# **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构建一个清晰的蓝图，明确系统的功能架构、数据组织、运行环境以及系统架构，以指导后续的详细设计与开发工作，确保满足图书馆日常业务运营需求，提高图书管理效率和读者服务质量。

二、系统概述  
图书管理系统是一个集成化信息管理平台，面向图书馆管理员和读者两大用户群体，涵盖图书资源管理、读者信息管理、借阅流程管理、预约服务以及系统维护管理等核心业务功能，实现图书信息的精准掌控、借阅流程的自动化与便捷化、读者服务的个性化提升以及系统的稳定可靠运行。

三、功能模块设计

1. ****图书管理模块****
2. ****图书入库****：管理员录入新书详细信息，包括 ISBN 号、书名、作者、出版社、出版日期、价格、分类等，系统自动生成唯一图书 ID，同时记录入库时间、入库数量，支持批量导入功能，提高入库效率。通过 ISBN 号自动联网查询补充部分信息（如书名、作者，若 ISBN 数据库接口支持），确保信息准确性。
3. ****图书查询****：提供多维度检索，如按书名、作者、出版社、ISBN 号、分类、出版年份精准查询或模糊查询，支持多条件组合查询。查询结果展示图书基本信息、馆藏数量、可借数量、借阅状态等，还可按借阅频率、出版时间等排序，方便快速定位目标图书。
4. ****图书修改****：根据图书 ID 定位图书，管理员可修改除图书 ID 外的所有信息，如书籍版本更新、出版社变更等，修改记录自动保存到日志，便于追溯信息变更历史。
5. ****图书下架****：对于损坏、遗失、淘汰的图书，管理员执行下架操作，记录下架原因，更新库存状态，关联的借阅记录置为无效（若有），确保数据一致性。
6. ****图书盘点****：定期生成盘点清单，涵盖所有图书信息，管理员对照实际库存核对，标记差异图书（数量不符、位置错误等），系统根据盘点结果自动更新图书库存与位置信息，保证账实相符。
7. ****读者管理模块****
8. ****读者注册****：新读者在线提交个人信息，如姓名、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（学生、教师、普通市民等），系统验证信息合法性（如身份证号格式、邮箱格式），通过后生成读者账号，初始密码以短信或邮件形式告知读者，完成注册流程。
9. ****读者信息修改****：读者登录后可自行修改部分个人信息，如联系电话、电子邮箱；管理员可根据读者 ID 全面修改读者信息，包括审核读者注册申请、调整读者类型、冻结违规读者账号等，保障读者信息动态管理。
10. ****读者查询****：按多种条件（姓名、身份证号、读者账号等）查询读者基本信息、借阅历史、当前借阅状态，方便管理员跟踪读者借阅情况，对逾期未还、频繁借阅等特殊读者进行重点关注。
11. ****借阅管理模块****
12. ****借阅办理****：读者选定图书后，系统验证读者身份与借阅资格（是否存在逾期未还、是否超借阅上限），若合格则自动生成借阅记录，记录借阅时间、应还时间，更新图书借阅状态为 “借出”、库存数量减一，同时向读者发送借阅成功通知（含借阅图书明细、借阅期限）。
13. ****归还办理****：读者归还图书，管理员扫码或手动输入图书 ID，系统确认图书无误后，更新借阅记录中的归还时间，根据借阅规则判断是否逾期，若逾期自动计算罚款金额，更新图书状态为 “在库”，库存数量加一，向读者发送归还成功通知及逾期罚款信息（若有）。
14. ****续借管理****：在借阅期限截止前一定天数（如 3 天），系统自动提醒读者可续借，读者申请续借时，系统检查图书是否符合续借条件（无其他读者预约、未超续借次数上限），若满足则延长借阅期限，更新应还时间，发送续借成功通知；否则反馈续借失败原因。
15. ****借阅规则设置****：管理员可灵活定制不同读者类型的借阅上限（册数）、借阅期限（天数）、续借次数、逾期罚款标准等规则，以适应图书馆多样化运营需求，确保借阅管理公平合理。
16. ****预约管理模块****
17. ****图书预约****：当图书已借出，读者可预约，输入期望借阅时间段，系统按预约时间先后排序，记录预约信息。图书归还后，系统依序通知预约读者借阅，保留预约等待期限（如 2 天），逾期未借自动取消预约，释放图书资源。
18. ****预约查询与取消****：读者可随时查看自身预约状态（排队中、可借阅等），在预约期限内可取消预约；管理员能查询所有预约情况，据此合理调配馆藏资源，提前准备热门图书供应。
19. ****系统管理模块****
20. ****用户权限管理****：为管理员和读者分配不同权限，管理员细分为图书管理员、借阅管理员、系统管理员等角色，各角色对应不同功能模块操作权限，如图书管理员负责图书入库、修改等，借阅管理员专注借阅归还业务，系统管理员掌控系统参数、用户权限设置等全局操作，确保系统安全有序运行。
21. ****数据备份与恢复****：每日自动备份关键业务数据（图书、读者、借阅等表）至本地存储与云端，备份文件按日期命名并保留一定周期（如 30 天），遇到数据丢失或损坏时，可快速选择对应备份文件恢复数据，保障业务连续性。
22. ****系统参数设置****：管理员配置图书馆开放时间、闭馆时间、节假日安排、通知公告模板等参数，这些信息在系统首页、借阅提醒等多处展示，方便读者知晓图书馆运营动态，优化服务体验。
23. ****日志管理****：系统自动记录所有用户操作日志，涵盖登录时间、操作内容、操作结果等，管理员可按时间、用户角色、操作类型等筛选查询日志，用于追溯问题、审计操作，维护系统合规性。

四、数据库设计

1. ****ER 图****

****1.实体****：

****1.图书（Book）****：具有属性图书 ID（主键）、ISBN 号、书名、作者、出版社、出版日期、价格、分类 ID（关联分类表）、库存数量、馆藏位置、借阅状态，描述图书基本特征与状态信息。

****2.读者（Reader）****：含读者 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、账号状态，用于存储读者身份与个人资料。

****3.借阅（Borrow）****：以借阅 ID（主键）关联图书 ID（外键指向图书表）与读者 ID（外键指向读者表），另有借阅时间、应还时间、实际归还时间、逾期天数、是否续借等属性，记录图书借阅过程详情。

****4.分类（Category）****：有分类 ID（主键）、分类名称，为图书分类提供标准，便于检索与管理。

****5.预约（Reservation）****：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，支持图书预约业务数据存储。

****2.关系****：

1.图书与借阅是一对多关系，一本图书可被多次借阅；读者与借阅也是一对多关系，一位读者可多次借阅不同图书；图书与分类是多对一关系，多本图书可归为同一分类；读者与预约是一对多关系，一位读者可多次预约不同图书；图书与预约同样是一对多关系，一本图书可被多位读者预约。

1. ****数据表设计****
   1. ****Book 表****：

| **字段名** | **数据类型** | **主键** | **外键** | **描述** |
| --- | --- | --- | --- | --- |
| book\_id | INT | 是 | 否 | 图书 ID，自增长 |
| isbn | VARCHAR(20) | 否 | 否 | ISBN 号 |
| title | VARCHAR(100) | 否 | 否 | 书名 |
| author | VARCHAR(50) | 否 | 否 | 作者 |
| publisher | VARCHAR(50) | 否 | 否 | 出版社 |
| publication\_date | DATE | 否 | 否 | 出版日期 |
| price | DECIMAL(8,2) | 否 | 否 | 价格 |
| category\_id | INT | 否 | 是，关联 Category 表的 category\_id | 分类 ID |
| stock\_quantity | INT | 否 | 否 | 库存数量 |
| location | VARCHAR(50) | 否 | 否 | 馆藏位置 |
| borrow\_status | ENUM (' 在库 ',' 借出 ') | 否 | 否 | 借阅状态 |

* 1. ****Reader 表****：

| **字段名** | **数据类型** | **主键** | **外键** | **描述** |
| --- | --- | --- | --- | --- |
| reader\_id | INT | 是 | 否 | 读者 ID，自增长 |
| name | VARCHAR(50) | 否 | 否 | 姓名 |
| gender | CHAR(1) | 否 | 否 | 性别 |
| age | INT | 否 | 否 | 年龄 |
| identity\_card | VARCHAR(18) | 否 | 否 | 身份证号 |
| phone | VARCHAR(20) | 否 | 否 | 联系电话 |
| email | VARCHAR(50) | 否 | 否 | 电子邮箱 |
| organization | VARCHAR(50) | 否 | 否 | 所在单位或学校 |
| reader\_type | VARCHAR(20) | 否 | 否 | 读者类型 |
| account\_status | ENUM (' 正常 ',' 冻结 ') | 否 | 否 | 账号状态 |

* 1. ****Borrow 表****：

| **字段名** | **数据类型** | **主键** | **外键** | **描述** |
| --- | --- | --- | --- | --- |
| borrow\_id | INT | 是 | 否 | 借阅 ID，自增长 |
| book\_id | INT | 否 | 是，关联 Book 表的 book\_id | 图书 ID |
| reader\_id | INT | 否 | 是，关联 Reader 表的 reader\_id | 读者 ID |
| borrow\_time | DATETIME | 否 | 否 | 借阅时间 |
| due\_time | DATETIME | 否 | 否 | 应还时间 |
| return\_time | DATETIME | 否 | 否 | 实际归还时间 |
| overdue\_days | INT | 否 | 否 | 逾期天数 |
| is\_renewed | BOOLEAN | 否 | 否 | 是否续借 |

* 1. ****Category 表****：

| **字段名** | **数据类型** | **主键** | **外键** | **描述** |
| --- | --- | --- | --- | --- |
| category\_id | INT | 是 | 否 | 分类 ID，自增长 |
| category\_name | VARCHAR(50) | 否 | 否 | 分类名称 |

* 1. ****Reservation 表****：

| **字段名** | **数据类型** | **主键** | **外键** | **描述** |
| --- | --- | --- | --- | --- |
| reservation\_id | INT | 是 | 否 | 预约 ID，自增长 |
| book\_id | INT | 否 | 是，关联 Book 表的 book\_id | 图书 ID |
| reader\_id | INT | 否 | 是，关联 Reader 表的 reader\_id | 读者 ID |
| reservation\_time | DATETIME | 否 | 否 | 预约时间 |
| expected\_borrow\_time | DATETIME | 否 | 否 | 期望借阅时间段 |

五、系统结构

1. ****分层架构****

****1.表现层****：采用 Web 技术构建用户界面，使用 HTML、CSS、JavaScript 实现页面布局与交互效果，结合前端框架（如 Vue.js 或 React）提升开发效率与用户体验，负责接收用户输入、展示系统输出，将用户请求传递给业务逻辑层处理。

****2.业务逻辑层****：用面向对象编程语言（如 Java、Python）编写，包含各类业务规则与流程实现，接收表现层请求，调用数据访问层操作数据库，处理数据并返回结果给表现层，是系统核心处理单元，如借阅业务逻辑中验证读者资格、计算借阅期限等在此完成。

****3.数据访问层****：运用数据库访问技术（如 JDBC、SQLAlchemy），针对不同业务需求对数据库表执行增删改查操作，向业务逻辑层提供统一数据接口，隐藏数据库细节，确保数据存取高效准确。

1. ****模块间协作****

1.表现层与业务逻辑层通过接口调用实现交互，业务逻辑层为表现层提供服务方法，表现层传递用户参数并接收处理结果。业务逻辑层与数据访问层同样基于接口通信，业务逻辑层定义数据操作需求，数据访问层实现具体数据库 SQL 操作并反馈执行结果，各层职责明确，松耦合协作保障系统灵活性与可维护性。

六、运行平台及架构

1. ****硬件平台****

****1.服务器****：配置高性能服务器，建议 CPU 为 [具体型号]，多核处理器（如四核或八核），内存 16GB 及以上，硬盘采用高速固态硬盘（SSD）500GB 及以上，保障系统运行速度与数据存储读写效率，满足图书馆日常业务高峰需求。

****2.网络设备****：配备千兆以太网交换机，确保内部网络高速稳定连接，若面向外网开放，需部署防火墙、入侵检测系统（IDS）等网络安全设备，防范外部网络攻击，保障系统网络安全。

1. ****软件平台****

****1.操作系统****：服务器端选用 Linux 操作系统（如 CentOS 或 Ubuntu），基于其稳定性、安全性与开源特性，便于系统部署、维护与性能优化，客户端支持主流桌面操作系统（如 Windows、Mac OS）及移动操作系统（如 Android、iOS）浏览器访问。

****2.数据库管理系统****：采用关系型数据库 MySQL 或 Oracle，考虑到成本与易用性，MySQL 为较优选择，能有效存储与管理系统海量数据，提供高效数据查询与事务处理能力。

****3.应用服务器****：选用 Tomcat 或 Jetty 等 Java 应用服务器（若基于 Java 开发），负责部署运行图书管理系统的 Web 应用程序，将用户请求分发给相应业务逻辑处理模块，实现系统高效运行。

1. ****架构模式****
2. 采用 B/S（浏览器 / 服务器）架构，用户通过浏览器访问图书管理系统，无需安装专门客户端软件，降低用户使用门槛，便于系统维护与升级，只需在服务器端更新代码，用户即可即时使用新功能，适应图书馆多用户、跨地域使用场景。同时，可结合负载均衡技术（如 Nginx）将用户请求均匀分配到多个应用服务器实例，提升系统并发处理能力，避免单点过载，保障系统高可用性。

**六． 表结构设计示例（MySQL 语法）**

**-- 图书表**

**CREATE TABLE books (**

**book\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**isbn VARCHAR(13) NOT NULL UNIQUE,**

**title VARCHAR(255) NOT NULL,**

**author VARCHAR(255) NOT NULL,**

**publisher VARCHAR(255),**

**publication\_year INT,**

**price DECIMAL(8, 2),**

**stock\_quantity INT DEFAULT 0,**

**category VARCHAR(50),**

**book\_cover VARCHAR(255), -- 存储封面图片路径**

**book\_introduction TEXT -- 存储图书简介**

**);**

**-- 读者表**

**CREATE TABLE readers (**

**reader\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**name VARCHAR(255) NOT NULL,**

**gender ENUM('男', '女', '其他') NOT NULL,**

**age INT,**

**contact\_number VARCHAR(15),**

**email VARCHAR(255),**

**id\_number VARCHAR(18) NOT NULL UNIQUE,**

**registration\_date DATE,**

**borrowing\_card\_number VARCHAR(20) -- 借阅证号**

**);**

**-- 借阅表**

**CREATE TABLE borrows (**

**borrow\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**reader\_id INT NOT NULL,**

**book\_isbn VARCHAR(13) NOT NULL,**

**borrow\_date DATE NOT NULL,**

**due\_date DATE NOT NULL,**

**return\_date DATE,**

**status ENUM('未归还', '已归还') DEFAULT '未归还',**

**overdue\_fee DECIMAL(8, 2) DEFAULT 0, -- 逾期费用**

**FOREIGN KEY (reader\_id) REFERENCES readers(reader\_id),**

**FOREIGN KEY (book\_isbn) REFERENCES books(isbn)**

**);**

**-- 图书分类表**

**CREATE TABLE book\_categories (**

**category\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**category\_name VARCHAR(50) NOT NULL UNIQUE,**

**category\_description VARCHAR(255)**

**);**

**-- 图书分类关系表**

**CREATE TABLE book\_category\_relations (**

**relation\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**book\_id INT NOT NULL,**

**category\_id INT NOT NULL,**

**FOREIGN KEY (book\_id) REFERENCES books(book\_id),**

**FOREIGN KEY (category\_id) REFERENCES book\_categories(category\_id)**

**);**

**-- 图书位置表**

**CREATE TABLE book\_locations (**

**location\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**book\_id INT NOT NULL,**

**shelf\_number VARCHAR(10),**

**layer\_number INT,**

**FOREIGN KEY (book\_id) REFERENCES books(book\_id)**

**);**

**-- 读者信息变更申请表**

**CREATE TABLE reader\_change\_requests (**

**request\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**reader\_id INT NOT NULL,**

**change\_type VARCHAR(50) NOT NULL,**

**change\_reason VARCHAR(255) NOT NULL,**

**request\_status ENUM('待审核', '已通过', '已拒绝') DEFAULT '待审核',**

**FOREIGN KEY (reader\_id) REFERENCES readers(reader\_id)**

**);**

**-- 借阅规则表**

**CREATE TABLE borrowing\_rules (**

**rule\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**reader\_type VARCHAR(20) NOT NULL,**

**book\_type VARCHAR(50) NOT NULL,**

**borrowing\_quantity\_limit INT NOT NULL,**

**borrowing\_period INT NOT NULL**

**);**

**-- 费用表**

**CREATE TABLE fees (**

**fee\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**reader\_id INT NOT NULL,**

**book\_isbn VARCHAR(13) NOT NULL,**

**overdue\_days INT,**

**amount DECIMAL(8, 2)**

**);**

**-- 采购订单表**

**CREATE TABLE purchase\_orders (**

**order\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**book\_id INT,**

**order\_date DATE NOT NULL,**

**supplier VARCHAR(255),**

**quantity INT NOT NULL,**

**order\_status ENUM('已下单', '已入库', '已取消') DEFAULT '已下单',**

**FOREIGN KEY (book\_id) REFERENCES books(book\_id)**

**);**

**-- 主题词表**

**CREATE TABLE keywords (**

**keyword\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**keyword\_name VARCHAR(50) NOT NULL UNIQUE**

**);**

**-- 图书主题词关系表**

**CREATE TABLE book\_keyword\_relations (**

**relation\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**book\_id INT NOT NULL,**

**keyword\_id INT NOT NULL,**

**FOREIGN KEY (book\_id) REFERENCES books(book\_id),**

**FOREIGN KEY (keyword\_id) REFERENCES keywords(keyword\_id)**

**);**