软件综合实践课程报告

## 一.引言

### （1）项目背景

随着信息技术的迅猛发展，计算机的普及率不断提高，越来越多的领域开始引入信息化管理。图书馆作为知识和信息的重要载体，也迫切需要借助现代化的技术手段来提升管理效率和服务质量。此外，随着图书数量的不断增多，传统的手工管理方式已经难以满足日益复杂的管理需求，而庞大的图书资源，也带来了数量庞大的读者，使读者需求日益多样化。

**1. 图书数量的激增**

现代图书馆面临的一个重要挑战是图书数量的激增。随着图书出版行业的快速发展，新书的数量每年都在增加。传统的手工管理方式已经难以应对如此庞大的图书量，容易出现管理混乱、图书丢失和误借的现象。一本图书如果不能被及时归类和记录，很可能在图书馆的某个角落被遗忘，无法被需要的读者找到。图书馆需要一种高效的方式来记录和管理这些图书，以确保它们能够被准确地分类、存储和查找。

**2. 读者需求的多样化**

新时代的读者对图书馆的期望不仅限于借阅图书，他们希望能够方便、快捷地获取图书信息。读者希望通过系统查询图书的详细信息，包括内容简介、作者信息、出版信息等，并能方便地查看图书的借阅状态、推荐指数等。读者还希望能够在线预约图书，查看借阅历史，甚至是获取个性化的图书推荐。为了满足这些需求，图书馆管理系统需要具备强大的查询功能和灵活的用户界面，以提升读者的使用体验。

**3. 读者对评价和互动的重视**

读者评价已经成为影响图书借阅的重要因素。一个有效的图书馆管理系统应能够允许读者对图书进行评分和评论，从而帮助其他读者选择合适的图书。同时，这些评价数据也能为图书馆的藏书调整和优化提供参考依据。读者的反馈不仅可以帮助其他读者选择图书，还可以帮助图书馆了解哪些图书最受欢迎，哪些图书需要增加库存，甚至是淘汰一些不受欢迎的图书。通过引入读者评价和互动功能，图书馆可以更好地了解读者的需求和偏好，从而提供更好的服务。

**4. 信息化和智能化管理的普及**

随着计算机技术的进步，信息化、智能化管理成为可能。通过信息化管理，图书馆可以实现图书的自动分类、借还书的智能化处理、读者行为的自动分析等，从而大幅提高管理效率，降低人工成本。例如，系统可以自动记录每本书的借阅情况，计算每本书的借阅频率，预测未来的借阅需求，从而帮助图书馆合理安排图书的采购和存储。此外，系统还可以提供数据备份和恢复功能，确保图书信息和借阅记录的安全性，防止数据丢失。

### （2）编写目的

该系统的主要目的是通过信息化手段，实现图书的电子化管理，简化图书的借还流程，提供便捷的图书查询和管理功能，满足新时代读者多样化的需求，提升图书馆的整体服务水平。

**具体目标包括：**

1. **提高管理效率：**通过系统化的管理方式，减少人工操作，提高图书借还的效率。
2. **提升用户体验：**读者可以通过系统方便地查询图书信息、借阅记录、借阅状态等，增强用户体验。
3. **支持读者评价：**允许读者对图书进行评分和评论，提供图书推荐功能。

**目标用户**：

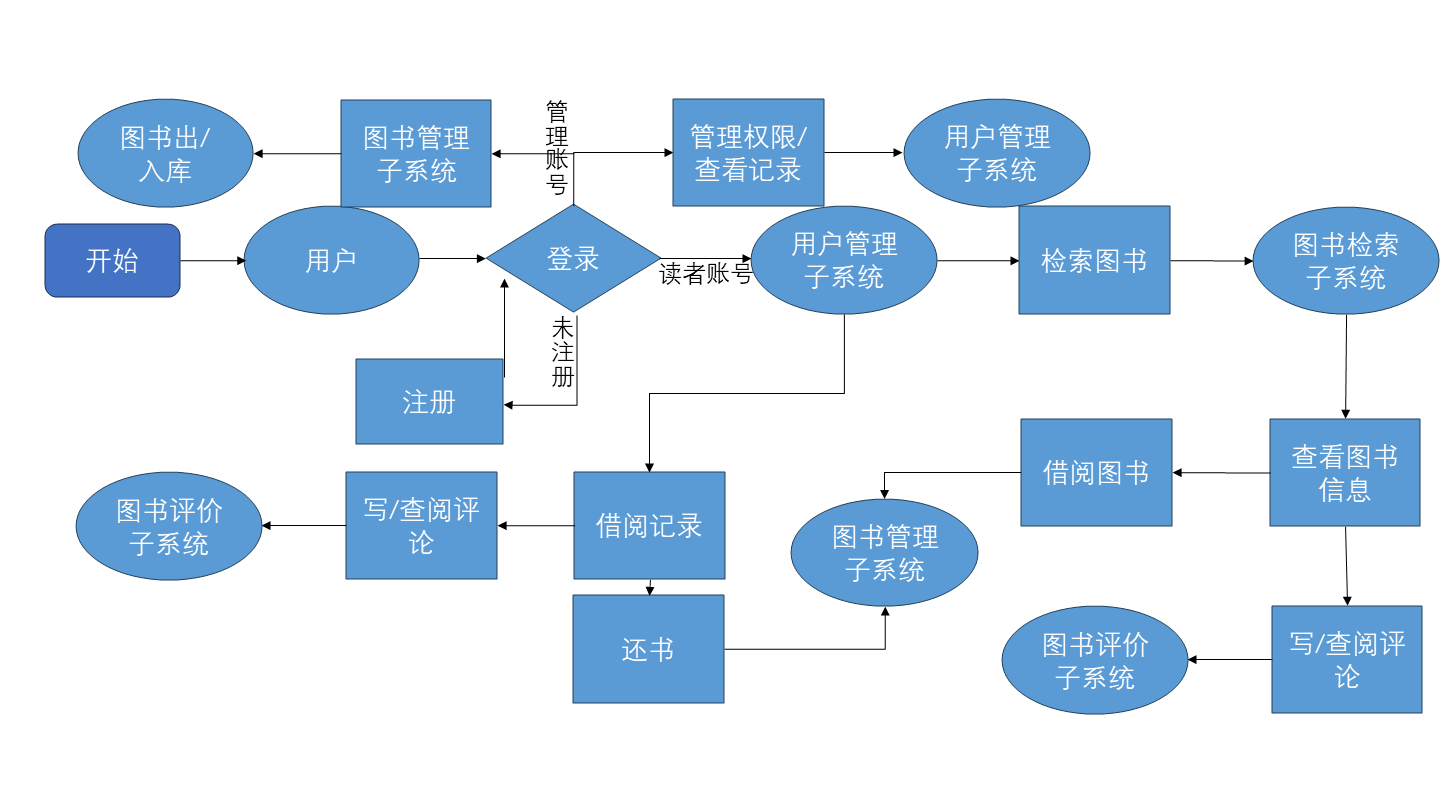
1. **图书馆管理员**：负责图书的入库、出库、管理以及系统的维护。
2. **读者**：包括学生、教师及其他图书馆用户，主要使用系统查询图书信息、借阅和归还图书。

**项目预期成果：**

1. **实现图书的信息化管理：**包括图书信息录入、修改、删除、查询等功能。通过系统，图书馆管理员可以轻松录入新书信息，更新图书记录，删除旧书数据，查询任意图书的详细信息。
2. **实现读者的信息化管理：**包括读者信息的录入、修改、查询等功能。系统将帮助图书馆管理员管理读者信息，如注册新读者、更新读者资料、查询读者的借阅历史。
3. **实现图书的借还管理：**包括借书、还书、续借、预约等功能。读者可以在线借阅和归还图书，查看自己的借阅记录和还书期限，申请续借，预约已被借出的图书。
4. **提供读者评价和互动功能：**支持读者对图书进行评分和评论。读者可以分享他们对图书的看法和阅读体验，帮助其他读者选择图书。

## 二.需求分析

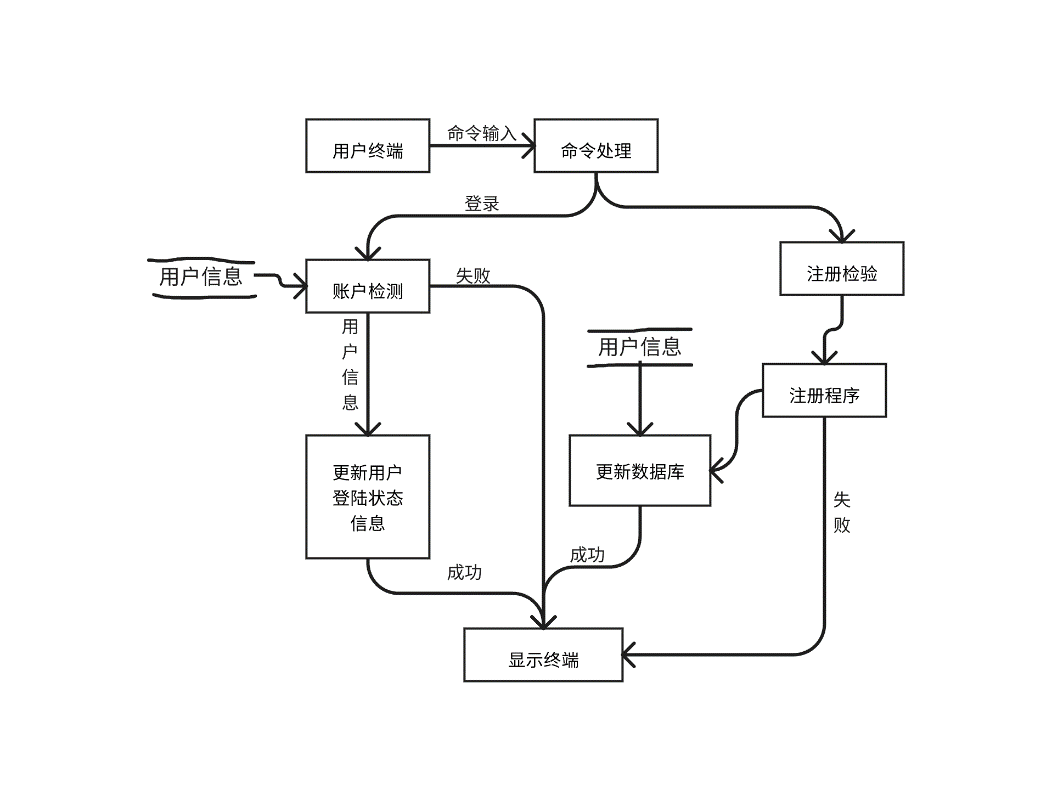
### （1）业务需求



用户登录时区分出管理员账号，读者账号，未注册账号，并进入不同的界面（未注册账号停留当前界面），管理员登录后可通过图书管理子系统进行图书出入库操作，通过用户管理子系统管理权限和查看记录；读者登录后可以使用用户管理子系统的借阅记录查询功能、图书检索系统检索图书与查看图书信息功能、图书管理系统借阅与归还功能、图书评价子系统写/查阅评论功能。

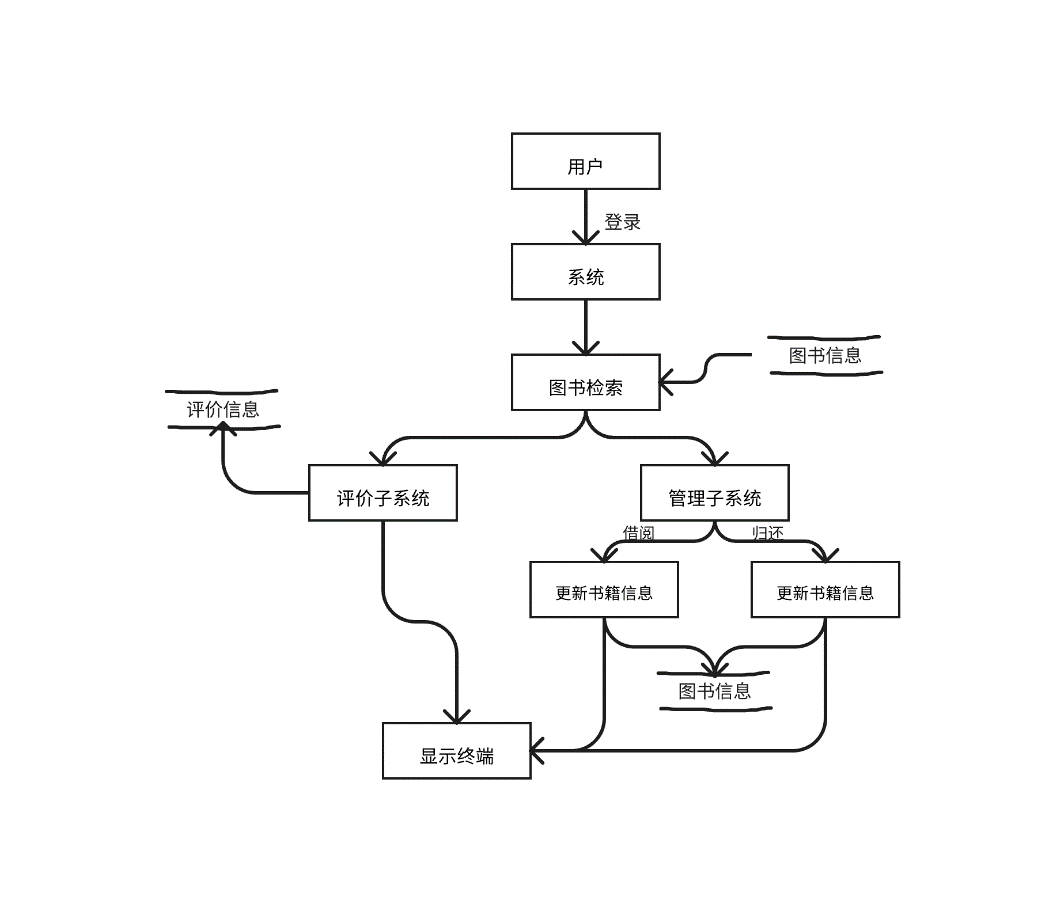
### （2）功能需求

* 1. **用户登录**

****

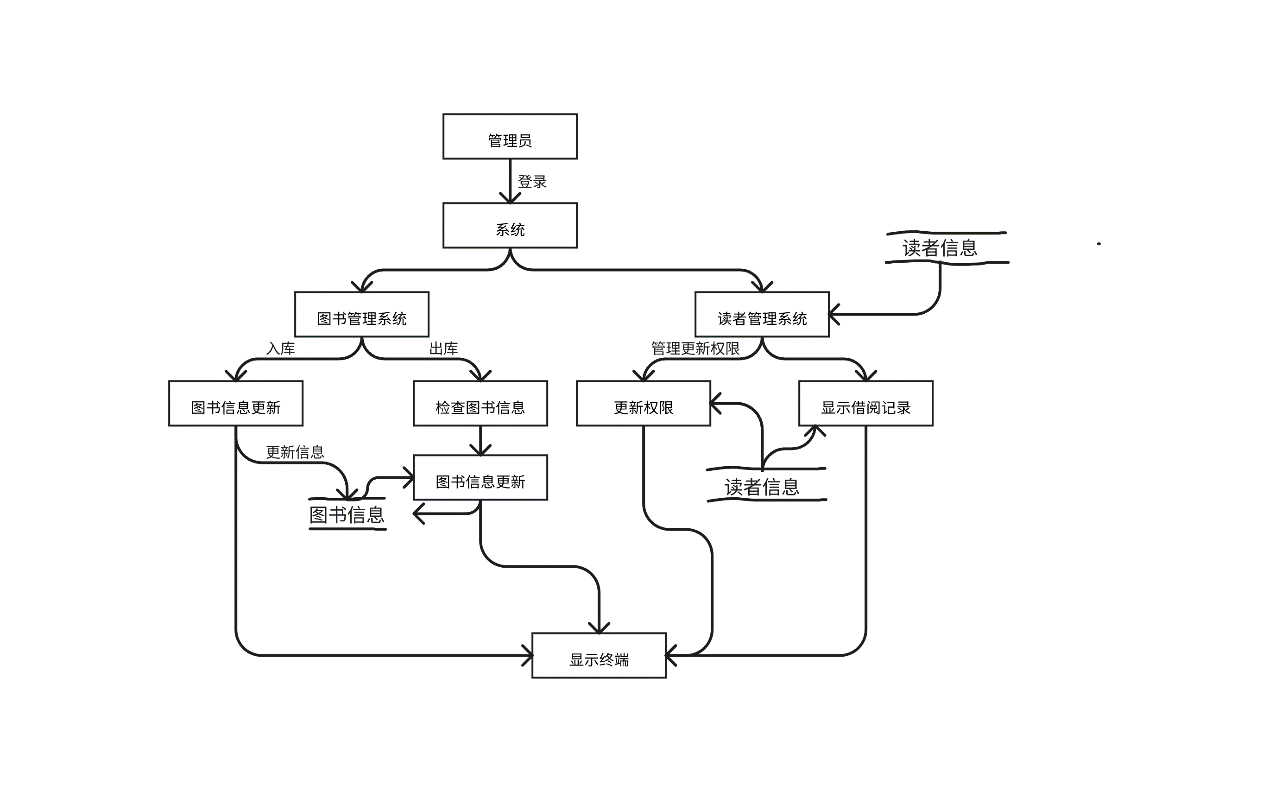
用户通过登陆页面进行输入账号和密码，再由前端页面传输的数据进行数据库的查询从而完成登录

* 1. **读者借阅评价图书**

****

用户登录后可以进行图书查询以及借还书等功能

* 1. **管理员管理读者与图书**

****

管理员登录后可以对图书进行管理，包括对图书类别的增删改以及对图书的增删改，也可以对用户进行修改，可以进行图书评论的审核以及删除。

### （3）非功能性需求

1. **安全性**

图书馆管理系统包括用户管理子系统，储存用户的账号密码，考虑到读者借阅权利的保证与隐私的保护，需实现密码在数据库的透明加密

1. **可维护性**

图书馆管理系统在之后的应用中有可能进行升级维护，因此在系统的编写中应注重代码的可读性与分层架构的实现，以减小后续维护升级阻力。

1. **易用性**

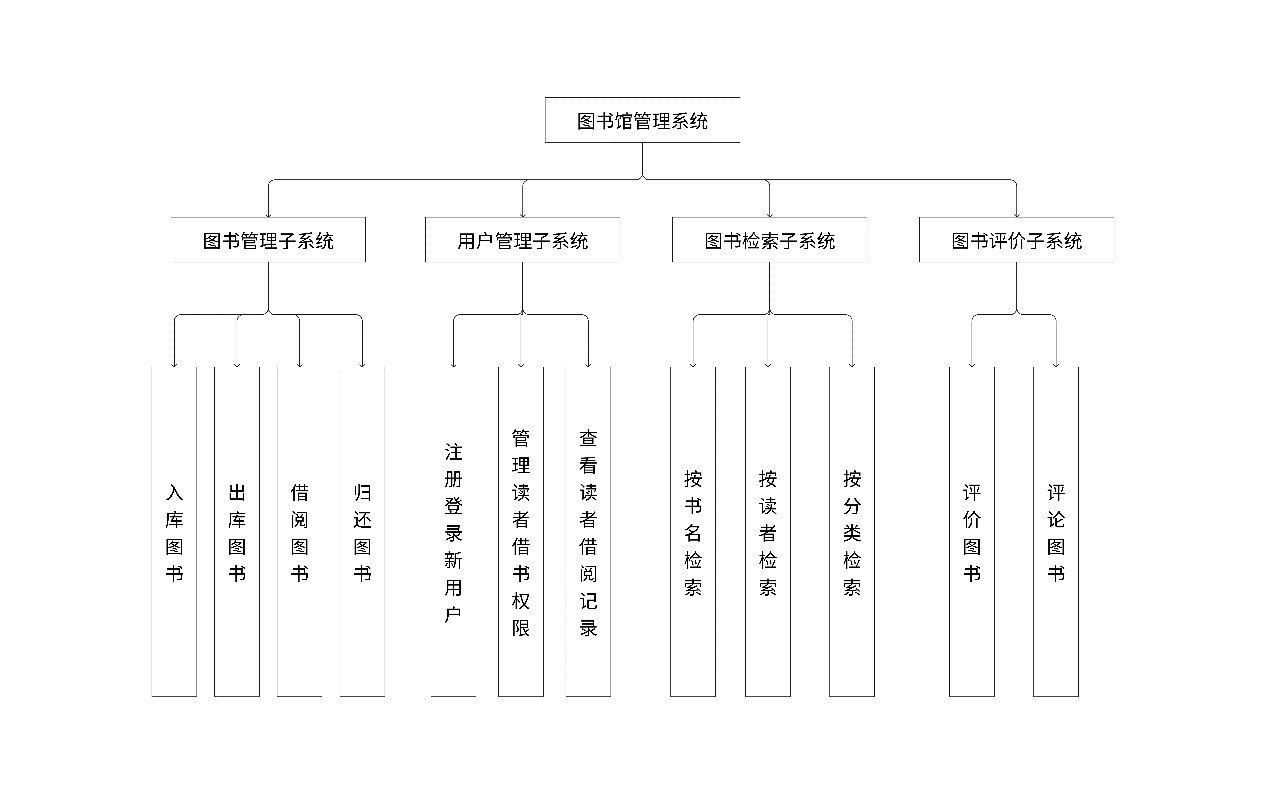
图书馆管理系统主要面对高校师生，应尽量使用用户友好型界面，简洁直观、易于操作。

1. **可移植性**

数据库在不同环境下具有大小写敏感性不一致问题，在系统的编写中应注意以提高系统的可移植性。

## 三.概要设计

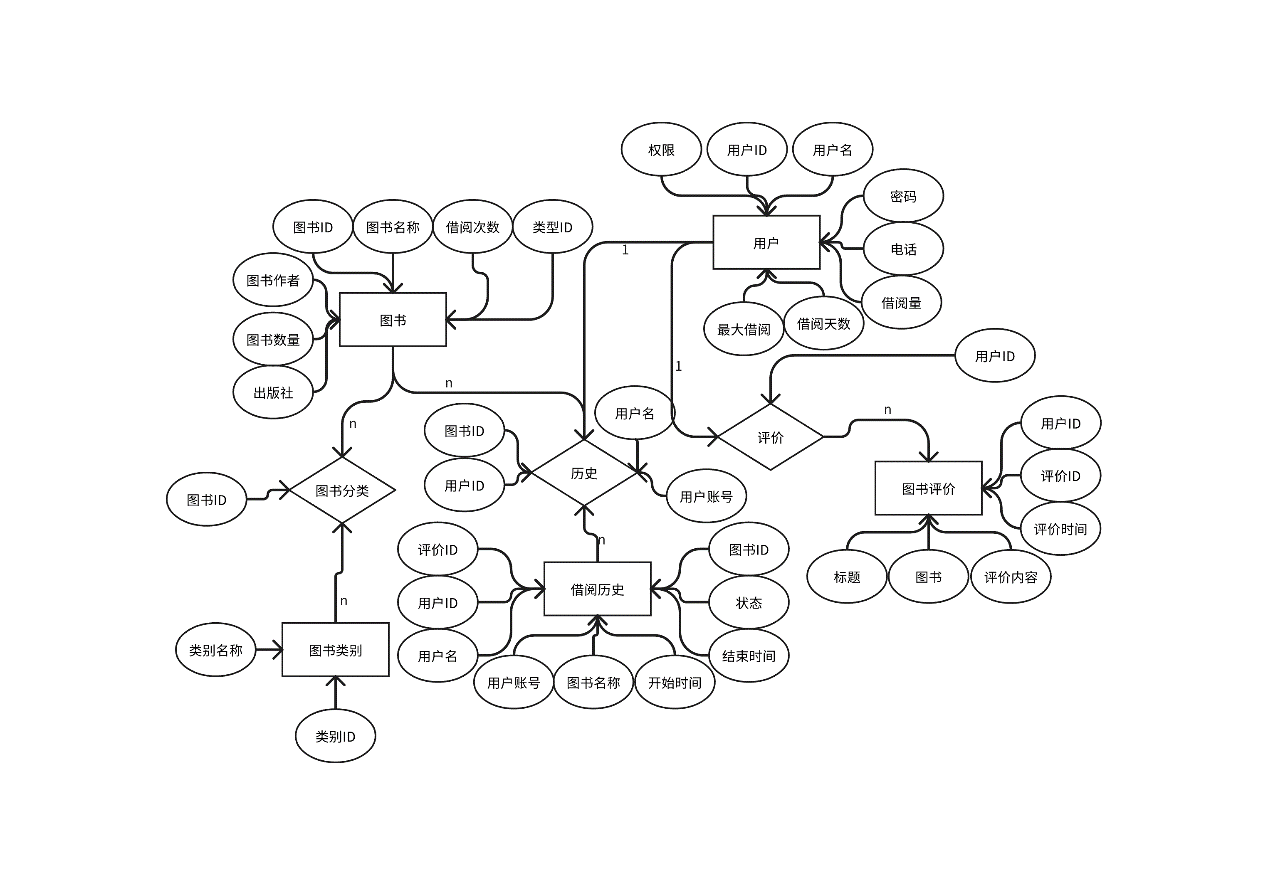
### （1）系统结构与组成



图书馆管理系统分为四个子系统：图书管理子系统包括出/入库图书，借阅/归还图书功能；用户管理子系统包括注册登录新用户、管理读者借书权力与查看读者借阅记录功能；图书检索子系统具有检索功能，包括按书名检索、按作者检索、按分类检索；图书评价子系统包括对图书的评价相关功能

### （2）数据库设计

#### 1.概念设计



该数据库共包含5个表，分别为图书表t\_book，图书类别t\_type，用户表t\_user，借阅历史表t\_history以及评价评论表t\_evaluate。

#### 逻辑设计

**1、E-R图转化为表**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t\_user用户数据库表 | | | | | | | | | | | |
| 表名 | 列名 | | 注释 | | 数据类型 | | 默认值 | 键 | | | 是否为null |
| t\_user | uid | | 用户表主键ID | | int(11) unsigned | | null | PRI | | | NO |
| t\_user | account | | 账号 | | varchar(55) | | null |  | | | YES |
| t\_user | name | | 姓名 | | varchar(55) | | null |  | | | YES |
| t\_user | password | | 密码 | | varchar(55) | | null |  | | | YES |
| t\_user | phone | | 手机 | | varchar(11) | | null |  | | | YES |
| t\_user | times | | 借阅量 | | int(11) | | null |  | | | YES |
| t\_user | lend\_num | | 可借阅天数 | | int(11) | | null |  | | | YES |
| t\_user | max\_num | | 最大可借数量 | | int(11) | | null |  | | | YES |
| t\_user | role | | 1用户 2管理员 | | int(1) | | null |  | | | YES |
| t\_book图书数据库表 | | | | | | | | | | | | |
| 表名 | 列名 | 注释 | | 数据类型 | | 默认值 | | | 键 | 是否为null | | |
| t\_book | bid | 图书信息的ID | | int(11) | | null | | | PRI | NO | | |
| t\_book | book\_name | 图书名称 | | varchar(55) | | null | | |  | YES | | |
| t\_book | author | 作者 | | varchar(55) | | null | | |  | YES | | |
| t\_book | num | 库存 | | int(11) | | null | | |  | YES | | |
| t\_book | press | 出版社 | | varchar(55) | | null | | |  | YES | | |
| t\_book | tid | 类型ID | | int(11) | | null | | |  | YES | | |
| t\_book | times | 被借阅次数 | | int(11) | | null | | |  | YES | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t\_type图书类别数据库表 | | | | | | |
| 表名 | 列名 | 注释 | 数据类型 | 默认值 | 键 | 是否为null |
| t\_type | tid | 类型主键ID | int(11) | null | PRI | NO |
| t\_type | type\_name | 图书类型名称 | varchar(55) | null |  | YES |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t\_history借阅历史数据库表 | | | | | | |
| 表名 | 列名 | 注释 | 数据类型 | 默认值 | 键 | 是否为null |
| t\_history | hid | 记录表ID | int(11) | null | PRI | NO |
| t\_history | uid | 用户ID | int(11) | null |  | YES |
| t\_history | name | 用户姓名 | varchar(55) | null |  | YES |
| t\_history | account | 用户账号 | varchar(55) | null |  | YES |
| t\_history | bid | 图书ID | int(11) | null |  | YES |
| t\_history | book\_name | 图书名称 | varchar(255) | null |  | YES |
| t\_history | begin\_time | 借书时间 | datetime | null |  | YES |
| t\_history | end\_time | 还书时间 | datetime | null |  | YES |
| t\_history | status | 借阅状态 | int(2) | null |  | YES |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t\_evaluate评价和评论数据库表 | | | | | | |
| 表名 | 列名 | 注释 | 数据类型 | 默认值 | 键 | 是否为null |
| t\_evaluate | pid | 主键ID | int(11) | null | PRI | NO |
| t\_evaluate | uid | 评价人ID | int(11) | null |  | YES |
| t\_evaluate | title | 标题 | varchar(55) | null |  | YES |
| t\_evaluate | book | 图书 | varchar(55) | null |  | YES |
| t\_evaluate | content | 内容 | varchar(255) | null |  | YES |
| t\_evaluate | time | 评价时间 | datetime | null |  | YES |

## 四

## .详细设计与实现

### （1）数据库实现

### （2）主要模块实现

## 五.结束语

### （1）设计中遇到的问题与解决方法

所用MYSQL密码字段由password更名为authentication和密码策略要求大小写特殊字符导致密码初始化失败：了解到原因之后将网络教程中的字段更改并设置了带大小写特殊字符的密码。

开发环境不统一导致开发进度同步后频繁报错：尽量在一台机器上完成一个功能的实现。

项目设计初期方向错误，与可行性分析中预定方案不一致：改用tomcat装载javaweb项目，贴近预定方案中的方法。

### （2）有待改进的地方

一开始预设场景为高校图书馆，账号一般批量导入，未考虑其它图书馆场景下可能会有的自助注册需求。改进方向：增加用户自助注册功能，使系统能够适应更多类型的图书馆。提供用户注册表单，并添加必要的验证和审核流程。

未考虑到修改密码需求，未设计代码修改功能。改进方向：在用户管理模块中增加修改密码功能。用户应能够通过输入旧密码并设置新密码来更新他们的账户信息。

借阅期限只有显示作用，到期未设置惩罚。改进方向：实现自动提醒功能，设置借阅超期的惩罚机制，如限制借阅权限、罚款等。系统应能够自动记录超期情况并执行相应的惩罚措施。

数据库未进行设计优化，大规模数据下可能存储和检索效率不高。改进方向：为常用查询添加适当的索引，提高数据检索效率；进行数据库规范化设计，避免数据冗余和更新异常，提高数据一致性和完整性。

**（3）收获和体会**

体会到了Java强大的生态系统和社区支持：学会了利用Java生态系统中的各种库和框架，提高了开发效率和代码质量、通过社区支持解决了开发过程中遇到的许多问题，加快了项目进度

感受到了Tomcat的轻量级和高效性能：Tomcat的配置和管理相对简单，支持多种操作系统，适合快速搭建和运行Web应用

体会到了前端技术的普及和易用性：HTML、CSS和JavaScript是Web前端开发的核心技术，具有普及性和易用性，使得Web应用的用户界面开发更加灵活和强大。

感受到了Git的使用对分布式开发的重要作用：Git的分布式特性允许开发者在本地仓库进行开发并进行频繁的提交，再通过pull request进行代码审查和合并，从而提高团队协作的效率。

了解了web项目的设计过程与其中会经常碰到的问题：如部分未在功能需求说明内提及的非功能性需求问题，非开发环境的部署需要解决的依赖问题等

感受到了分层设计的重要性：良好的架构设计能够提升系统的可维护性和可扩展性。

## 六.参考文献

<https://www.bilibili.com/video/BV1it41157ph/?p=2&spm_id_from=pageDriver&vd_source=5f53fdf3075245880b597df51767aac7>