# **《图书管理系统概要设计》**

## **一、引言**

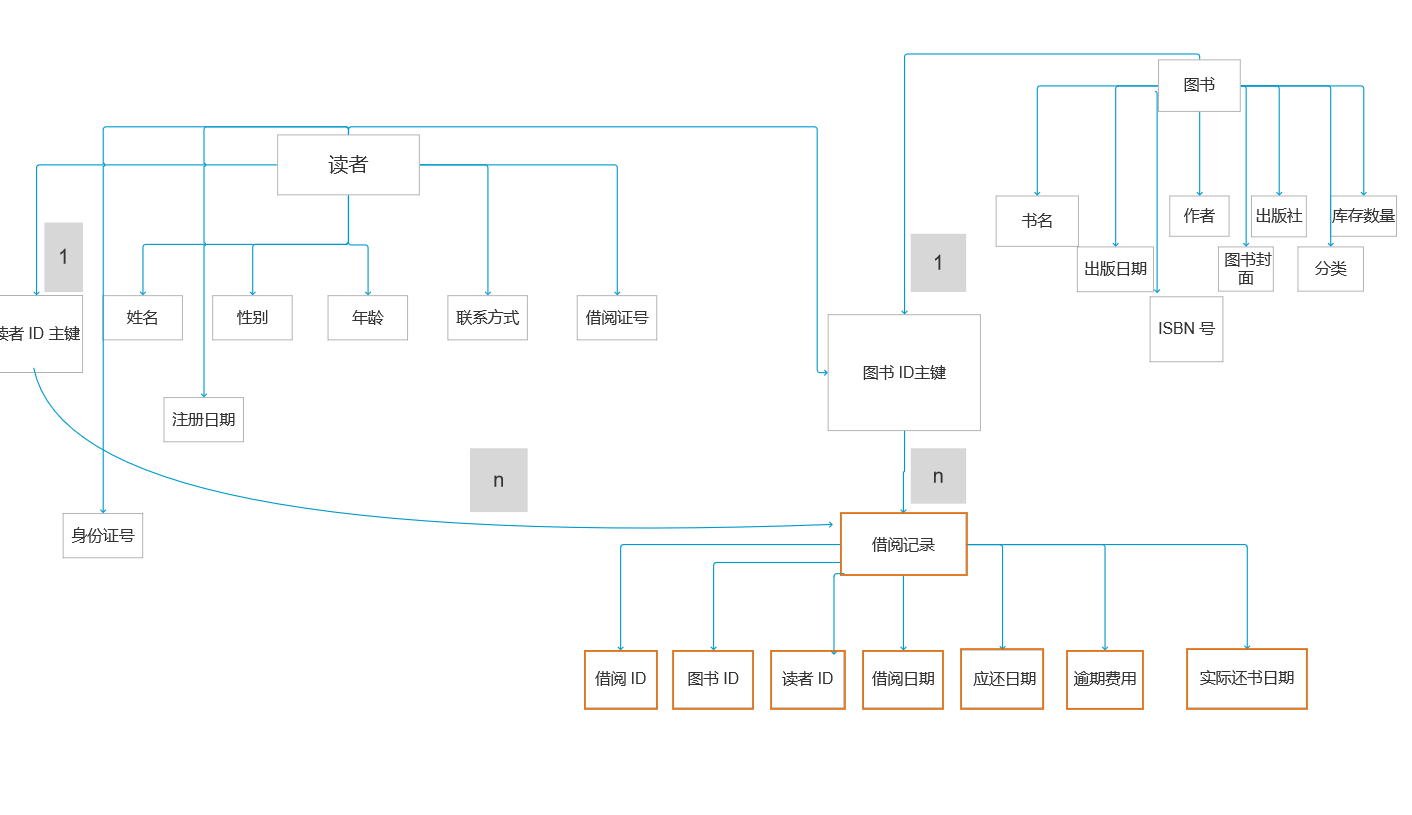
本概要设计旨在为图书管理系统提供一个高层次的架构蓝图，依据前期的需求分析报告，规划系统的模块组成、数据结构、系统架构以及主要流程，确保系统开发的高效性、可维护性与扩展性，以满足图书馆日常运营管理及读者服务需求。

## **二、系统概述**

图书管理系统是一个面向图书馆工作人员和读者的综合性信息管理平台，涵盖图书管理、读者管理、借阅管理、预约管理以及系统管理等核心业务功能，通过信息化手段提升图书管理的精准度与效率，优化读者借阅体验，为图书馆的科学决策提供数据支持。

## **三、系统架构设计**

1. ****数据库架构****
   1. 选用关系型数据库（如 MySQL、Oracle）来存储系统数据，基于数据库规范化设计原则，设计以下主要数据表：
      1. ****图书表（Book）****：包含图书 ID（主键）、ISBN 号、书名、作者、出版社、出版日期、版次、价格、分类 ID（关联分类表）、库存数量、馆藏位置、是否可借等字段，用于详细记录每本图书的信息。
      2. ****读者表（Reader）****：记录读者的个人信息，包括读者 ID（主键）、姓名、性别、年龄、身份证号、联系电话、电子邮箱、所在单位或学校、读者类型、注册日期、账号状态等字段，为读者管理提供数据支撑。
      3. ****借阅表（Borrow）****：关联图书与读者的借阅关系，包含借阅 ID（主键）、图书 ID（外键关联图书表）、读者 ID（外键关联读者表）、借阅时间、应还时间、实际归还时间、逾期天数、是否续借等字段，用于跟踪图书借阅流程。
      4. ****分类表（Category）****：存储图书分类信息，如分类 ID（主键）、分类名称，便于对图书进行分类管理与检索。
      5. ****预约表（Reservation）****：记录读者的图书预约信息，包括预约 ID（主键）、图书 ID（外键关联图书表）、读者 ID（外键关联读者表）、预约时间、期望借阅时间段等字段，实现图书预约功能的数据管理。
   2. E-R图展示：

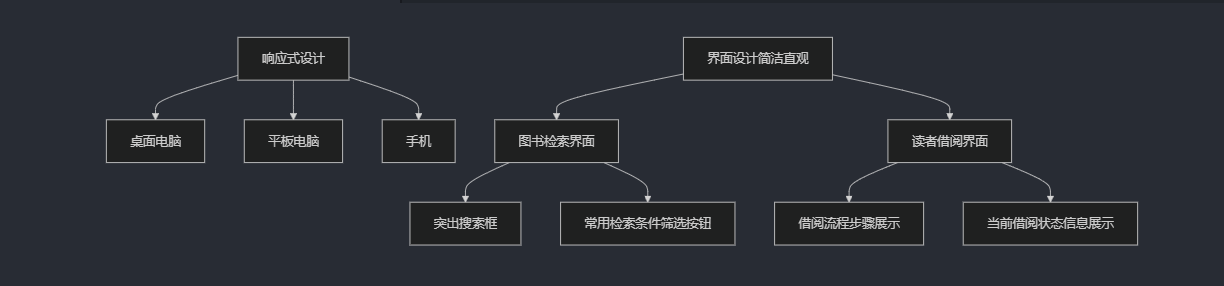


1. **系统结构**

1. 表现层

• 采用响应式设计，适配不同终端设备屏幕尺寸，无论是桌面电脑、平板电脑还是手机，都能提供良好的用户体验。

• 界面设计简洁直观，遵循常见的图书馆管理系统交互模式，方便管理员和读者快速上手操作。例如，图书检索界面设置突出的搜索框和常用检索条件筛选按钮；读者借阅界面清晰展示借阅流程步骤和当前借阅状态信息等。

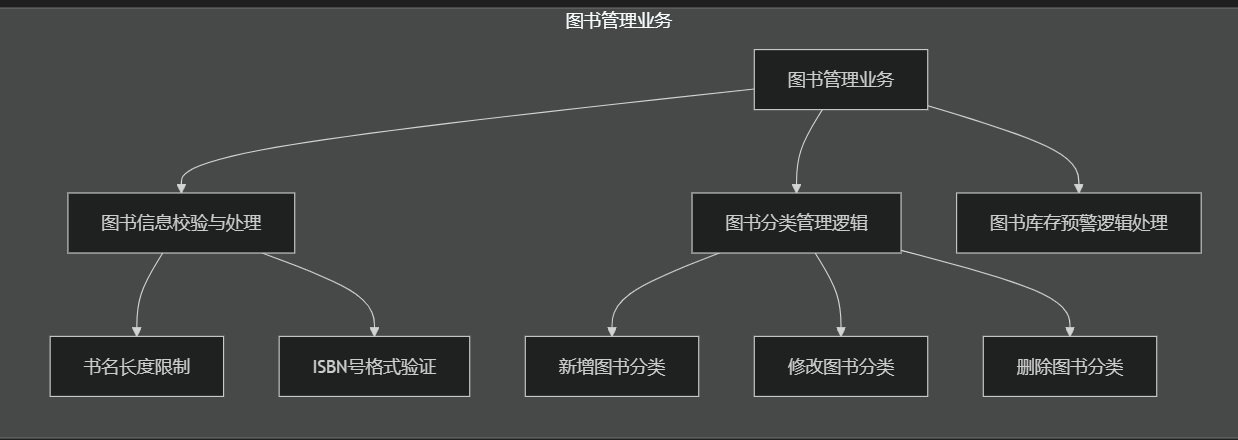


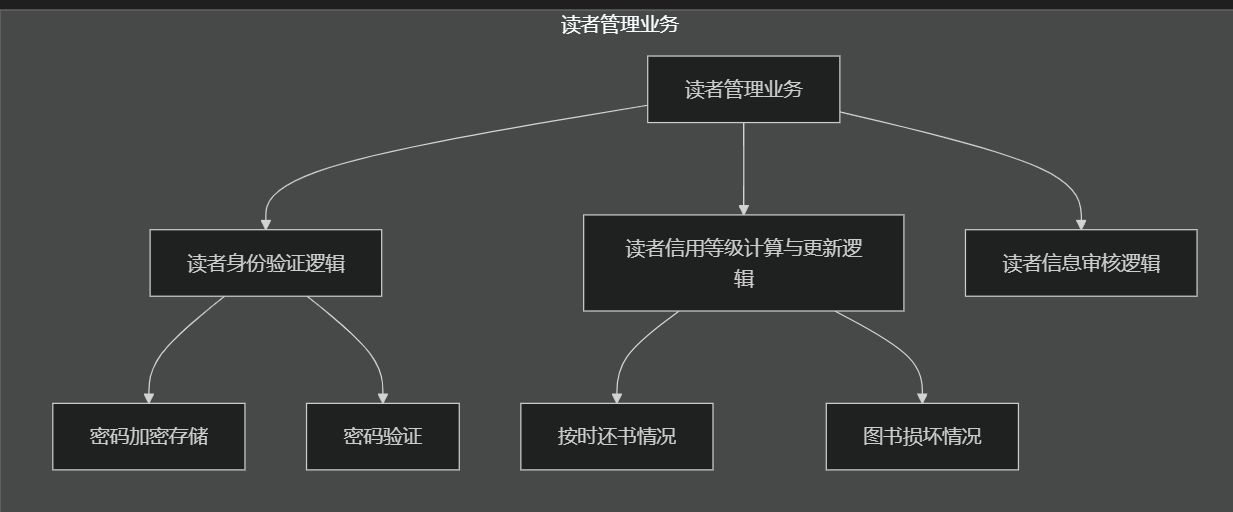
1. 业务逻辑层

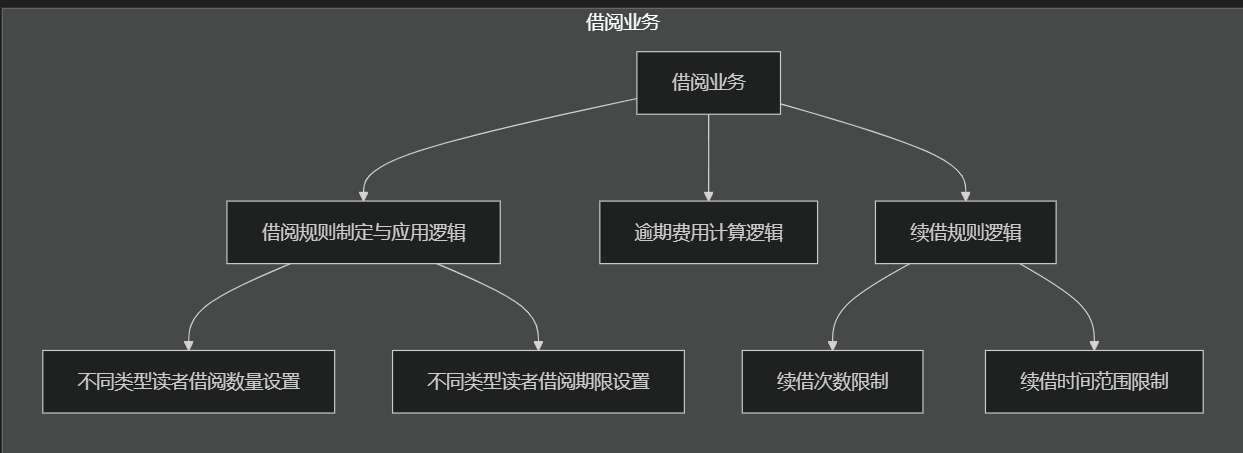
• 图书管理业务：包括图书信息的校验与处理，如书名长度限制、ISBN 号格式验证等；图书分类管理逻辑，如新增、修改、删除图书分类；图书库存预警逻辑处理等。

• 读者管理业务：读者身份验证逻辑，如密码加密存储与验证；读者信用等级计算与更新逻辑，根据借阅记录中的按时还书情况、图书损坏情况等综合计算；读者信息审核逻辑等。

• 借阅业务：借阅规则制定与应用逻辑，如不同类型读者借阅数量和期限的设置与判断；逾期费用计算逻辑，按照逾期天数和图书馆规定的费率计算；续借规则逻辑，如续借次数限制、续借时间范围限制等。



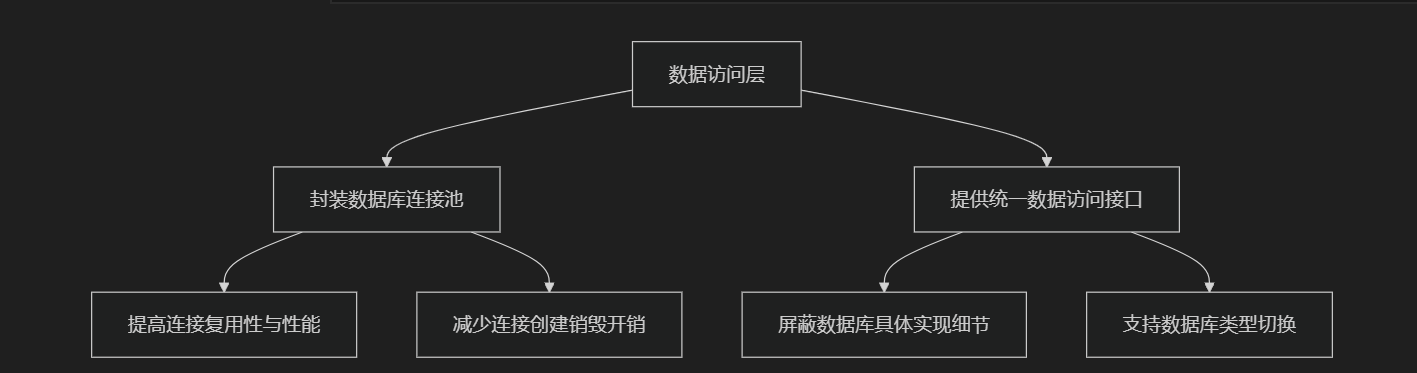




1. 数据访问层

• 封装数据库连接池，提高数据库连接的复用性和性能，减少频繁创建和销毁连接的开销。

• 提供统一的数据访问接口，对上层业务逻辑层屏蔽数据库的具体实现细节，如数据库类型（MySQL、Oracle 等）切换时，只需修改数据访问层的底层实现，而不影响业务逻辑层代码。



## **五、系统流程设计**

1. ****图书入库流程****
   1. 管理员登录系统，进入图书管理模块的图书入库子模块。
   2. 选择手工录入或批量导入方式，若手工录入，填写图书详细信息，系统实时校验必填字段；若批量导入，选择符合格式要求的 Excel 文件。
   3. 录入完成后，系统自动生成图书 ID，根据 ISBN 号（若有）查询补充部分信息，插入图书表，并更新库存数量为初始值。
   4. 入库成功后，系统提示入库信息，包括图书 ID、书名、入库时间等，同时记录入库日志。
2. ****读者借阅流程****
   1. 读者登录系统，在图书查询模块检索所需图书。
   2. 选定图书后，点击借阅按钮，系统验证读者身份及借阅资格。
   3. 若符合条件，系统办理借阅手续，在借阅表中记录借阅信息，更新图书表库存与借阅状态，向读者发送借阅成功通知。
   4. 读者收到通知，确认借阅成功，可查看个人借阅历史记录借阅详情。
3. ****图书归还流程****
   1. 读者携带图书到图书馆，管理员登录系统，进入借阅管理模块的归还办理子模块。
   2. 通过扫码或手动输入图书 ID，系统查找借阅记录，确认归还图书无误。
   3. 系统更新借阅表实际归还时间，计算逾期天数，生成罚款记录（若有），更新图书表借阅状态为 “在库”，向读者发送归还成功通知及逾期情况说明。
   4. 读者收到通知，确认归还完成，如有罚款可按规定缴纳。

## **六、接口设计**

1. ****内部接口****
   1. 表现层与业务逻辑层之间通过接口进行交互，表现层调用业务逻辑层的方法来实现业务功能，业务逻辑层返回处理结果给表现层。例如，在图书查询功能中，表现层接收用户输入的查询条件，调用业务逻辑层的图书查询方法，业务逻辑层组织查询条件并调用数据访问层在图书表中检索，最后将查询结果返回给表现层展示。
   2. 业务逻辑层与数据访问层同样通过接口解耦，业务逻辑层定义数据操作需求，数据访问层实现具体的数据库操作并返回数据。如借阅办理业务，业务逻辑层要求数据访问层插入借阅记录、更新图书表相关字段，数据访问层执行对应的 SQL 语句并反馈执行结果。
2. ****外部接口****
   1. 与财务系统接口：采用 Web Service 或 RESTful API 方式，在借阅归还模块产生逾期罚款、图书赔偿费用时，向财务系统发送包含读者 ID、费用明细、产生原因等信息的请求，财务系统处理后返回缴费确认信息，图书管理系统据此更新读者欠费状态。
   2. 与门禁系统接口：通过串口通信或网络协议（如 TCP/IP）连接，当读者存在逾期未还图书或违规行为时，图书管理系统向门禁系统发送限制读者进入的指令，门禁系统反馈读者出入馆记录给图书管理系统，辅助统计读者到馆率等数据。
   3. 与第三方图书数据平台接口：使用 HTTP 请求调用第三方平台提供的 API，在图书入库或查询功能中，按需获取图书元数据、书评、热门推荐等信息，丰富馆藏资源信息，提升读者服务品质。

## **七、数据存储设计**

1. 数据库选型：基于系统数据量、性能要求、成本效益等因素，选用 MySQL 数据库作为数据存储核心，它具有开源免费、性能稳定、易于维护等优点，能满足图书管理系统的日常运营需求。
2. 数据存储结构：按照数据库设计的规范化原则，构建上述提及的图书表、读者表、借阅表、分类表、预约表等数据表，各表之间通过主键与外键建立关联关系，确保数据的完整性与一致性。例如，借阅表中的图书 ID 关联图书表的主键，读者 ID 关联读者表的主键，通过这种关联，方便查询图书的借阅情况以及读者的借阅历史。
3. 数据备份策略：除了每日自动备份关键业务数据表外，每周进行一次全库备份，备份文件存储在本地冗余磁盘阵列，并定期上传至云存储，以防止本地灾难导致数据丢失；备份文件保留周期为 6 个月，便于在需要时恢复较长时间范围内的数据。

## **八、系统部署设计**

1. ****硬件环境****
   1. 应用服务器：选用配置为 [具体 CPU 型号、核数、内存容量、硬盘容量] 的服务器，运行图书管理系统的应用程序，负责处理用户请求与业务逻辑运算，确保系统的响应速度与吞吐量。
   2. 数据库服务器：配备独立的高性能数据库服务器，配置 [更高性能的 CPU、内存、大容量高速硬盘]，专门用于存储与管理系统数据，保证数据的读写速度与安全性，采用 RAID 技术（如 RAID 5 或 RAID 10）实现数据冗余备份，防止硬盘故障导致数据丢失。
   3. 网络设备：部署千兆以太网交换机，确保服务器与客户端之间的网络通信稳定高速，配备防火墙、入侵检测系统（IDS）等网络安全设备，保障系统网络环境的安全。
2. ****软件环境****
   1. 操作系统：应用服务器安装 Linux（如 CentOS 7）操作系统，利用其稳定性与安全性优势，便于部署与管理 Java 等应用程序；数据库服务器同样选用 Linux 操作系统，优化数据库运行环境。
   2. 数据库管理系统：在数据库服务器上安装 MySQL 8.0 数据库，配置合适的参数（如缓存大小、连接数等），提升数据库性能。
   3. 应用程序框架：在应用服务器上部署基于 Java 的 Spring Boot 框架构建的图书管理系统应用程序，结合前端 Vue.js 框架打造用户友好的界面，利用 Spring Boot 的快速开发、自动配置等特性，提高系统开发效率与运维便利性。
3. ****部署架构****
   1. 采用负载均衡技术，将多台应用服务器组成集群，前端通过负载均衡器（如 Nginx）均匀分配用户请求，避免单台服务器负载过重，提高系统的可用性与扩展性；数据库服务器采用主从复制架构，主库负责写操作，从库负责读操作，通过数据同步机制保证数据一致性，提升数据库的并发处理能力，满足图书管理系统在业务高峰期的需求。