



浙江农林大学  
ZHEJIANG A&F UNIVERSITY

## 数据库原理与技术课程设计报告 (2023 届)

题    目：\_\_\_\_\_图书管理系统\_\_\_\_\_

组    长：\_\_\_\_\_章心阳\_\_\_\_\_

学号	姓名	分工	组内评分
202205010402	章心阳	数据库的搭建工作，建表、游标、存储过程、触发器等以及报告 sql 语句部分的撰写	优
202205010401	何文静	系统用户端的实现以及报告数据库需求分析部分、项目意义和总结的撰写	良
202205010403	陈伟润	前端页面设计以及报告功能结构图、E-R 图的绘制	优
202205010409	刘烨	系统管理员端的实现以及报告运行结果部分的撰写	良

---

## 目录

1. 作品设计开发的背景及意义.....	1
1.1. 背景.....	1
1.2. 目的.....	1
1.3. 意义.....	1
2. 数据库设计.....	2
2.1. 数据库需求分析.....	2
2.1.1 注册登录.....	2
2.1.2 违反罚金.....	2
2.1.3 书籍信息.....	3
2.1.4 书籍库存.....	3
2.1.5 基础数据管理.....	4
2.1.6 书籍评论.....	4
2.1.7 书籍收藏.....	5
2.1.8 书籍借阅.....	5
2.1.9 公告信息.....	5
2.1.10 留言板.....	6
2.1.11 轮播图.....	6
2.2. 表关系图.....	7
2.3. E-R 模型.....	8
2.3.1 管理员端.....	8
2.3.2 普通用户端.....	8
2.3.3 总 E-R 图.....	9
2.4. 创建视图和表.....	9
2.4.1 建表语句.....	9
2.4.2 建视图语句.....	14
2.5. 触发器.....	15
2.6. 存储结构.....	15
2.6.1 存储过程.....	15
2.6.2 存储函数.....	17
3. 系统总体分析.....	17
3.1. 主要内容.....	17
3.2. 总体设计原则.....	18

---

3.3. 开发环境.....	18
3.4. 开发工具.....	18
3.4.1 IntelliJ IDEA .....	18
3.4.2 MySQL .....	18
3.4.3 Tomcat .....	18
3.5. 系统功能结构图.....	19
3.5.1 管理员端.....	19
3.5.2 普通用户端.....	20
4. 实现过程及运行结果.....	20
4.1. 管理员端.....	20
4.1.1 管理员登录.....	20
4.1.2 管理员修改信息.....	21
4.1.3 管理员管理.....	21
4.1.4 用户管理.....	21
4.1.5 书籍管理.....	22
4.1.6 书籍借阅管理.....	22
4.1.7 书籍出入库管理.....	23
4.1.8 罚金管理.....	23
4.1.9 公告管理.....	23
4.1.10 书籍类型管理.....	24
4.1.11 书架类型管理.....	24
4.1.12 留言板管理.....	25
4.1.13 公告信息管理.....	25
4.1.14 轮播图管理.....	25
4.2. 普通用户端.....	26
4.2.1 书籍查看.....	26
4.2.2 留言板.....	26
4.2.3 公告展示.....	26
4.2.4 书籍借阅.....	27
4.2.5 用户个人信息管理.....	27
5. 课程设计心得总结.....	28

---

## 1. 作品设计开发的背景及意义

### 1.1. 背景

传统图书管理方式的不足：传统的图书管理方式往往依赖于手工记录、查询和统计，这种方式效率低下且易出错。随着图书数量和借阅次数的增加，这种管理方式越来越难以满足现代图书馆的管理需求。

信息化时代的需求：随着社会信息化进程的加速，各行各业都在寻求利用信息技术提高工作效率。图书馆作为重要的文献信息中心，也需要通过信息化手段来提升服务质量和效率。

用户需求的变化：随着读者对图书馆服务需求的日益多样化，图书馆需要提供更加便捷、高效的图书借阅和查询服务。通过图书管理系统网站，读者可以随时随地查询和借阅图书，大大提高了服务的便捷性和效率。

数据库课程学习的需要：数据库课程设计是数据库原理等相关课程的重要实践环节。通过实现图书管理系统网站，学生可以将所学的理论知识应用于实际项目中，加深对数据库设计、数据库管理和数据库应用等方面的理解，提高解决实际问题的能力。

### 1.2. 目的

本网站，旨在通过小组合作完成课程设计的方式实现一个图书管理系统的网站，深入理解并应用数据库设计的相关理论，如关系数据理论，按照数据库设计步骤完成完整的数据库设计，包括需求分析、概念设计、逻辑设计、物理设计和实施等。经过图书管理的实际流程，包括图书的编目、借阅、归还、查询等，在对图书管理业务的理解基础上实现对数据库的增、删、改、查等操作，以及触发器、储存过程的应用。同时，提高成员的编程技能，熟悉编程语言，掌握数据库编程技术。除此之外，网站实现需要分工合作，共同完成任务，从而可达到培养成员团队协作能力和沟通能力的目的。

### 1.3. 意义

理论知识的实践应用：本实现图书管理系统网站，可以将所学的数据库理论知识如关系数据模型、SQL 语言、数据库设计原则等应用到实际项目中，从而加深对理论知识的理解与掌握。

提升问题解决能力：在实验过程中，会遇到各种实际问题，如性能优化、安全性问题、用户友好性设计等。通过解决这些问题，学生可以锻炼和提高自己的问题分析与解决能力。

掌握现代信息技术：本图书管理系统网站是一个典型的信息系统应用，它涉及了数据库技术、Web 开发技术、网络通信技术等现代信息技术。通过实验，可以掌握这些技术的基本原理和应用方法，为未来的学习和工作打下坚实的基础。

## 2. 数据库设计

### 2.1. 数据库需求分析

#### 2.1.1 注册登录


功能描述：

用户可以注册一个账户，这个账户实际效力相当于一张借书卡，用户可以通过账号进行图书借阅。用户通过登录名和密码登录系统后，系统将匹配用户相关信息，根据用户角色和权限进入正常界面或者后台管理界面。其中，管理员可以增加、查询、删除和修改普通用户和管理员的账号。用户和管理员都可以通过个人中心界面，进行对自己的账号信息进行查看和修改等操作。

1) 用户表(索引 id, 账号 account, 密码 password, 昵称 name, 头像 picture, 电话 phone, 身份证号 identification, 邮箱 email, 性别 sex, 状态 status, 更新时间 update\_time, 身份 role) 即命为 user 表。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
account	varchar	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>		账号
password	varchar	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>		密码
name	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>		昵称
picture	varchar	150		<input type="checkbox"/>	<input type="checkbox"/>		头像
phone	char	11		<input type="checkbox"/>	<input type="checkbox"/>		电话
identification	char	18		<input type="checkbox"/>	<input type="checkbox"/>		身份证号
email	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		邮箱
sex	int			<input type="checkbox"/>	<input type="checkbox"/>		性别
status	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		状态
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间
role	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		身份

2) 身份标记表(索引 id, 用户索引 user\_id, 标记 token, 创建时间 create\_time, 更新时间 update\_time) 即命为 token 表, 其中 user\_id 是 token 表的外键, 它引用了 user 表的主键 id。


名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
user_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户id
token	varchar	200		<input type="checkbox"/>	<input type="checkbox"/>		
create_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

#### 2.1.2 违反罚金

功能描述：

由于种种原因，用户将会因为违反相关规定而受到处罚，用户可以看到处罚以及相关原因；管理员可以对罚金进行查看与删除的操作。

1) 罚金表(索引 id, 用户索引 user\_id, 金额 value, 原因 reason, 处罚时间 create\_time) 即命为 penalty 表, 其中 user\_id 是 penalty 表的外键, 它引用了 user 表的主键 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
user_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户id
value	decimal	5	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>		金额
reason	int			<input type="checkbox"/>	<input type="checkbox"/>		罚款原因
create_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		处罚时间

### 2.1.3 书籍信息

功能描述：

用户可以对书籍进行查看，可以获取书籍名称、作者、出版社等相关信息；管理员可以对书籍相关信息进行修改，包括书籍名称、作者、出版社等信息，还可以新增书籍以及删除书籍，从而进行对书籍的管理。

1) 书籍表(索引 id, 书籍编号 uuid, 名称 name, 作者 author, 出版社 publisher, 图片 picture, 书籍类型 type\_book, 书架类型 type\_shelf, 地址 address, 数量 cnt, 点击数量 click, 介绍 introduction, 状态 status, 更新时间 update\_time) 即命为 book 表, 其中 type\_book、type\_shelf 是 book 表的外键, 它分别引用了 book\_type 表的主键 id 和 shelf\_type 表的主键 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
uuid	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		
name	varchar	200		<input checked="" type="checkbox"/>	<input type="checkbox"/>		名称
author	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		作者
publisher	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		出版社
picture	varchar	150		<input type="checkbox"/>	<input type="checkbox"/>		图片
type_book	int			<input type="checkbox"/>	<input type="checkbox"/>		书籍类型
type_shelf	int			<input type="checkbox"/>	<input type="checkbox"/>		书架类型
address	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>		地址
cnt	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		数量
click	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		点击次数
introduction	text			<input type="checkbox"/>	<input type="checkbox"/>		介绍
status	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		状态
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

### 2.1.4 书籍库存

功能描述：

管理员在此界面可以看到当前书籍的库存以及借阅情况，在相关界面可以对书籍进行出入库操作。

1) 库存更新表(索引 id, 出入库编号 uuid, 名称 name, 类型 type, 备注 introduction, 时间 time, 书籍索引 book\_id, 数量 number) 即命为 storage\_change 表, 其中 book\_id 是 storage\_change 表的外键, 它分别引用了 book 表的主键 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
uuid	varchar	200		<input checked="" type="checkbox"/>	<input type="checkbox"/>		出入库编号
name	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>		名称
type	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		类型
introduction	text			<input type="checkbox"/>	<input type="checkbox"/>		备注
time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		时间
book_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		书籍id
number	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		本数

2) 库存表(索引 id, 书籍索引 book\_id, 数量 num, 备注 introduction, 更新时间 update\_time) 即命为 inventory 表, 其中 book\_id 是 inventory 表的外键, 它

引用了 book 表的主键 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
book_id	int			<input type="checkbox"/>	<input type="checkbox"/>		书籍id
num	int			<input type="checkbox"/>	<input type="checkbox"/>		盘点数量
introduction	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>		备注
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

## 2.1.5 基础数据管理


功能描述：

管理员可对公告类型、书籍类型、书籍所在书架进行增加、删除、修改、查看等业务管理。

1) 公告类型表（索引 id，类型名称 name，更新时间 update\_time）即命为 notice\_type 表。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
name	varchar	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>		姓名
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

2) 书籍类型表（索引 id，类型名称 name，更新时间 update\_time）即命为 book\_type 表。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
name	varchar	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>		名称
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

3) 书架类型表（索引 id，类型名称 name，更新时间 update\_time）即命为 shelf\_type 表。


名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
name	varchar	50		<input checked="" type="checkbox"/>	<input type="checkbox"/>		名称
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

## 2.1.6 书籍评论

功能描述：

用户可以对选定的书籍进行评价；管理员可以针对用户提出的书籍评价进行回复、删除等操作。

1) 书籍评论表（索引 id，书籍索引 book\_id，用户索引 user\_id，评论内容 comment，创建时间 create\_time，回复 reply）即命为 book\_comment 表，其中 book\_id 和 user\_id 是 book\_comment 表的外键，它分别引用了 book 表的主键 id 和 user 表的主键 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
book_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		书籍id
user_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户id
comment	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>		内容
create_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		评论时间
reply	int			<input type="checkbox"/>	<input type="checkbox"/>		回复

## 2.1.7 书籍收藏

功能描述：

用户可以再书籍界面对需要的书籍进行收藏，可以再个人界面查看并管理已经收藏了的书籍；管理员可对书籍收藏情况进行增加、删除、修改、查看等业务管理。

1)收集表（索引 id，书籍索引 book\_id，用户 user\_id，更新时间 update\_time）即命为 collection 表，其中 book\_id 和 user\_id 是 collection 表的外键，它分别引用了 book 表的主键 id 和 user 表的主键 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
book_id	int			<input type="checkbox"/>	<input type="checkbox"/>		书籍id
user_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户id
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

## 2.1.8 书籍借阅


功能描述：

用户可以再图书界面进行借阅和归还图书；管理员可对书籍借阅情况进行增加、删除、修改、查看等业务管理。

1) 借阅表（索引 id，书籍索引 book\_id，用户 user\_id，开始时间 start\_time，天数 day，状态 status，存在与否 exist，更新时间 update\_time）borrow 表，即命为 borrow 表，其中 book\_id、user\_id 和 status 是 borrow 表的外键，它分别引用了 book 表的主键 id 和 user 表的主键 id 以及 status 表的 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
book_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		书籍id
user_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户id
start_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		开始时间
day	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		天数
status	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		状态
exist	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否被删除
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

2) 借阅状态表（索引 id，名称 name，更新时间 update\_time）即命为 status 表

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
name	varchar	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>		名称
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间

## 2.1.9 公告信息


功能描述：

用户可以主要界面查看公告管理员可以再后台对公告进行管理，可以修改，新增和删除公告。

1)公告表（索引 id，标题 title，类型 type，图片 picture，更新时间 update\_time，内容 introduction，用户索引 user\_id）即命为 notice 表，其中 user\_id 和 type 是



notice 表的外键，它分别引用了 user 表的主键 id 以及 notice\_type 表的主键 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
title	varchar	200		<input checked="" type="checkbox"/>	<input type="checkbox"/>		标题
type	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		类型
picture	varchar	200		<input type="checkbox"/>	<input type="checkbox"/>		图片
update_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		更新时间
introduction	text			<input type="checkbox"/>	<input type="checkbox"/>		内容
user_id	int			<input type="checkbox"/>	<input type="checkbox"/>		发布者id

## 2.1.10 留言板

功能描述：

用户可以在留言板界面进行留言，发表自己的想法；管理员可以在后台对选定的留言进行删除操作。

1) 留言表（索引 id，主题 theme，用户索引 user\_id，内容 introduction，创建时间 create\_time）即命为 board 表，其中 user\_id 是 board 表的外键，它引用了 user 表的主键 id。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
theme	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>		主题
user_id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户id
introduction	text			<input type="checkbox"/>	<input type="checkbox"/>		内容
create_time	timestamp			<input checked="" type="checkbox"/>	<input type="checkbox"/>		创建时间

## 2.1.11 轮播图

功能描述：

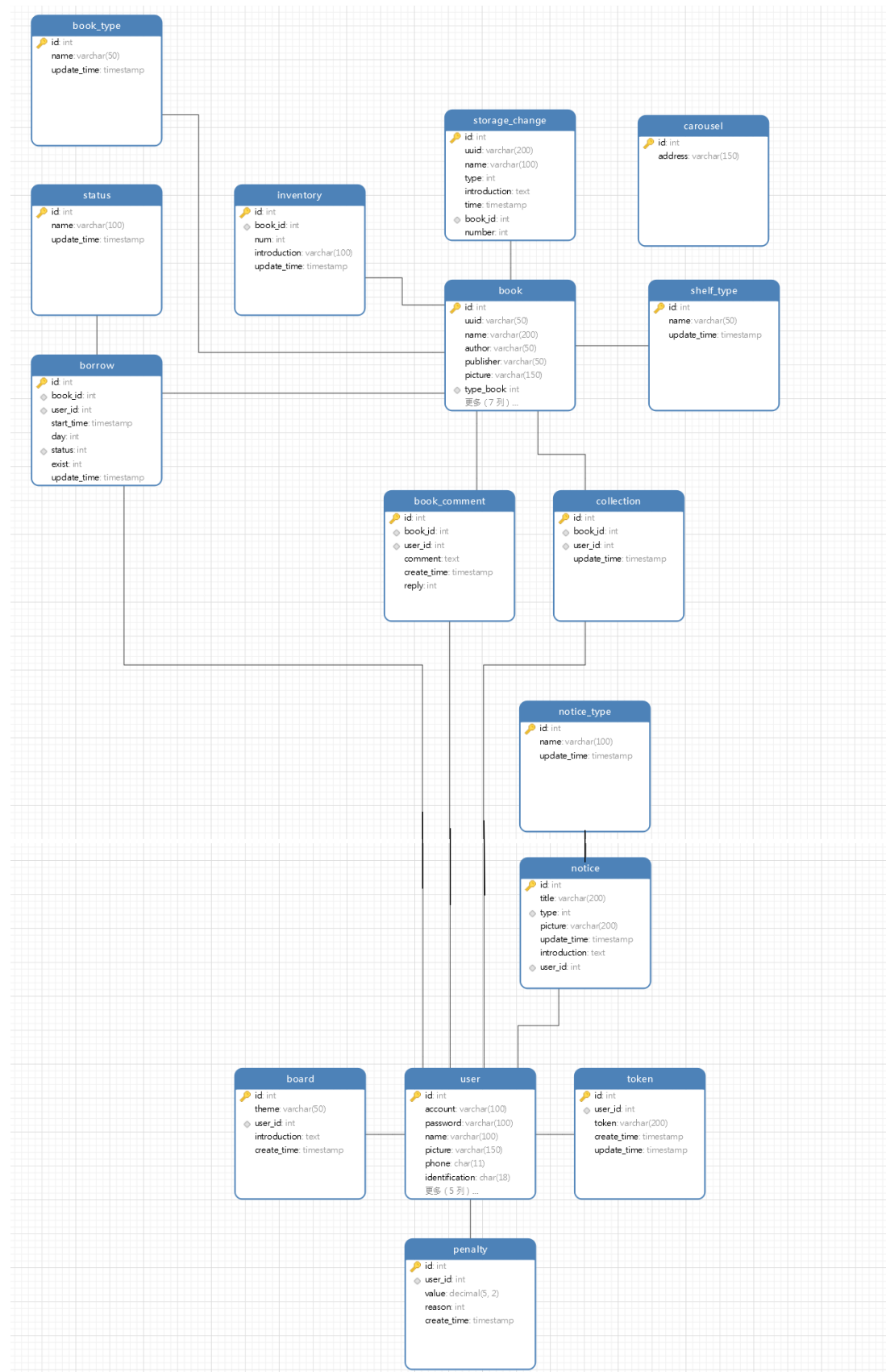
管理员在后台界面可以新增删除和修改轮播图；用户可以在主界面观看到轮播图。

1) 轮播图表（索引 id，图片位置 address），即命为 carousel 表。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	主键
address	varchar	150		<input checked="" type="checkbox"/>	<input type="checkbox"/>		存放地址

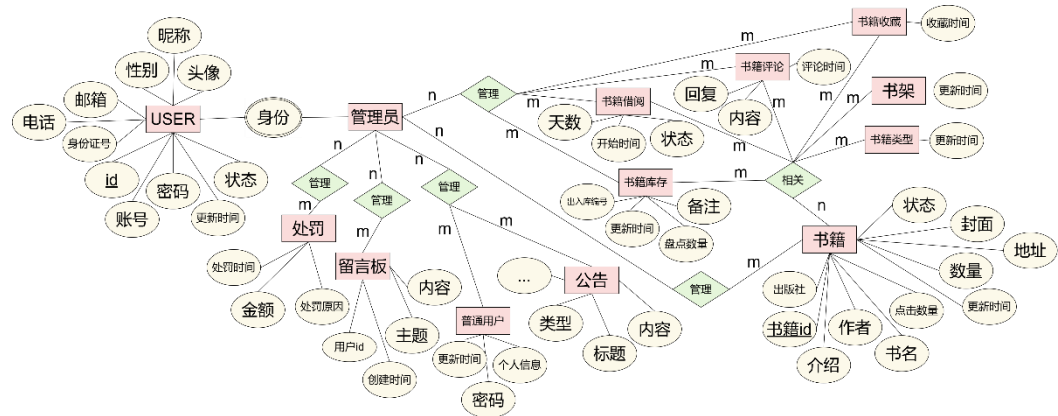
注：带下划线表示主键

## 2.2. 表关系图

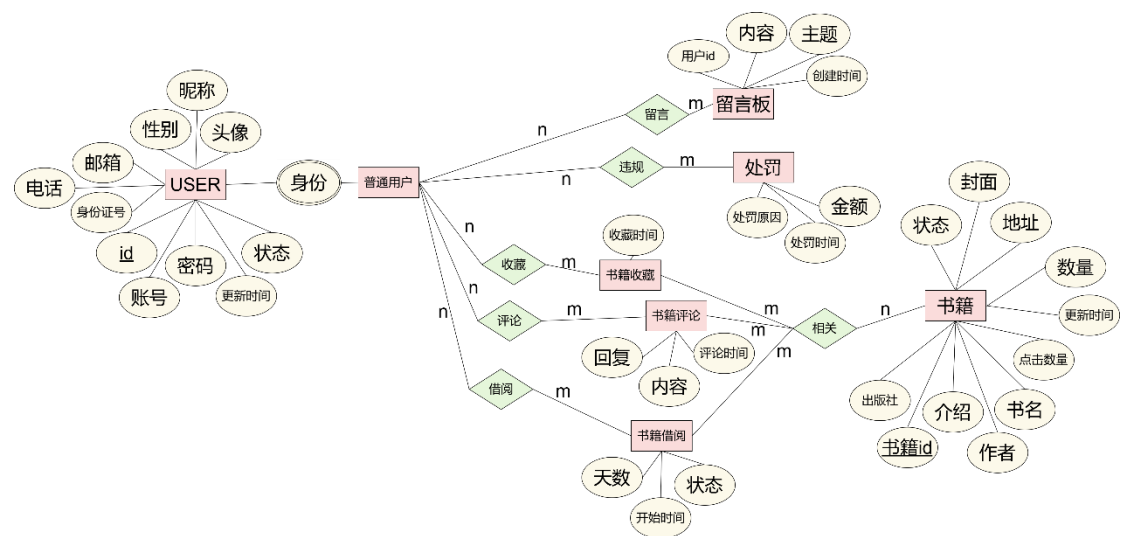


## 2.3. E-R 模型

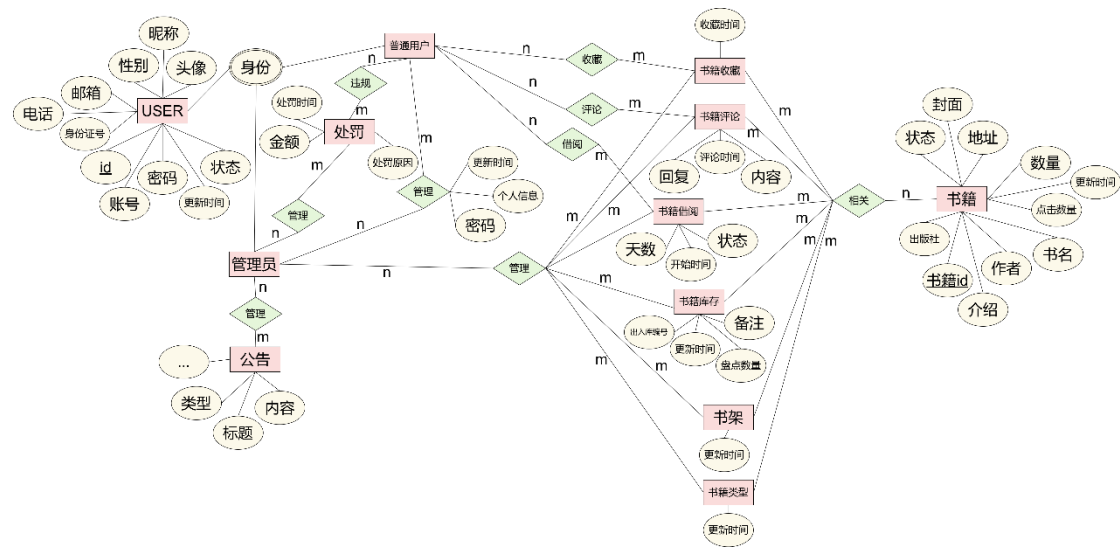
### 2.3.1 管理员端



### 2.3.2 普通用户端



### 2.3.3 总 E-R 图



## 2.4. 创建视图和表

### 2.4.1 建表语句

-- board（公告表） 结构

drop table if exists board;

create table board (

id int not null auto\_increment comment '主键',

theme varchar(50) comment '主题',

user\_id int not null comment '用户 id',

introduction text comment '内容',

create\_time timestamp not null default current\_timestamp comment '创建时间',

primary key (id),

constraint board\_userid\_fk foreign key (user\_id) references user (id)

) comment = '留言板';

-- book（书籍表） 结构

drop table if exists book;

create table book (

id int not null auto\_increment comment '主键',

uuid varchar(50),

name varchar(200) not null comment '名称',

author varchar(50) comment '作者',

publisher varchar(50) comment '出版社',

picture varchar(150) comment '图片',

type\_book int null comment '书籍类型',

---

```
type_shelf int null comment '书架类型',
address varchar(100) comment '地址',
cnt int not null comment '数量',
click int not null default 0 comment '点击次数',
introduction text comment '介绍',
status int not null default 1 comment '状态',
update_time timestamp not null default current_timestamp comment '更新时间',
primary key (id),
constraint book_typebook_fk foreign key (type_book) references book_type (id),
constraint book_typeshelf_fk foreign key (type_shelf) references shelf_type (id)
) comment = '书籍';
```

```
-- book_comment (书籍评论表) 结构
drop table if exists book_comment;
create table book_comment (
    id int not null auto_increment comment '主键',
    book_id int not null comment '书籍 id',
    user_id int not null comment '用户 id',
    comment text not null comment '内容',
    create_time timestamp not null default current_timestamp comment '评论时间',
    reply int null comment '回复',
    primary key (id),
    constraint book_comment_bookid_fk foreign key (book_id) references book (id),
    constraint book_comment_userid_fk foreign key (user_id) references user (id)
) comment = '书籍评论';
```

```
-- book_type (书籍类型表) 结构
drop table if exists book_type;
create table book_type (
    id int not null auto_increment comment '主键',
    name varchar(50) not null comment '名称',
    update_time timestamp not null default current_timestamp comment '更新时间',
    primary key (id)
) comment = '书籍类型';
```

```
-- borrow (借阅表) 结构
drop table if exists borrow;
create table borrow (
    id int not null auto_increment comment '主键',
    book_id int not null comment '书籍 id',
    user_id int not null comment '用户 id',
    start_time timestamp not null default current_timestamp comment '开始时间',
    day int not null comment '天数',
    status int not null default 1 comment '状态',
```

---

```
    exist int not null default 1 comment '是否被删除',
    update_time timestamp not null default current_timestamp comment '更新时间',
    primary key (id),
    constraint borrow_bookid_fk foreign key (book_id) references book (id),
    constraint borrow_status_fk foreign key (status) references status (id),
    constraint borrow_userid_fk foreign key (user_id) references user (id)
) comment = '借阅';
```

```
-- carousel（轮播图表） 结构
drop table if exists carousel;
create table carousel (
    id int not null auto_increment comment '主键',
    address varchar(150) not null comment '存放地址',
    primary key (id)
) comment = '轮播图';
```

```
-- collection（收藏表） 结构
drop table if exists collection;
create table collection (
    id int not null auto_increment comment '主键',
    book_id int null comment '书籍 id',
    user_id int not null comment '用户 id',
    update_time timestamp not null default current_timestamp comment '更新时间',
    primary key (id),
    constraint collection_bookid_fk foreign key (book_id) references book (id),
    constraint collection_userid_fk foreign key (user_id) references user (id)
) comment = '收藏';
```

```
-- inventory（盘点表） 结构
drop table if exists inventory;
create table inventory (
    id int not null auto_increment comment '主键',
    book_id int null comment '书籍 id',
    num int null comment '盘点数量',
    introduction varchar(100) comment '备注',
    update_time timestamp not null default current_timestamp comment '更新时间',
    primary key (id),
    index inventory_bookid_fk(book_id),
    constraint inventory_bookid_fk foreign key (book_id) references book (id)
) comment = '盘点';
```

```
-- notice（公告表） 结构
drop table if exists notice;
create table notice (
```

---

```

    id int not null auto_increment comment '主键',
    title varchar(200) not null comment '标题',
    type int not null comment '类型',
    picture varchar(200) comment '图片',
    update_time timestamp not null default current_timestamp comment '更新时间',
    introduction text comment '内容',
    user_id int null comment '发布者 id',
    primary key (id),
    index notice_type_fk(type),
    index notice_userid_fk(user_id),
    constraint notice_type_fk foreign key (type) references notice_type (id),
    constraint notice_userid_fk foreign key (user_id) references user (id)
) comment = '公告';

-- notice_type (公告类型表) 结构
drop table if exists notice_type;
create table notice_type (
    id int not null auto_increment comment '主键',
    name varchar(100) not null comment '姓名',
    update_time timestamp not null default current_timestamp comment '更新时间',
    primary key (id)
) comment = '公告类型';

-- penalty (罚金表) 结构
drop table if exists penalty;
create table penalty (
    id int not null auto_increment comment '主键',
    user_id int not null comment '用户 id',
    value decimal(5, 2) not null comment '金额',
    reason int null comment '罚款原因',
    create_time timestamp not null default current_timestamp comment '处罚时间',
    primary key (id),
    index penalty_userid_fk(user_id),
    constraint penalty_userid_fk foreign key (user_id) references user (id)
) comment = '罚金';

-- shelf_type (书架类型) 结构
drop table if exists shelf_type;
create table shelf_type (
    id int not null auto_increment comment '主键',
    name varchar(50) not null comment '名称',
    update_time timestamp not null default current_timestamp comment '更新时间',
    primary key (id)
) comment = '书架类型';

```

---

```

-- status（借阅状态表） 结构
drop table if exists status;
create table status (
    id int not null auto_increment comment '主键',
    name varchar(100) not null comment '名称',
    update_time timestamp not null default current_timestamp comment '更新时间',
    primary key (id)
) comment = '借阅状态';

-- storage_change（出入库表） 结构
drop table if exists storage_change;
create table storage_change (
    id int not null auto_increment comment '主键',
    uuid varchar(200) not null comment '出入库编号',
    name varchar(100) comment '名称',
    type int not null comment '类型',
    introduction text comment '备注',
    time timestamp not null default current_timestamp comment '时间',
    book_id int not null comment '书籍 id',
    number int not null comment '本数',
    primary key (id),
    constraint storage_change_bookid_fk foreign key (book_id) references book (id)
) comment = '出入库';

-- token 结构
drop table if exists token;
create table token (
    id int not null auto_increment comment '主键',
    user_id int not null comment '用户 id',
    token varchar(200),
    create_time timestamp not null default current_timestamp comment '创建时间',
    update_time timestamp not null default current_timestamp comment '更新时间',
    primary key (id),
    constraint token_userid_fk foreign key (user_id) references user (id)
);

-- user（用户表） 结构
drop table if exists user;
create table user (
    id int not null auto_increment comment '主键',
    account varchar(100) not null comment '账号',
    password varchar(100) not null comment '密码',
    name varchar(100) comment '昵称',

```



---

```
    picture varchar(150) comment '头像',
    phone char(11) comment '电话',
    identification char(18) comment '身份证号',
    email varchar(20) comment '邮箱',
    sex int null comment '性别',
    status int not null default 1 comment '状态',
    update_time timestamp not null default current_timestamp comment '更新时间',
    role int not null comment '身份',
    primary key (id)
) comment = '用户';
```

## 2.4.2 建视图语句

-- 视图：书籍借阅记录

```
select
    book.id,
    book.name,
    book.picture,
    book.type_book,
    borrow.id,
    borrow.book_id,
    borrow.start_time,
    borrow.user_id,
    borrow.day,
    borrow.status
from book inner join borrow
    on book.id = borrow.book_id
where book.id = borrow.book_id;
```

-- 视图：用户可视书籍表

```
select
    book.id,
    book.name,
    book.author,
    book.publisher,
    book.picture,
    book.type_book,
    book.type_shelf,
    book.introduction
from book;
```

---

## 2.5. 触发器

```
-- 触发器：借书后更新书籍库存和状态
drop trigger if exists trigger1;
delimiter //
create trigger trigger1 after insert on borrow
for each row
begin
    update book
    set cnt = cnt - 1
    where id = new.book_id;
    if (select cnt from book where id = new.book_id) = 0 then
        update book
        set status = 2
        where id = new.book_id;
    else
        update book
        set status = 1
        where id = new.book_id;
    end if;
end
//
-- 触发器：还书后更新书籍库存和状态
begin
    call updateBorrow();
end
```

## 2.6. 存储结构

### 2.6.1 存储过程

```
-- 存储过程：借书
drop procedure if exists borrowbook;
delimiter //
create procedure borrowbook(userid int, bookid int, day int)
begin
    insert into borrow(book_id, user_id, day)
    values (bookid, userid, day);
end
//
-- 存储过程：还书
drop procedure if exists returnbook;
```

---

```

delimiter //
create procedure returnbook(id int)
begin
    declare bookid int;
    select book_id into bookid
    from borrow
    where borrow.id = id;
    update borrow
    set status = 2
    where borrow.id = id;
    update book
    set cnt = cnt + 1,
        status = 1
    where id = bookid;
end
//
-- 存储过程：更新借书状态
drop procedure if exists updateborrow;
delimiter //
CREATE PROCEDURE `updateborrow`()
begin
    declare done int default 0;
    declare borrow_id int;
    declare book_id int;
    declare user_id int;
    declare start_time timestamp;
    declare day int;
    declare status int;
    declare cur cursor for
        select id, book_id, user_id, start_time, day, status
        from borrow
        where shuji_yuyue_types = 1;
    declare continue handler for not found set done = 1;
    open cur;
    read_loop: loop
        fetch cur into borrow_id, book_id, user_id, start_time, day, status;

        if done then
            leave read_loop;
        end if;
        if timestampadd(day, day, start_time) < now() then
            update borrow
            set status = 102
            where id = borrow_id;

```

---

```
        end if;
    end loop;
    close cur;
end
```

## 2.6.2 存储函数

```
-- 存储函数：入出库
drop procedure if exists storagechange;
delimiter //
CREATE DEFINER=`root`@`localhost` FUNCTION `storage_change`(
    id INT,
    type INT,
    book_id INT,
    number INT
) RETURNS int
BEGIN
    IF type = 1 THEN
        UPDATE book
        SET cnt = cnt - number
        WHERE id = book_id;
    ELSE
        UPDATE book
        SET cnt = cnt + number
        WHERE id = book_id;
    END IF;

    RETURN id;
END
```

## 3. 系统总体分析

### 3.1. 主要内容

利用 Vue、SpringBoot、MyBatis 等技术开发一个具有实际意义的图书管理系统。

能实现与数据库的连接。

基本全部实现期末考试评分标准中的要求。

---

## 3.2. 总体设计原则

- (1) 开放性、可扩充性、可靠性原则
- (2) 安全性原则
- (3) 良好的用户操作界面
- (4) 实用性原则

## 3.3. 开发环境

操作系统: Windows 11  
开发语言: JAVA、Vue  
浏览器: Chrome  
服务器: Tomcat10.0

## 3.4. 开发工具

### 3.4.1 IntelliJ IDEA

IntelliJ IDEA 集成了开发 J2EE 程序的必要插件, 是 Java 编程语言开发的集成环境, IntelliJ 在业界被公认为最好的 Java 开发工具, 尤其在智能代码助手、代码自动提示、重构、JavaEE 支持、各类版本工具(git、svn 等)、JUnit、CVS 整合、代码分析、创新的 GUI 设计等方面的功能可以说是超常的。支持 HTML, CSS, PHP, MySQL, Python 等等需要的技术。

### 3.4.2 MySQL

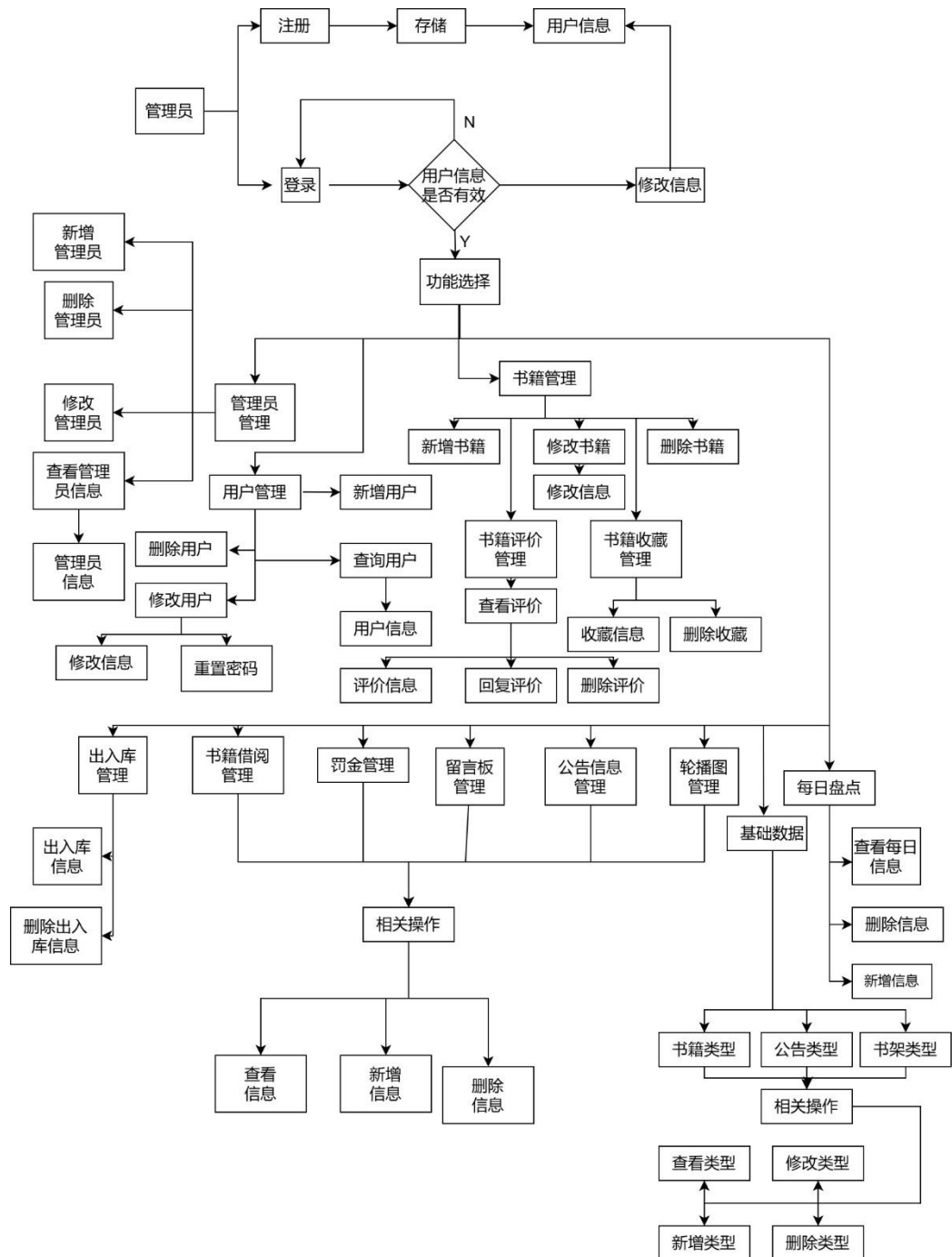
MySQL 是一个关系型数据库管理系统, 是最流行的关系型数据库管理系统之一, 在 WEB 应用方面, MySQL 是最好的 RDBMS 应用软件之一。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策, 分为社区版和商业版, 由于其体积小、速度快、总体拥有成本低, 尤其是开放源码这一特点, 一般中小型网站的开发都选择 MySQL 作为网站数据库。

### 3.4.3 Tomcat

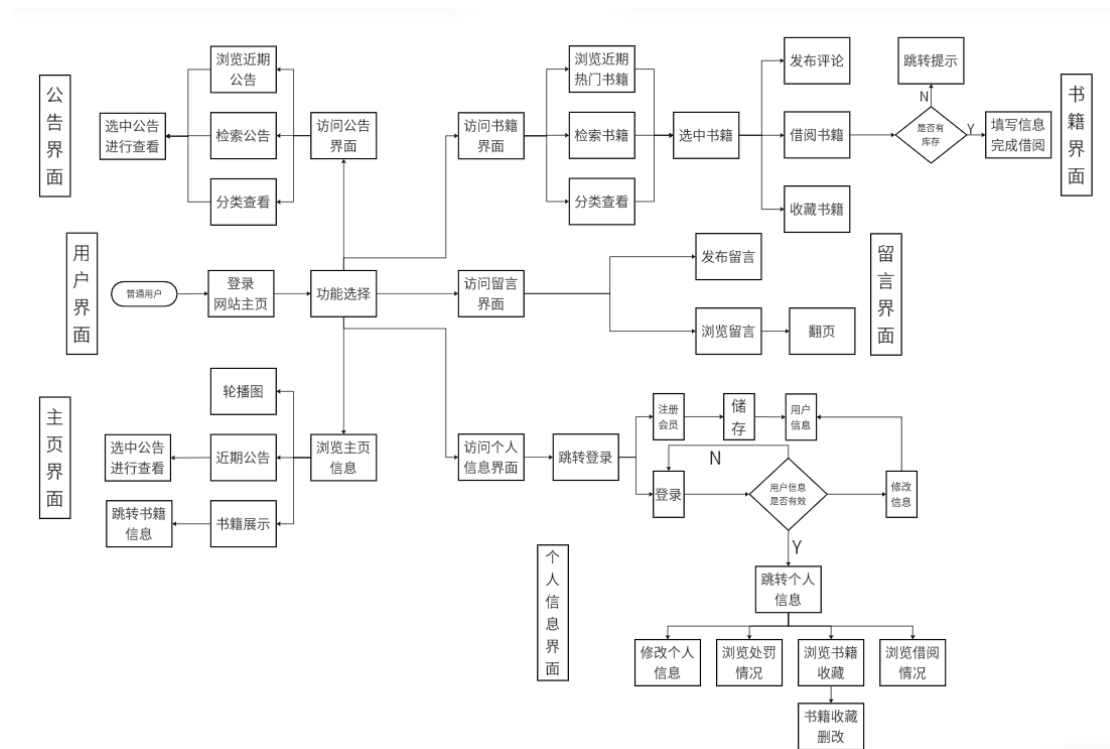
Tomcat 是 Apache 软件基金会 (Apache Software Foundation) 的 Jakarta 项目中的一个核心项目, 由 Apache、Sun 和其他一些公司及个人共同开发而成。是目前比较流行的 Web 应用服务器。

### 3.5. 系统功能结构图

#### 3.5.1 管理员端



### 3.5.2 普通用户端



## 4. 实现过程及运行结果

### 4.1. 管理员端

#### 4.1.1 管理员登录

功能实现：将管理员输入的信息与后台 mysql 中存储的信息进行比对即可



## 4.1.2 管理员修改信息

功能实现：将管理员输入的信息更新到数据库中即可

首页 / 修改密码

\* 原密码

\* 新密码

\* 确认密码

确定

首页 / 个人信息

用户名

admin

修改

## 4.1.3 管理员管理

功能实现：通过插入，更新，删除数据库管理员相关信息即可。

首页 / 管理信息

用户名

查询

+ 新增

删除

	索引	用户名	密码	角色	操作
<input type="checkbox"/>	1	admin	123456	管理员	<div><div>详情</div><div>修改</div><div>删除</div></div>

共 1 条 10条/页 < 1 > 前往 1 页

## 4.1.4 用户管理

功能实现：通过插入，更新，删除数据库用户相关信息即可

首页 / 用户

用户姓名

用户性别

性别

用户手机号

查询

+ 新增

删除

	索引	账户	用户姓名	头像	用户手机号	用户身份证号	邮箱	性别	操作
<input type="checkbox"/>	1	444	cwr		12345678910	11111122223333444	4@qq.com	女	<div><div>详情</div><div>修改</div><div>删除</div><div>重置密码</div></div>
<input type="checkbox"/>	2	333	ly		17703786903	340702200311197654	3@qq.com	男	<div><div>详情</div><div>修改</div><div>删除</div><div>重置密码</div></div>



### 4.1.5 书籍管理

功能分析：通过插入，更新，删除数据库书籍相关信息即可。

首页 / 书籍

书籍编号

书籍名称

书籍类型

所在书架

查询

+ 新增

删除

报表

导出

索引	书籍编号	书籍名称	书籍作者	出版社	书籍照片	书籍类型	所在书架	详细地址	书籍库存	热度	
<input type="checkbox"/>	1	1681111069976	自然哲学的数学原理	艾萨克·牛顿	重庆出版社		天文学、地球科学	东湖校区图书馆	1012	439	<div>详情</div>
<input type="checkbox"/>	2	1681111069996	国富论	亚当斯密	上海三联书店		政治、法律	东湖校区图书馆	1015	137	<div>详情</div>
<input type="checkbox"/>	3	1681111070014	枪炮、病菌与钢铁：人类社会的命运	贾雷德·戴蒙德	上海译文出版社		自然科学总论	东湖校区图书馆	1011	154	<div>详情</div>

首页 / 书籍

\* 书籍编号

1681111069976

\* 书籍名称

自然哲学的数学原理

\* 书籍作者

艾萨克·牛顿

\* 出版社

重庆出版社

\* 书籍照片

\* 书籍类型

天文学、地球科学

\* 所在书架

天文学、地球科学

\* 详细地址

东湖校区图书馆

\* 书籍库存

1012

\* 书籍介绍

B I U G

14px

文本

标准字体

书籍介绍13

### 4.1.6 书籍借阅管理

功能实现：通过数据库查询查找到对应的书籍借阅情况，并且可以通过发送对应请求，对数据库中借阅记录进行删除操作。

首页 / 书籍借阅

借阅状态

书籍编号

书籍名称

书籍类型

用户姓名

查询

删除

索引	书籍编号	书籍名称	书籍照片	书籍类型	用户姓名	头像	用户手机号	借阅时间	借阅天数	借阅状态
<input type="checkbox"/>	1	1681111070022	斜目而视：透过通俗文化看拉康	哲学、宗教	zxy		17703786901	2024-06-26 16:41:53	1	已评价
<input type="checkbox"/>	2	1681111069976	自然哲学的数学原理	天文学、地球科学	zxy		17703786901	2024-06-01 16:28:33	1	提前还书

### 4.1.7 书籍出入库管理

功能实现：通过对数据库进行查询，并整合成 list 在网页上进行显示。还可以通过网页向服务端发送请求，来对数据库中书籍出入库数据进行管理。

首页 / 出入库

出入库名称

出入库类型

查询

删除

出库

入库

	索引	出入库流水号	出入库名称	出入库类型	添加时间	操作
<input type="checkbox"/>	1	1719378128838	出库3	出库	2024-06-26 13:02:09	<div>详情</div> <div>删除</div> <div>查看列表详情</div>
<input type="checkbox"/>	2	1681113432890	出库书籍1、2、3 各10本	出库	2024-06-25 15:16:31	<div>详情</div> <div>删除</div> <div>查看列表详情</div>

### 4.1.8 罚金管理

功能实现：通过数据库查询到对应的罚金数据库内容，可以通过网页向服务端发送请求，服务端可以根据 id 等信息来对罚金信息在数据库中删除相关信息。


首页 / 罚金信息

用户姓名

查询

新增

删除

	索引	用户姓名	头像	用户手机号	缴纳金额	添加时间	操作
<input type="checkbox"/>	1	ly		17703786903	20.87	2024-06-25 15:16:31	<div>详情</div> <div>删除</div>
<input type="checkbox"/>	2	ly		17703786903	373.48	2024-06-25 15:16:31	<div>详情</div> <div>删除</div>

### 4.1.9 公告管理

功能实现：可以通过数据库查询公告信息，也可以向服务端发送请求来新增、修改和删除公告。

公告标题

公告类型

查询

+ 新增

删除

索引	公告标题	公告类型	公告图片	添加时间	操作
1	参与调查, 改善服务	公告类型2		2023-04-10 15:17:49	<div>详情</div> <div>修改</div> <div>删除</div>
2	图书捐赠活动进行中	公告类型2		2023-04-10 15:17:49	<div>详情</div> <div>修改</div> <div>删除</div>
3	参与阅读, 赢取奖品	公告类型1		2023-04-10 15:17:49	<div>详情</div> <div>修改</div> <div>删除</div>

## 4.1.10 书籍类型管理

功能实现：可以通过数据库查询书籍类型信息，也可以向服务端发送请求来新增、修改和删除书籍类型。

书籍类型

查询

+ 新增

删除

索引	书籍类型编码	书籍类型名称	操作
1	13	天文学、地球科学	<div>详情</div> <div>修改</div> <div>删除</div>
2	12	数理科学和化学	<div>详情</div> <div>修改</div> <div>删除</div>

## 4.1.11 书架类型管理

功能实现：可以通过数据库查询书架类型信息，也可以向服务端发送请求来新增、修改和删除书架类型。

所在书架

查询

+ 新增

删除

索引	所在书架编码	所在书架名称	操作
1	13	天文学、地球科学	<div>详情</div> <div>修改</div> <div>删除</div>
2	12	数理科学和化学	<div>详情</div> <div>修改</div> <div>删除</div>

### 4.1.12 留言板管理

功能实现：可以通过数据库查询留言板信息，也可以向服务端发送请求来新增、修改和删除留言板。

首页 / 留言板

留言板标题 用户姓名 查询

+ 新增

索引	身份	姓名	手机号	头像	留言标题	发帖时间	操作
1	用户	zxy	17703786901		1	2024-06-26 16:22:06	<div>详情查看留言板回复修改删除留言</div>
2	用户	zxy	17703786901		很好用	2024-06-26 12:59:41	<div>详情查看留言板回复修改删除留言</div>

### 4.1.13 公告信息管理

功能实现：可以通过数据库查询公告信息，也可以向服务端发送请求来新增、修改和删除公告信息。

首页 / 公告信息

公告标题 公告类型 查询

+ 新增 删除




索引	公告标题	公告类型	公告图片	添加时间	操作
<input type="checkbox"/> 1	参与调查，改善服务	公告类型2		2023-04-10 15:17:49	<div>详情修改删除</div>
<input type="checkbox"/> 2	图书馆活动进行中	公告类型2		2023-04-10 15:17:49	<div>详情修改删除</div>

### 4.1.14 轮播图管理

功能管理：可以通过数据库查询轮播图信息，也可以向服务端发送请求来新增、修改和删除轮播图信息。

首页 / 轮播图

+ 新增 删除

索引	名称	图	操作
<input type="checkbox"/> 1	轮播图1		<div>详情修改删除</div>
<input type="checkbox"/> 2	轮播图2		<div>详情修改删除</div>
<input type="checkbox"/> 3	轮播图3		<div>详情修改删除</div>

共 3 条 10条/页 1 页

## 4.2. 普通用户端

### 4.2.1 书籍查看

功能实现：通过对数据库进行查询，将查询到的书籍信息整合成 list 并返回网页显示。



### 4.2.2 留言板

功能实现：通过对数据库进行查询，将查询到的书籍信息整合成 list 并返回网页显示。



### 4.2.3 公告展示

功能实现：通过对数据库查询公告信息，整合成 list 的数据并返回网页展示。



#### 4.2.4 书籍借阅

功能实现：通过对数据库进行查询，将查询到的数据信息合并成 list 并返回网页。

图书管理系统							
首页 留言板 公告信息 书籍 个人中心							
全部 已评价 已还书 已到期 提前还书 借阅中							
书籍名称	书籍类型	书籍照片	借阅时间	借阅天数	借阅状态	添加时间	操作
斜目而视：透过通俗文化看拉康	哲学、宗教		2024-06-26 16:41:53	1	已评价	2024-06-26 16:42:03	
自然哲学的数学原理	天文学、地球科学		2024-06-01 16:28:33	1	提前还书	2024-06-26 16:28:41	
国富论	政治、法律		2024-06-12 16:26:52	0	提前还书	2024-06-26 16:27:05	

#### 4.2.5 用户个人信息管理

功能实现：根据所需要的信息发送对应的网页请求，由服务端相应从数据库返回信息。通过网页的表单发送请求即可完成信息更新和改写的相关操作。

## 5. 课程设计心得总结

经过此次实现，我们小组深刻认识到理论知识与实际应用相结合的重要性。在课程设计之初，我们一直只是进行零碎的数据库作业，没有真正理解到数据库深邃的知识。通过平时实验课的练习，我们虽然对数据库的基本概念、SQL 语言等有一定的了解，但在实际操作中还是遇到了很多困难。通过不断地查阅资料、请教老师和同学，我们逐渐将理论知识与实际应用相结合，掌握了如何设计数据库结构、如何编写 SQL 语句、如何进行数据查询和更新等操作。

我们也感受到了团队合作的力量。在课程设计中，小组成员共同完成了系统的需求分析、设计、编码和测试等各个环节。在合作过程中，我们遇到了很多问题和挑战，但通过相互沟通、协作和共同努力，我们最终克服了这些困难，成功完成了系统的开发。

此外，我们对系统设计有了更加深刻的认识与理解。在设计系统时，我们需要考虑很多因素，如系统的功能需求、性能需求、安全性需求等。我们需要对系统进行全面分析，确定系统的整体架构、模块划分、数据库设计等。

总之，这次数据库课程设计实现图书管理系统网站的实践经历让我们都受益匪浅。我们不仅掌握了数据库设计、开发和管理等方面的知识和技能，还提高了自身的实际操作能力、问题解决能力和团队协作能力。我们相信这些经验和体会将对我们未来的学习和工作将产生积极的影响。