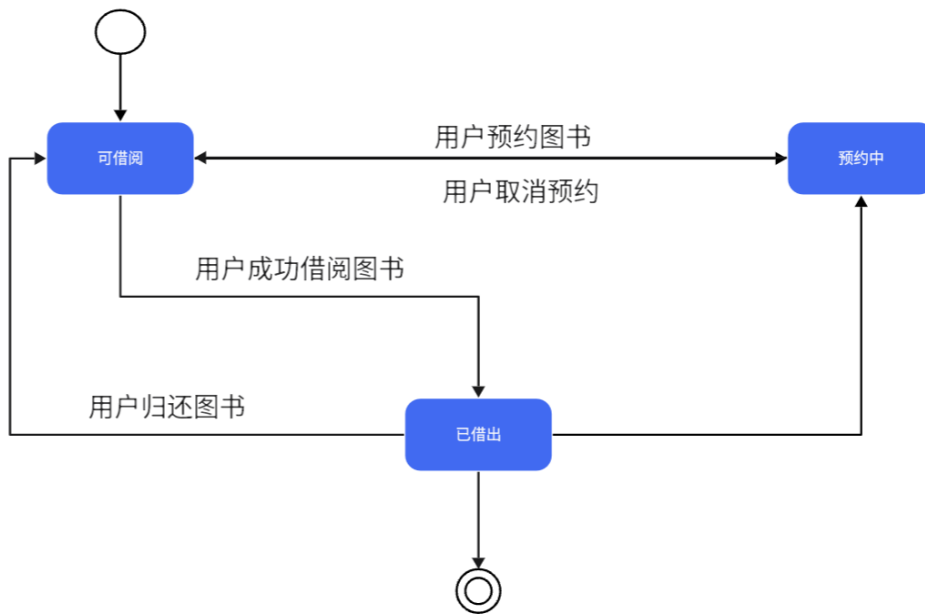


# 内容建模

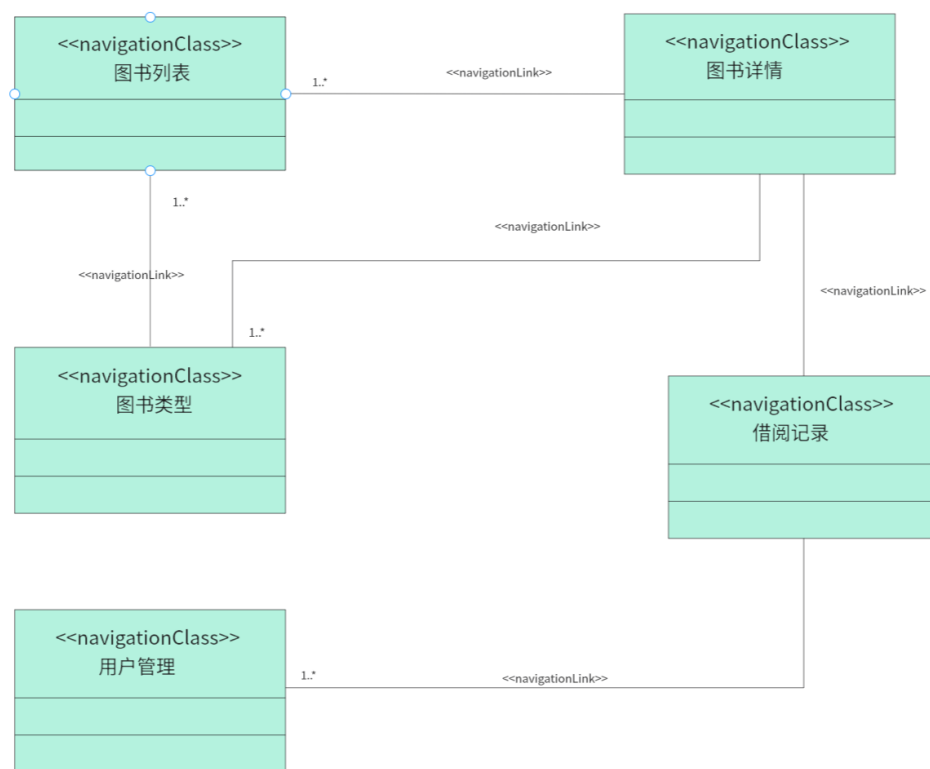
## 02.

### 动态模型（序列图）

交互示例：读者借阅图书时的系统调用链，展示前后端交互过程。



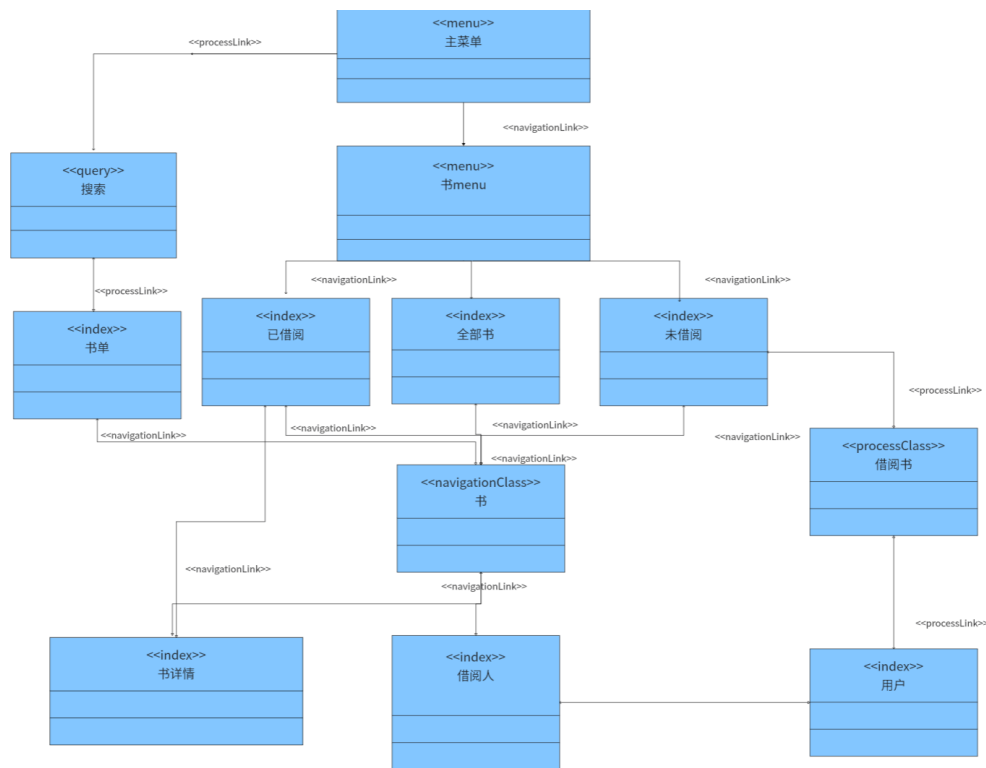
# 超文本建模



## 静态导航结构

页面层级：首页→图书列表→借阅表单，合理规划用户浏览路径。

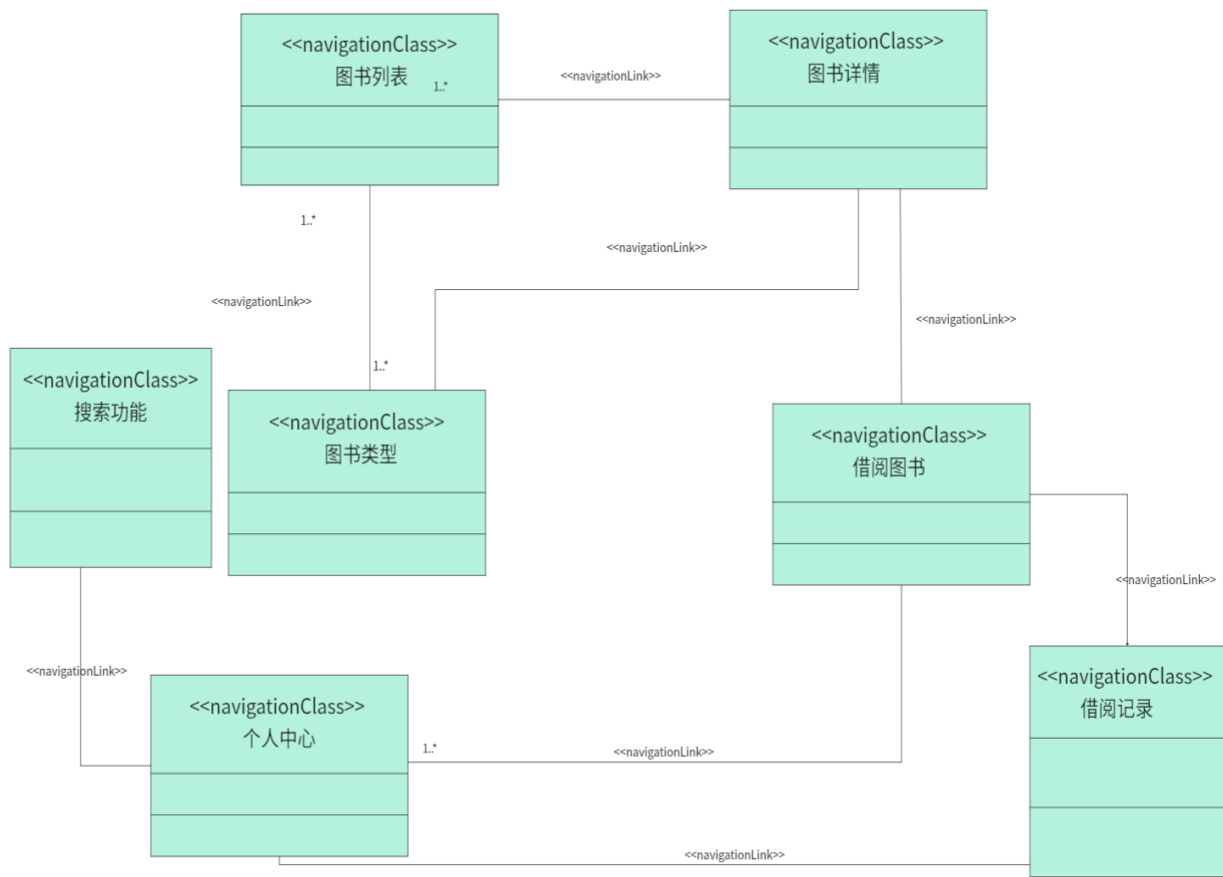
# 超文本建模



## 动态行为

页面跳转逻辑：登录→权限校验→功能页，根据用户角色展示不同界面。

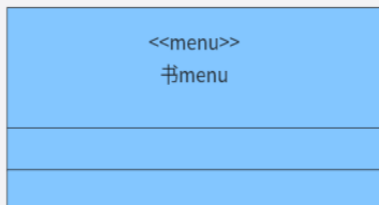
# 适应性建模



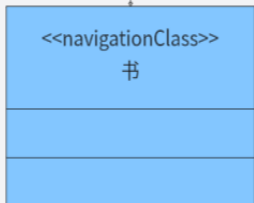
## 01. / 个性化适配

角色区分：管理员与读者界面差异化，满足不同用户需求。

# 适应性建模



用户习惯借阅书类型的所有书



## 02. / 动态模型示例

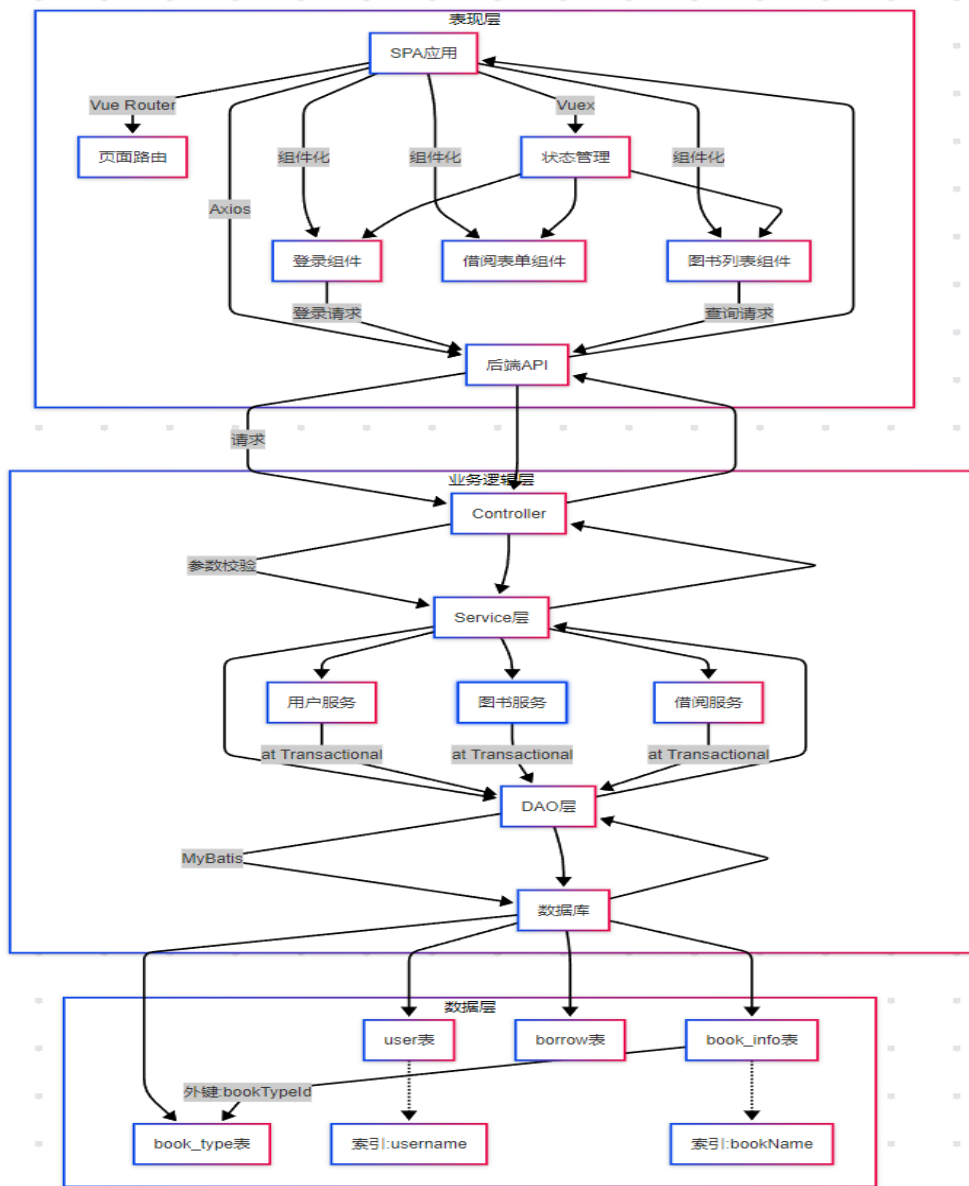
当用户搜索图书时，系统会分析其历史借阅记录，优先推荐与其偏好匹配的书籍类型（如文学、科技等），从而提升搜索效率和个性化体验。

## PART 03

# Web应用架构设计



# 分层架构



## 前端层 (Vue.js)

组件化开发，提高代码复用性和可维护性。  
Vuex状态管理，实现全局状态共享。

## 后端层 (Spring Boot)

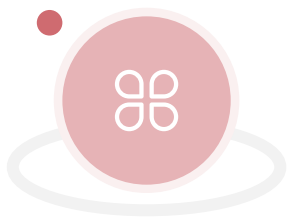
控制器→服务层→DAO层，清晰分层架构。  
模块化：用户/图书/借阅服务，便于后续扩展和维护。

## 数据层 (MySQL)

三范式设计，确保数据一致性和完整性。  
索引优化，提高查询效率。

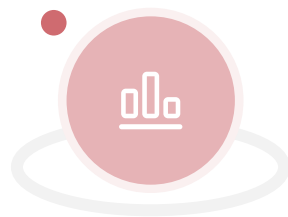


# »»» 关键设计原则



## 接口设计

RESTful API，遵循标准规范，便于前后端分离开发。  
JWT认证，保障用户信息安全，实现无状态认证。



## 服务拆分

高内聚低耦合，支持未来微服务化改造。  
按业务模块拆分，便于团队协作和代码管理。



## 数据一致性

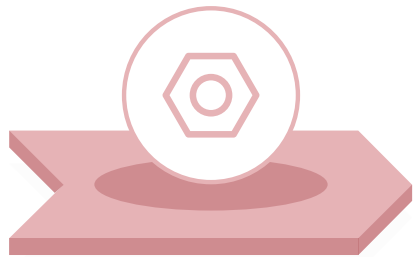
Spring事务管理，确保借阅操作等关键业务数据一致性。  
数据库事务隔离级别设置，防止并发问题。

# PART 04

## Web应用设计



# »»» 前后端协作策略



## 接口优先

---

早期定义API规范，Mock Server支持并行开发。  
使用Swagger等工具生成接口文档，方便前后端沟通。

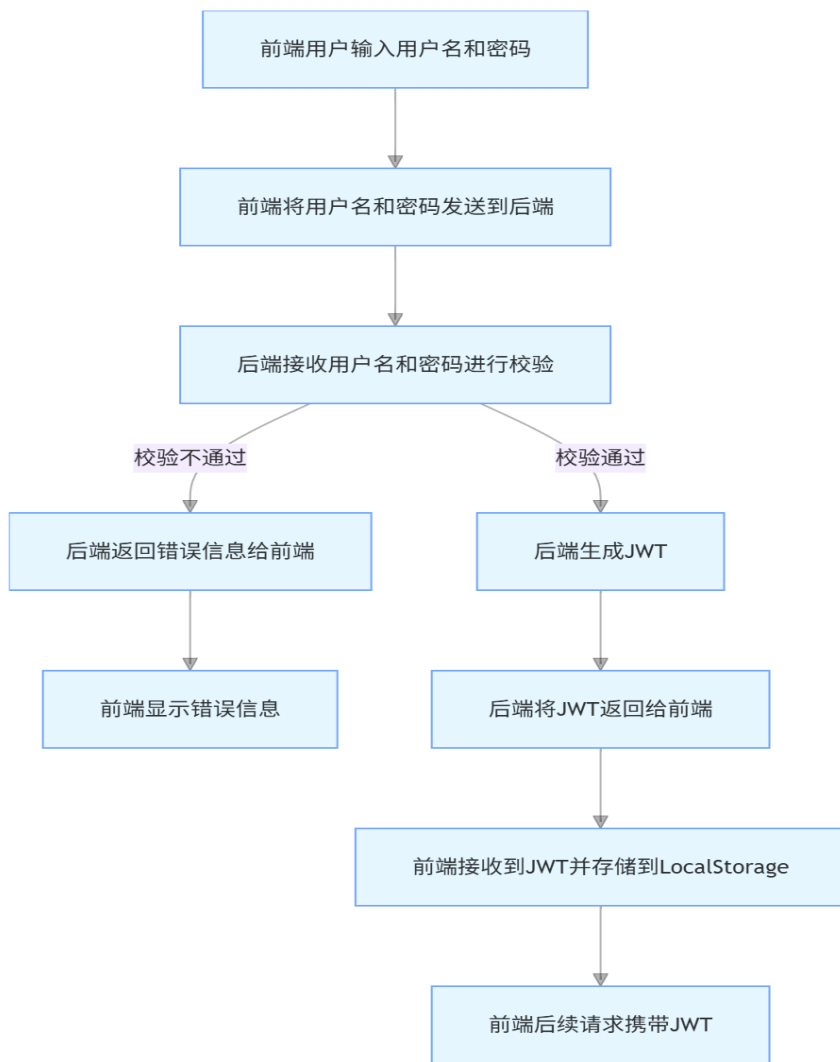


## 模块化设计

---

前端：Vue组件（如BookList.vue），封装独立功能模块。  
后端：Service层封装业务逻辑，便于复用和扩展。

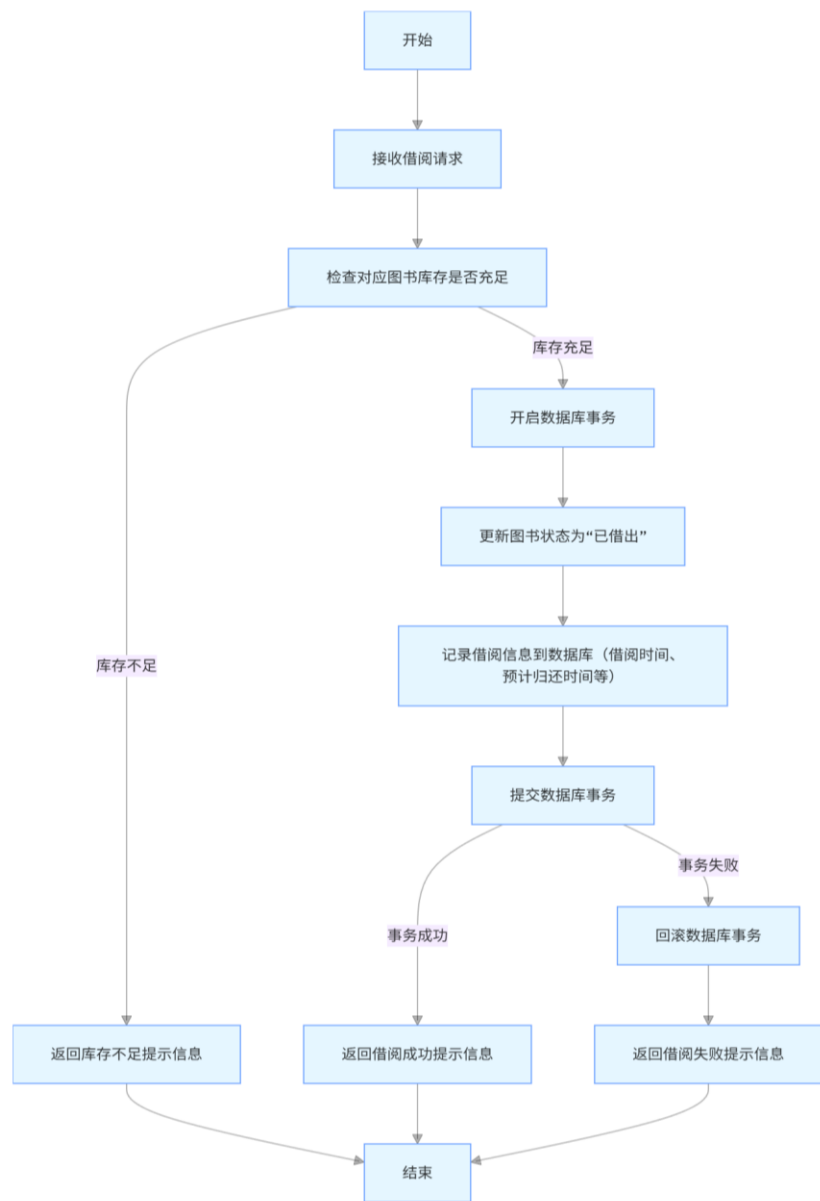
# 核心流程示例



## 用户登录

前端提交用户名和密码，  
后端进行校验。  
校验通过后生成JWT，存  
储至LocalStorage，实现  
免登录状态。

# 核心流程示例



## 图书借阅



事务管理：检查库存是否充足，更新图书状态为“已借出”。记录借阅信息到数据库，包括借阅时间、预计归还时间等。

# PART 05

## 总结与展望



# 当前成果

README	 
TASK 1	
已创建仓库并加入小组成员	
TASK 2	
已确定小组选题：图书馆管理系统。 项目介绍文档撰写如下：	
项目介绍：【腾讯文档】项目介绍 <a href="https://docs.qq.com/doc/DWEF0S1hpWk55bnZv">https://docs.qq.com/doc/DWEF0S1hpWk55bnZv</a>	
TASK 3	
已完成项目的需求描述撰写：	
需求描述：【腾讯文档】需求描述 <a href="https://docs.qq.com/doc/DWGNtbm50S3pnVUd3">https://docs.qq.com/doc/DWGNtbm50S3pnVUd3</a>	
TASK 4	
已完成项目的建模文档撰写：	
【腾讯文档】Web应用建模 <a href="https://docs.qq.com/doc/DWE5aYUV6bkRCcGx5">https://docs.qq.com/doc/DWE5aYUV6bkRCcGx5</a>	
TASK 5	
确立了前后端并行开发，面向接口和服务的架构，具体实现请参见master分支和fore-end分支，完成了架构设计报告	
TASK 6	
完成了Web应用的设计	

完成需求分析，明确项目目标和核心功能需求。

完成建模工作，包括功能需求建模、内容建模、超文本建模和适应性建模。

完成架构设计，确定分层架构和关键设计原则。

# »»» 后续计划

01

进行微服务化改造，使用Spring Cloud实现服务拆分和分布式管理。

02

引入Redis缓存优化，提高系统性能，减少数据库压力。

03

实现容器化部署，使用Docker和Kubernetes进行应用部署和管理。



# 谢谢大家！

---

Web工程第一组