

**课 程 设 计 报 告**

|  |  |
| --- | --- |
| 院 系： | 计算机学院 |
| 专业名称： | 软件工程 |
| 课程名称： | 软件设计基础实践 |
| 学生姓名： | 04222119 张子恒 |
| 指导教师： | 崔瀚文 |
| 完成时间： | 2023年12月 |

第二题： C类题-图书信息管理系统

1. 问题的需求分析

**题目要求：**

1、 用数据库建立 1 或 2 个图书信息表。（不限使用哪种数据库）

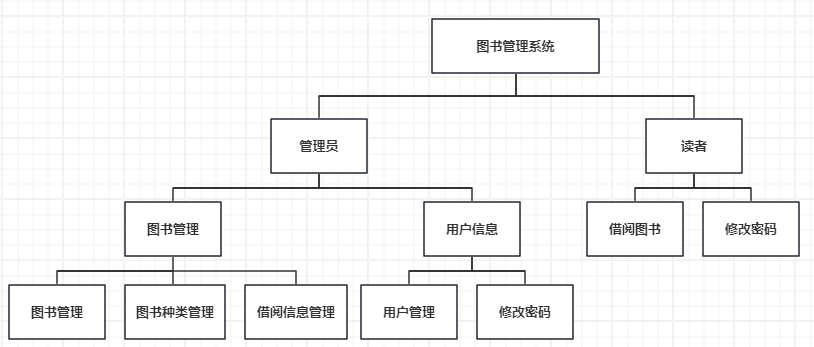
2、 能连接数据库并实现对图书的查询、增、删、改等功能。

**问题分析**：图书信息管理系统是一个用于管理图书馆内图书信息的系统。它的主要功能包括添加、删除、修改和查询图书信息，以及管理读者信息等。

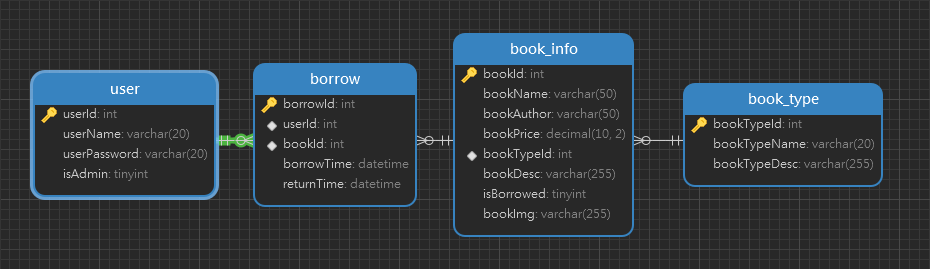
2、设计思路

1. 数据库设计：设计一个适合存储图书和读者信息的数据库，包括表的设计和关系的建立。可以使用关系型数据库如MySQL进行设计。
2. 用户界面设计：使用户能够方便地进行图书和读者信息的管理。使用前端框架技术Vue并结合图形界面库Element UI进行设计。
3. 功能模块划分：将系统的功能划分为不同的模块，如添加图书模块、删除图书模块、修改图书模块、查询图书模块和借阅管理模块等。每个模块负责处理特定的功能。
4. 数据库操作：根据用户的操作请求，执行相应的数据库操作，如插入、删除、更新和查询等。可以使用SQL语句。

图书管理系统HPIO图



数据库ER图



3、源码（详细设计）

application.properties

# 端口号  
server.port=8092  
# 上下文根  
server.servlet.context-path=/BookManager  
  
# 配置数据库  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/book\_manager?useAffectedRows=true  
spring.datasource.username=root  
spring.datasource.password=root  
  
# 配置Redis  
spring.redis.host=192.168.88.130  
spring.redis.port=6379  
spring.redis.password=123456  
  
# 上传文件大小限制  
spring.servlet.multipart.max-file-size=10MB  
spring.servlet.multipart.max-request-size=100MB  
  
# 关闭multipart，该添加会将springboot的默认的multipart取消，避免上传文件时报错  
spring.servlet.multipart.enabled=false

**以图书信息业务为例**

**package com.zh.model;**

public class BookInfo {  
  
 private Integer bookid;  
 //书名  
 private String bookname;  
 //作者  
 private String bookauthor;  
 //价格  
 private BigDecimal bookprice;  
 //类型id  
 private Integer booktypeid;  
 //类型名称  
 private String booktypename;  
 //简介  
 private String bookdesc;  
 //借书情况  
 private Byte isborrowed;  
 //图书照片  
 private String bookimg;  
  
 public Integer getBookid() {  
 return bookid;  
 }  
  
 public void setBookid(Integer bookid) {  
 this.bookid = bookid;  
 }  
  
 public String getBookname() {  
 return bookname;  
 }  
  
 public void setBookname(String bookname) {  
 this.bookname = bookname;  
 }  
  
 public String getBookauthor() {  
 return bookauthor;  
 }  
  
 public void setBookauthor(String bookauthor) {  
 this.bookauthor = bookauthor;  
 }  
  
 public BigDecimal getBookprice() {  
 return bookprice;  
 }  
  
 public void setBookprice(BigDecimal bookprice) {  
 this.bookprice = bookprice;  
 }  
  
 public Integer getBooktypeid() {  
 return booktypeid;  
 }  
  
 public void setBooktypeid(Integer booktypeid) {  
 this.booktypeid = booktypeid;  
 }  
  
 public String getBooktypename() {  
 return booktypename;  
 }  
  
 public void setBooktypename(String booktypename) {  
 this.booktypename = booktypename;  
 }  
  
 public String getBookdesc() {  
 return bookdesc;  
 }  
  
 public void setBookdesc(String bookdesc) {  
 this.bookdesc = bookdesc;  
 }  
  
 public Byte getIsborrowed() {  
 return isborrowed;  
 }  
  
 public void setIsborrowed(Byte isborrowed) {  
 this.isborrowed = isborrowed;  
 }  
  
 public String getBookimg() {  
 return bookimg;  
 }  
  
 public void setBookimg(String bookimg) {  
 this.bookimg = bookimg;  
 }  
}

**package com.zh.controller;**

@RestController  
@RequestMapping(value = "/bookInfo")  
public class BookInfoController {  
  
 @Autowired  
 BookInfoService bookInfoService;  
  
 */\*\*  
 \* 获取图书数量  
 \** ***@return*** *\*/* @GetMapping(value = "/getCount")  
 public Integer getCount(){  
 return bookInfoService.getCount();  
 }  
  
 */\*\*  
 \* 查询所有图书信息  
 \** ***@return*** *\*/* @GetMapping(value = "/queryBookInfos")  
 public List<BookInfo> queryBookInfos(){  
 return bookInfoService.queryBookInfos();  
 }  
  
 */\*\*  
 \* 分页搜索查询图书信息  
 \* params: {page, limit, bookname, bookauthor, booktypeid}  
 \** ***@param*** *params  
 \** ***@return*** *\*/* @GetMapping(value = "/queryBookInfosByPage")  
 public Map<String, Object> queryBookInfosByPage(@RequestParam Map<String, Object> params){  
 MyUtils.*parsePageParams*(params);  
 int count = bookInfoService.getSearchCount(params); // 获得总数  
 List<BookInfo> bookInfos = bookInfoService.searchBookInfosByPage(params); // 分页查询  
 return MyResult.*getListResultMap*(0, "success", count, bookInfos);  
 }  
  
 */\*\*  
 \* 添加图书信息  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* @PostMapping(value = "/addBookInfo")  
 public Integer addBookInfo(@RequestBody BookInfo bookInfo){  
 return bookInfoService.addBookInfo(bookInfo);  
 }  
  
 */\*\*  
 \* 删除图书信息  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* @DeleteMapping(value = "/deleteBookInfo")  
 public Integer deleteBookInfo(@RequestBody BookInfo bookInfo){  
 return bookInfoService.deleteBookInfo(bookInfo);  
 }  
  
 */\*\*  
 \* 删除一些图书信息  
 \** ***@param*** *bookInfos  
 \** ***@return*** *\*/* @DeleteMapping(value = "/deleteBookInfos")  
 public Integer deleteBookInfos(@RequestBody List<BookInfo> bookInfos){  
 return bookInfoService.deleteBookInfos(bookInfos);  
 }  
  
 */\*\*  
 \* 更新一些图书信息  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* @PutMapping(value = "/updateBookInfo")  
 public Integer updateBookInfo(@RequestBody BookInfo bookInfo){  
 return bookInfoService.updateBookInfo(bookInfo);  
 }  
}

**package com.zh.service;**

public interface BookInfoService {  
  
 */\*\*  
 \* 获取图书数量  
 \*  
 \** ***@return*** *\*/* Integer getCount();  
  
 */\*\*  
 \* 查询所有图书信息  
 \*  
 \** ***@return*** *\*/* List<BookInfo> queryBookInfos();  
  
 */\*\*  
 \* 根据id查询图书信息  
 \** ***@param*** *bookid  
 \** ***@return*** *\*/* BookInfo queryBookInfoById(Integer bookid);  
  
 */\*\*  
 \* 统计搜索总数  
 \** ***@param*** *params  
 \** ***@return*** *\*/* Integer getSearchCount(Map<String, Object> params);  
  
 */\*\*  
 \* 分页搜索查询图书信息  
 \*  
 \** ***@param*** *params  
 \** ***@return*** *\*/* List<BookInfo> searchBookInfosByPage(Map<String, Object> params);  
  
 */\*\*  
 \* 添加图书信息  
 \*  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* Integer addBookInfo(BookInfo bookInfo);  
  
 */\*\*  
 \* 删除图书信息  
 \*  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* Integer deleteBookInfo(BookInfo bookInfo);  
  
 */\*\*  
 \* 删除一些图书信息  
 \*  
 \** ***@param*** *bookInfos  
 \** ***@return*** *\*/* Integer deleteBookInfos(List<BookInfo> bookInfos);  
  
 */\*\*  
 \* 更新一些图书信息  
 \*  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* Integer updateBookInfo(BookInfo bookInfo);  
}

**package com.zh.service.impl;**

@Service  
public class BookInfoServiceImpl implements BookInfoService {  
  
 @Resource  
 private BookInfoMapper bookInfoMapper;  
  
 */\*\*  
 \* 获取图书数量  
 \*  
 \** ***@return*** *\*/* @Override  
 public Integer getCount() {  
 return bookInfoMapper.selectCount();  
 }  
  
 */\*\*  
 \* 查询所有图书信息  
 \*  
 \** ***@return*** *\*/* @Override  
 public List<BookInfo> queryBookInfos() {  
 return bookInfoMapper.selectAll();  
 }  
  
 */\*\*  
 \* 根据id查询图书信息  
 \*  
 \** ***@param*** *bookid  
 \** ***@return*** *\*/* @Override  
 public BookInfo queryBookInfoById(Integer bookid) {  
 return bookInfoMapper.selectByPrimaryKey(bookid);  
 }  
  
 */\*\*  
 \* 统计搜索总数  
 \*  
 \** ***@param*** *params  
 \** ***@return*** *\*/* @Override  
 public Integer getSearchCount(Map<String, Object> params) {  
 return bookInfoMapper.selectCountBySearch(params);  
 }  
  
 */\*\*  
 \* 分页搜索查询图书信息  
 \*  
 \** ***@param*** *params  
 \** ***@return*** *\*/* @Override  
 public List<BookInfo> searchBookInfosByPage(Map<String, Object> params) {  
 return bookInfoMapper.selectBySearch(params);  
 }  
  
 */\*\*  
 \* 添加图书信息  
 \*  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* @Override  
 public Integer addBookInfo(BookInfo bookInfo) {  
 return bookInfoMapper.insertSelective(bookInfo);  
 }  
  
 */\*\*  
 \* 删除图书信息  
 \*  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* @Override  
 public Integer deleteBookInfo(BookInfo bookInfo) {  
 int count = 0;  
 try {  
 count = bookInfoMapper.deleteByPrimaryKey(bookInfo.getBookid());  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return count;  
 }  
  
 */\*\*  
 \* 删除一些图书信息  
 \*  
 \** ***@param*** *bookInfos  
 \** ***@return*** *\*/* @Override  
 public Integer deleteBookInfos(List<BookInfo> bookInfos) {  
 int count = 0;  
 for (BookInfo bookInfo : bookInfos) {  
 count += deleteBookInfo(bookInfo);  
 }  
 return count;  
 }  
  
 */\*\*  
 \* 更新一些图书信息  
 \*  
 \** ***@param*** *bookInfo  
 \** ***@return*** *\*/* @Override  
 public Integer updateBookInfo(BookInfo bookInfo) {  
 return bookInfoMapper.updateByPrimaryKeySelective(bookInfo);  
 }  
  
}

**package com.zh.mapper;**

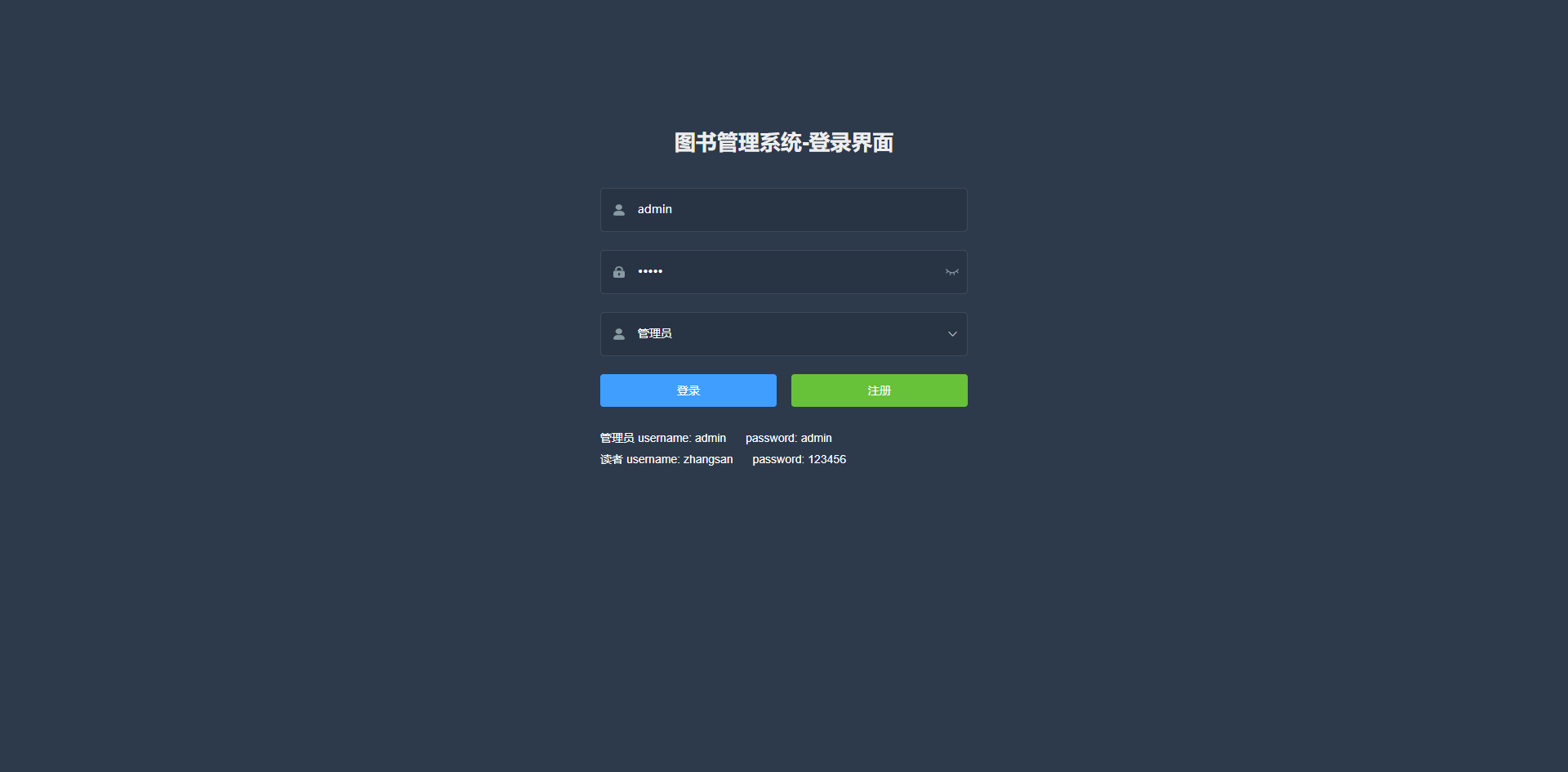
import com.zh.model.BookInfo;  
import org.apache.ibatis.annotations.Param;  
  
import java.util.List;  
import java.util.Map;  
  
public interface BookInfoMapper {  
  
 */\*\*  
 \* 获取图书数量  
 \** ***@return*** *\*/* Integer selectCount();  
  
 */\*\*  
 \* 查询所有图书信息  
 \** ***@return*** *\*/* List<BookInfo> selectAll();  
  
 */\*\*  
 \* 根据id查询图书  
 \** ***@param*** *bookid  
 \** ***@return*** *\*/* BookInfo selectByPrimaryKey(Integer bookid);  
  
 */\*\*  
 \* 统计搜索总数  
 \** ***@param*** *searchParam  
 \** ***@return*** *\*/* int selectCountBySearch(Map<String, Object> searchParam);  
  
 */\*\*  
 \*搜索查询图书信息  
 \** ***@param*** *searchParam  
 \** ***@return*** *\*/* List<BookInfo> selectBySearch(Map<String, Object> searchParam);  
  
 */\*\*  
 \* 根据id删除图书  
 \** ***@param*** *bookid  
 \** ***@return*** *\*/* int deleteByPrimaryKey(Integer bookid);  
  
 */\*\*  
 \* 添加图书  
 \** ***@param*** *record  
 \** ***@return*** *\*/* int insert(BookInfo record);  
  
 */\*\*  
 \* 添加图书-动态  
 \** ***@param*** *record  
 \** ***@return*** *\*/* int insertSelective(BookInfo record);  
  
 */\*\*  
 \* 更新图书信息-动态  
 \** ***@param*** *record  
 \** ***@return*** *\*/* int updateByPrimaryKeySelective(BookInfo record);  
  
 */\*\*  
 \* 更新图书信息  
 \** ***@param*** *record  
 \** ***@return*** *\*/* int updateByPrimaryKey(BookInfo record);  
  
 */\*\*  
 \* 分页查询图书  
 \** ***@param*** *begin  
 \** ***@param*** *size  
 \** ***@return*** *\*/* List<BookInfo> selectAllByLimit(@Param("begin") Integer begin, @Param("size") Integer size);  
  
 */\*\*  
 \* 根据种类统计图书数量  
 \** ***@param*** *map  
 \** ***@return*** *\*/* int selectCountByType(Map<String, Object> map);  
  
 */\*\*  
 \* 根据种类查询图书信息  
 \** ***@param*** *map  
 \** ***@return*** *\*/* List<BookInfo> selectByType(Map<String, Object> map);  
}

**BookInfoMapper.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="com.zh.mapper.BookInfoMapper">  
 <resultMap id="BaseResultMap" type="com.zh.model.BookInfo">  
 <id column="bookId" jdbcType="INTEGER" property="bookid" />  
 <result column="bookName" jdbcType="VARCHAR" property="bookname" />  
 <result column="bookAuthor" jdbcType="VARCHAR" property="bookauthor" />  
 <result column="bookPrice" jdbcType="DECIMAL" property="bookprice" />  
 <result column="bookTypeId" jdbcType="INTEGER" property="booktypeid" />  
 <result column="bookTypeName" jdbcType="INTEGER" property="booktypename" />  
 <result column="bookDesc" jdbcType="VARCHAR" property="bookdesc" />  
 <result column="isBorrowed" jdbcType="TINYINT" property="isborrowed" />  
 <result column="bookImg" jdbcType="TINYINT" property="bookimg" />  
 </resultMap>  
 <sql id="Base\_Column\_List">  
 bookId, bookName, bookAuthor, bookPrice, bookTypeId, bookDesc, isBorrowed, bookImg  
 </sql>  
  
 <select id="selectCount" resultType="int">  
 select count(\*) from book\_manager.book\_info  
 </select>  
  
 <select id="selectAll" resultMap="BaseResultMap">  
 select bookId, bookName, bookAuthor,bookPrice, bookTypeId, bookDesc,isBorrowed, bookImg,  
 (select bookTypeName from book\_manager.book\_type where book\_manager.book\_type.bookTypeId = book\_manager.book\_info.bookTypeId) as bookTypeName  
 from book\_manager.book\_info  
 </select>  
  
 <select id="selectByPrimaryKey" parameterType="java.lang.Integer" resultMap="BaseResultMap">  
 select  
 <include refid="Base\_Column\_List" />  
 from book\_manager.book\_info  
 where bookId = #{bookid,jdbcType=INTEGER}  
 </select>  
  
 <select id="selectCountBySearch" resultType="int">  
 select count(\*) from book\_manager.book\_info  
 <where>  
 <if test="bookname != null and bookname != '' ">and bookName like concat('%',#{bookname},'%')</if>  
 <if test="bookauthor != null and bookauthor != '' ">and bookAuthor like concat('%',#{bookauthor},'%')</if>  
 <if test="booktypeid != null and booktypeid != '' ">and bookTypeId = #{booktypeid}</if>  
 </where>  
 </select>  
  
 <select id="selectBySearch" resultMap="BaseResultMap">  
 select bookId, bookName, bookAuthor,bookPrice, bookTypeId, bookDesc,isBorrowed, bookImg,  
 (select bookTypeName from book\_manager.book\_type where book\_manager.book\_type.bookTypeId = book\_manager.book\_info.bookTypeId) as bookTypeName  
 from book\_manager.book\_info  
 <where>  
 <if test="bookname != null and bookname != '' ">and bookName like concat('%',#{bookname},'%')</if>  
 <if test="bookauthor != null and bookauthor != '' ">and bookAuthor like concat('%',#{bookauthor},'%')</if>  
 <if test="booktypeid != null and booktypeid != '' ">and bookTypeId = #{booktypeid}</if>  
 </where>  
 limit #{begin}, #{size}  
 </select>  
  
 <delete id="deleteByPrimaryKey" parameterType="java.lang.Integer">  
 delete from book\_manager.book\_info  
 where bookId = #{bookid,jdbcType=INTEGER}  
 </delete>  
  
 <insert id="insert" parameterType="com.zh.model.BookInfo">  
 insert into book\_manager.book\_info (bookId, bookName, bookAuthor,  
 bookPrice, bookTypeId, bookDesc,   
 isBorrowed, bookImg)  
 values (#{bookid,jdbcType=INTEGER}, #{bookname,jdbcType=VARCHAR}, #{bookauthor,jdbcType=VARCHAR},   
 #{bookprice,jdbcType=DECIMAL}, #{booktypeid,jdbcType=INTEGER}, #{bookdesc,jdbcType=VARCHAR},   
 #{isborrowed,jdbcType=TINYINT}, #{bookimg,jdbcType=TINYINT})  
 </insert>  
  
 <insert id="insertSelective" parameterType="com.zh.model.BookInfo">  
 insert into book\_manager.book\_info  
 <trim prefix="(" suffix=")" suffixOverrides=",">  
 <if test="bookid != null">bookId,</if>  
 <if test="bookname != null">bookName,</if>  
 <if test="bookauthor != null">bookAuthor,</if>  
 <if test="bookprice != null">bookPrice,</if>  
 <if test="booktypeid != null">bookTypeId,</if>  
 <if test="bookdesc != null">bookDesc,</if>  
 <if test="isborrowed != null">isBorrowed,</if>  
 <if test="bookimg != null">bookImg,</if>  
 </trim>  
 <trim prefix="values (" suffix=")" suffixOverrides=",">  
 <if test="bookid != null">#{bookid,jdbcType=INTEGER},</if>  
 <if test="bookname != null">#{bookname,jdbcType=VARCHAR},</if>  
 <if test="bookauthor != null">#{bookauthor,jdbcType=VARCHAR},</if>  
 <if test="bookprice != null">#{bookprice,jdbcType=DECIMAL},</if>  
 <if test="booktypeid != null">#{booktypeid,jdbcType=INTEGER},</if>  
 <if test="bookdesc != null">#{bookdesc,jdbcType=VARCHAR},</if>  
 <if test="isborrowed != null">#{isborrowed,jdbcType=TINYINT},</if>  
 <if test="bookimg != null">#{bookimg,jdbcType=TINYINT},</if>  
 </trim>  
 </insert>  
  
 <update id="updateByPrimaryKeySelective" parameterType="com.zh.model.BookInfo">  
 update book\_manager.book\_info  
 <set>  
 <if test="bookname != null">bookName = #{bookname,jdbcType=VARCHAR},</if>  
 <if test="bookauthor != null">bookAuthor = #{bookauthor,jdbcType=VARCHAR},</if>  
 <if test="bookprice != null">bookPrice = #{bookprice,jdbcType=DECIMAL},</if>  
 <if test="booktypeid != null">bookTypeId = #{booktypeid,jdbcType=INTEGER},</if>  
 <if test="bookdesc != null">bookDesc = #{bookdesc,jdbcType=VARCHAR},</if>  
 <if test="isborrowed != null">isBorrowed = #{isborrowed,jdbcType=TINYINT},</if>  
 <if test="bookimg != null">bookImg = #{bookimg,jdbcType=TINYINT},</if>  
 </set>  
 where bookId = #{bookid,jdbcType=INTEGER}  
 </update>  
  
 <update id="updateByPrimaryKey" parameterType="com.zh.model.BookInfo">  
 update book\_manager.book\_info  
 set bookName = #{bookname,jdbcType=VARCHAR},  
 bookAuthor = #{bookauthor,jdbcType=VARCHAR},  
 bookPrice = #{bookprice,jdbcType=DECIMAL},  
 bookTypeId = #{booktypeid,jdbcType=INTEGER},  
 bookDesc = #{bookdesc,jdbcType=VARCHAR},  
 isBorrowed = #{isborrowed,jdbcType=TINYINT},  
 bookImg = #{bookimg,jdbcType=TINYINT}  
 where bookId = #{bookid,jdbcType=INTEGER}  
 </update>  
  
 <select id="selectAllByLimit" resultMap="BaseResultMap">  
 select bookId, bookName, bookAuthor,bookPrice, bookTypeId, bookDesc,isBorrowed, bookImg,  
 (select bookTypeName from book\_manager.book\_type where book\_manager.book\_type.bookTypeId = book\_manager.book\_info.bookTypeId) as bookTypeName  
 from book\_manager.book\_info  
 limit #{begin}, #{size}  
 </select>  
  
  
  
  
 <select id="selectCountByType" resultType="int">  
 select count(\*) from book\_manager.book\_info  
 <where>  
 <if test="booktypeid != null and booktypeid != '' ">and bookTypeId = #{booktypeid}</if>  
 </where>  
 </select>  
  
 <select id="selectByType" resultMap="BaseResultMap">  
 select bookId, bookName, bookAuthor,bookPrice, bookTypeId, bookDesc,isBorrowed, bookImg,  
 (select bookTypeName from book\_manager.book\_type where book\_manager.book\_type.bookTypeId = book\_manager.book\_info.bookTypeId) as bookTypeName  
 from book\_manager.book\_info  
 <where>  
 <if test="booktypeid != null and booktypeid != '' ">and bookTypeId = #{booktypeid}</if>  
 </where>  
 limit #{begin}, #{size}  
 </select>  
  
</mapper>

4、程序运行、测试与分析

登录



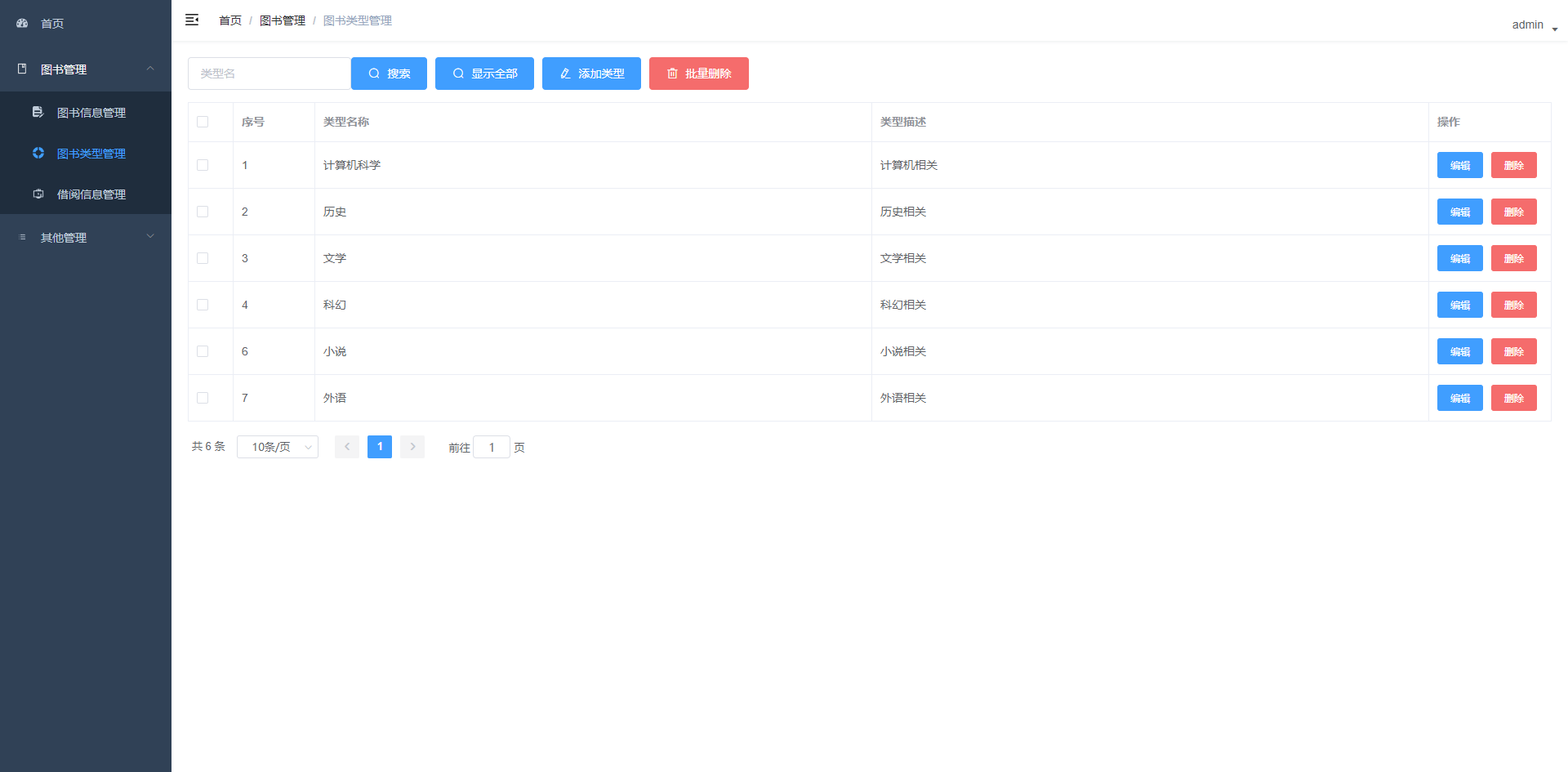
图书信息管理



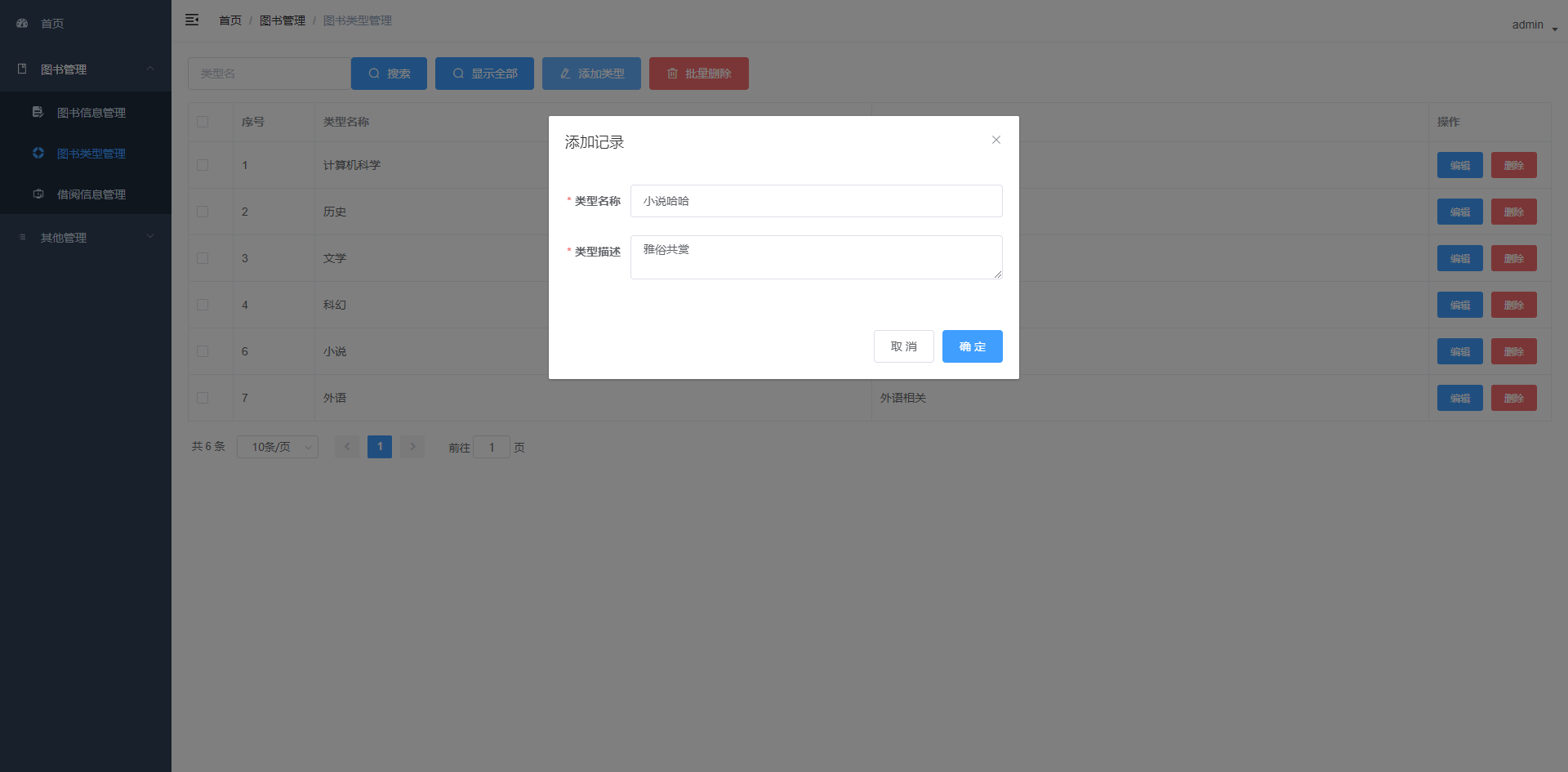
添加图书



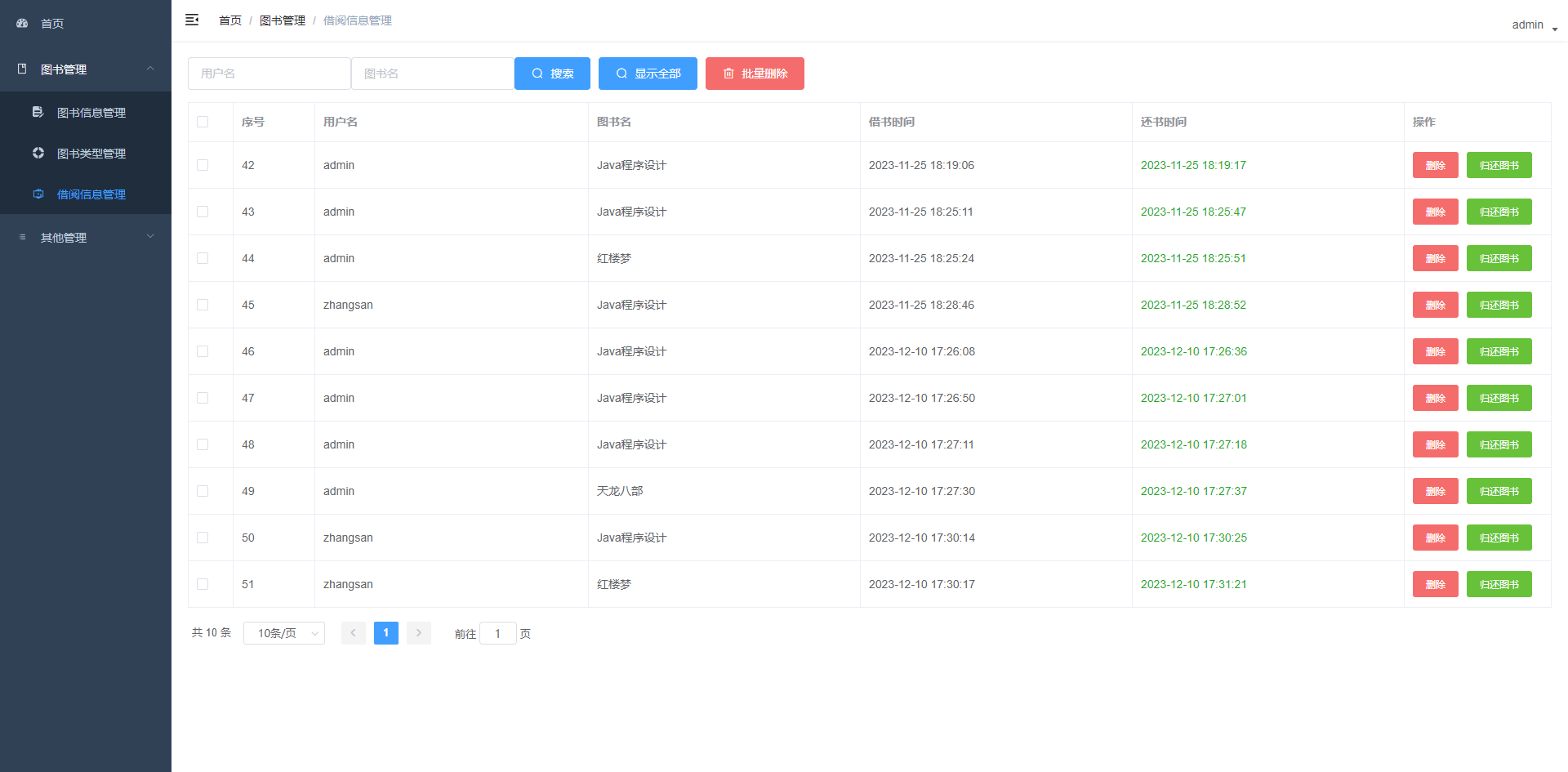
图书类型管理



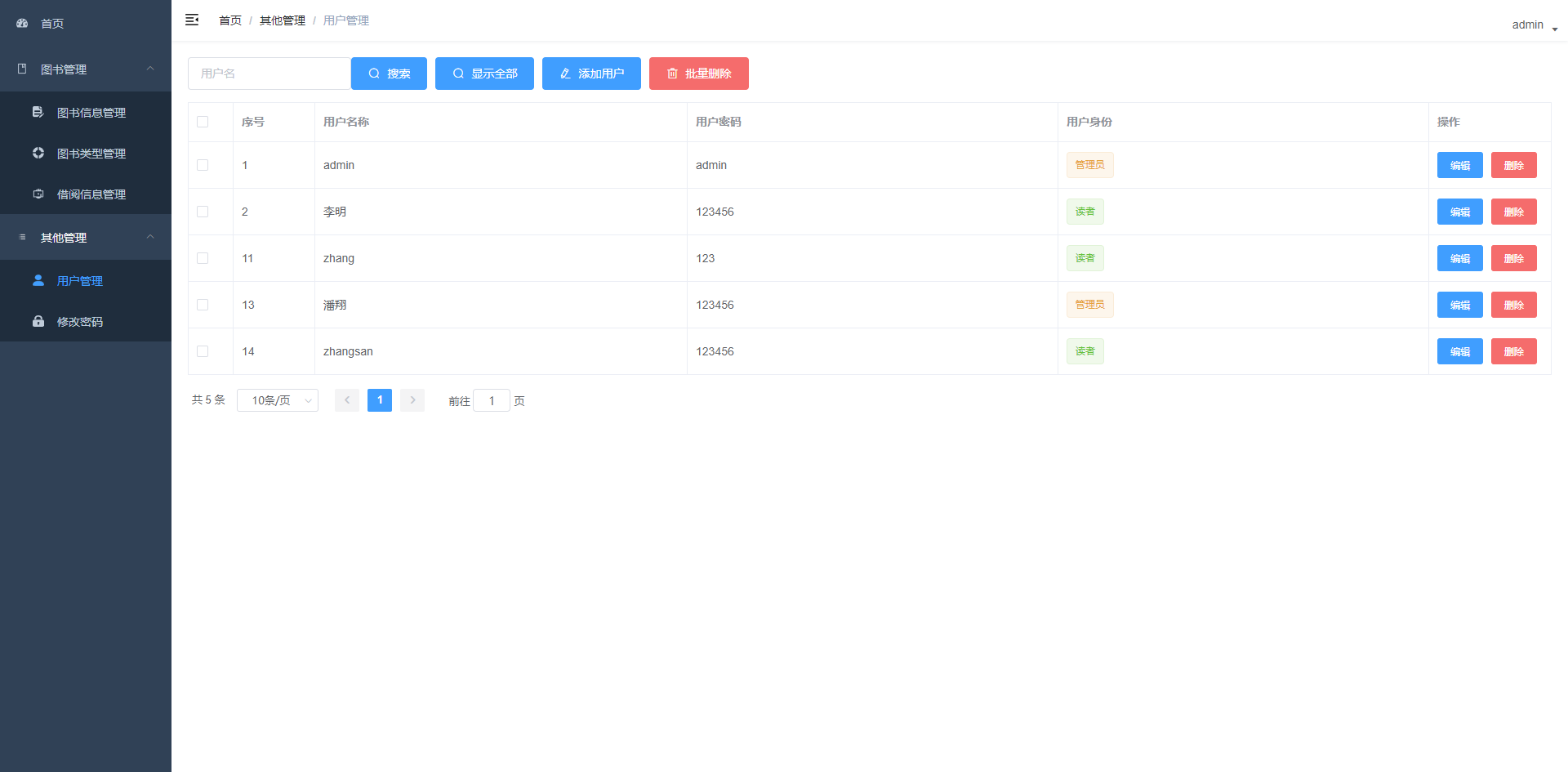
添加图书类型



借阅信息管理



用户管理



添加用户



修改密码

