



设计说明书

|  |  |
| --- | --- |
| 学院 | 计算机科学与工程学院 |
| 课程 | 面向对象程序设计综合实训 |
| 项目名称 | 图书管理系统的设计与实现 |
| 班级 | 软工（卓越）23102 |
| 组长 | 李一飞 |
| 成员 | 邹炜林、吉佳乐 |
| 时间 | 2024-2025学年第二学期 |

目 录

[1. 需求分析 1](#_Toc25008)

[1.1 管理员需求分析 1](#_Toc15447)

[1.2 用户需求分析 2](#_Toc7871)

[1.3用例规约 3](#_Toc26333)

[2. 系统设计 5](#_Toc11576)

[2.1 总体设计 5](#_Toc30710)

[2.2数据库设计 6](#_Toc4222)

[2.2.1概念结构设计 6](#_Toc9741)

[2.2.2逻辑结构设计 6](#_Toc11168)

[2.3详细设计 7](#_Toc23830)

[3. 系统实现 8](#_Toc855)

[3.1 项目结构 8](#_Toc7446)

[3.2功能实现 8](#_Toc13570)

[3.2.1管理员端功能实现 8](#_Toc23991)

[3.2.2读者端功能实现 16](#_Toc29273)

# 1. 需求分析

系统面向管理员和读者两类用户，分别实现图书管理、读者管理、借阅管理等功能。

## 1.1 管理员需求分析

通过该系统，管理员需要实现以下功能：

（1）注册：创建新的系统管理员账户。

（2）登录：管理员需登录系统后，才可操作系统。

（3）图书信息管理：管理员添加、修改、删除图书信息（书名、作者、出版社等）。

（5）读者信息管理：维护读者档案（姓名、学号/工号、借阅权限等）。

（6）库存管理：管理员监控图书库存状态，处理图书采购和报废。

（7）图书状态管理：管理员标记图书为可用、借出、维修或丢失状态。

（8）借阅管理：管理员设定借阅期限、续借次数、逾期罚款规则等。

（9）系统管理：管理员设置系统参数和运行规则。

管理员端用例图如图1.1所示。



图1.1管理员用例图

## 1.2 用户需求分析

通过该系统，读者需要实现以下功能：

（1）注册：读者输入个信信息进行注册。

（2）登录：读者需登录系统后，才可操作系统。

（3）查找：读者可以根据书名搜索图书，根据作者搜索图书，也可以按照书名、作者及出版社等信息实现图书精确搜索。

（4）用户个人信息管理：读者同过输入验证信息来查询个人信息和修改密码

（5）预约定阅管理：用户提交自己的预约信息，查看自己先前的预订的图书

（6）阅读历史和收藏：用进入特定的页面进行查看自己地阅读历史并管理自己的收藏

（7）借阅与归还：用户查看自己的借阅信息并归还

读者端用例图如图1.2所示。



图1.2读者用例图

## 1.3用例规约

1. 注册用例规约如表1.1所示。

表1.1读者注册用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 注册 |
| 用例说明 | 新读者用户进行注册 |
| 参与用户 | 读者 |
| 前置条件 | 无 |
| 基本事件流 | 读者点击注册按钮  填写新读者的个人信息  点击提交按钮  弹出对话框,提示“注册成功” |
| 其他事件流 | 若读者电话，则提示“读者电话不能为空” |
| 后置条件 | 新的住户信息成功添加到图书管理系统中 |

1. 读者登录用例规约如表1.2所示

表1.2读者登录用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 登录 |
| 用例说明 | 读者用户进行登录 |
| 参与用户 | 读者 |
| 前置条件 | 无 |
| 基本事件流 | 读者填写个人的账号密码  点击登录按钮  弹出对话框,提示“登录成功” |
| 其他事件 | 登录时提示网络异常 |
| 后置条件 | 新的住户登录信息成功添加到图书管理系统中 |

1. 读者查询图书用例规约如表1.3所示

表1.3读者查询图书用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 查询图书 |
| 用例说明 | 已经登录的读者查询图书信息 |
| 参与用户 | 读者 |
| 前置条件 | 读者已登录 |
| 基本事件流 | 用户在主系统界面的搜索框中输入书籍名称、作者、关键词等查询条件，点击搜索按钮  系统根据用户的查询条件，在数据库匹配并输出相应的结果 |
| 其他事件 | 若没有找到符合条件的书籍，提示用户“未找到相关书籍” |
| 后置条件 | 用户获取到符合条件的书籍列表信息 |

1. 读者借阅图书用例规约如表1.4所示

表1.4读者借阅图书用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 借阅图书 |
| 用例说明 | 已登录的读者借阅图书 |
| 参与用户 | 读者 |
| 前置条件 | 读者已经登录系统且查询到的图书为可借阅 |
| 基本事件流 | 读者在书籍列表中找到想要借阅的书籍，点击借阅按钮  系统弹出借阅确认对话框，显示书籍信息以及借阅期限等规则（如最长借阅天数  用户确认无误后点击确定  弹出对话框,提示“借阅成功 |
| 其他事件 | 若用户当前借阅的书籍数量已达系统规定的上限，提示用户无法继续借阅，需先归还部分书籍 |
| 后置条件 | 书籍状态更新为已借，用户的借阅记录中增加该书籍信息 |

1. 读者归还图书用例规约如表1.5所示

表1.5读者归还图书用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 归还图书 |
| 用例说明 | 已登录的读者归还图书 |
| 参与用户 | 读者 |
| 前置条件 | 用户已登录系统，且存在已借阅的书籍 |
| 基本事件流 | 用户在系统中查看自己的借阅记录，找到想要归还的书籍，点击归还按钮  系统弹出归还确认对话框，用户点击确认归还操作  用户确认后，系统将书籍状态更新为可借 |
| 其他事件 | 若归还过程中出现系统故障导致归还失败，提示用户稍后再试 |
| 后置条件 | 书籍状态更新为可借，读者借阅记录中的图书状态变为已归还 |

1. 读者预约图书用例规约如表1.6所示

表1.6读者归还图书用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 预约图书 |
| 用例说明 | 已登录的读者预约图书 |
| 参与用户 | 读者 |
| 前置条件 | 用户已登录系统，且查询到不可借出的图书 |
| 基本事件流 | 用户在书籍列表中找到想借阅但当前不可借的书籍，查看到该书籍支持预约功能后，点击预约按钮  系统弹出预约确认对话框  用户确认预约后，将用户信息加入到该书籍的预约队列中 |
| 其他事件 | 若预约过程中，用户取消预约，系统从预约队列中移除该用户信息 |
| 后置条件 | 书籍的预约队列中添加该用户信息 |

1. 读者个人信息查看与修改用例规约如表1.7所示

表1.7读者个人信息查看与修改用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 个人信息查看与修改 |
| 用例说明 | 已登录的读者查看与修改个人信息 |
| 参与用户 | 读者 |
| 前置条件 | 用户已登录系统 |
| 基本事件流 | 用户在系统主界面点击个人信息选项，进入个人信息页面  用户如需修改信息，点击修改按钮，在相应字段中输入新的信息  系统弹出对话框提示修改成功 |
| 其他事件 | 系统对修改后的信息进行格式校验（如手机号格式、邮箱格式等），若不符合要求，提示用户修改 |
| 后置条件 | 用户可查看并更新自己的个人信息，更新后的信息保存到数据库中 |

1. 管理员注册用例规约如表1.8所示

表1.8管理员注册用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 注册 |
| 用例说明 | 管理员用户进行注册 |
| 参与用户 | 管理员 |
| 前置条件 | 无 |
| 基本事件流 | 1.新用户申请注册  2.管理员审核  3.审核通过与反馈  4.用户登录 |
| 其他事件流 | 若管理员审核发现信息缺失（如未填写姓名）或格式有误（如密码过短），则提示“信息不完整或格式有误”  若审核时发现用户填写的账号已被注册，则提示“账号已存在” |
| 后置条件 | 若注册成功，新用户信息成功录入系统数据库，用户拥有系统登录权限，可进行相应操作  若注册失败，用户信息未录入数据库，需重新注册 |

1. 管理员登录用例规约如表1.9所示

表1.9 管理员登录用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 登录 |
| 用例说明 | 管理员用户进行登录 |
| 参与用户 | 管理员 |
| 前置条件 | 图书管理系统处于正常的状态，网络连接正常  管理员已获取系统分配的用户名和密码 |
| 基本事件流 | 1.进入登录界面  2.输入登录信息  3.提交登录请求  4.登录成功 |
| 其他事件流 | 若服务器验证时发现用户名或密码错误，系统弹出提示框显示“用户名或密码错误，请重新输入” |
| 后置条件 | 若登录成功，管理员获得系统操作权限，可进行后续的用户管理、图书管理等相关操作，同时系统更新管理员的登录状态为“已登录” |

1. 库存管理用例规约如表1.10所示

表1.10 库存管理用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 库存管理 |
| 用例说明 | 确保图书库存信息准确，满足读者借阅要求 |
| 参与用户 | 管理员 |
| 前置条件 | 图书管理系统就绪，数据标准统一 |
| 基本事件流 | 1. 采购申请 2. 订单处理 3. 到货验收 4. 入库登记 |
| 其他事件流 | 确认是系统数据错误，可直接修改库存数量并记录修改原因 |
| 后置条件 | 完成库存查询、入库、出库、盘点等操作后，系统保持在库存管理页面，方便管理员继续进行其他操作 |

1. 图书状态管理用例规约如表1.11所示

表1.11图书状态管理管理用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 图书状态管理 |
| 用例说明 | 确保读者能准确了解图书当前情况，同时方便图书的流通与维护管理 |
| 参与用户 | 管理员 |
| 前置条件 | 明确定义图书所有可能状态 |
| 基本事件流 | 1. 采购验收 2. 完成编目 3. 验收后标记 4. 评估 |
| 其他事件流 | 批量操作异常，无法完成状态变更，系统提示哪些图书操作失败及原因，管理员需对异常图书单独处理 |
| 后置条件 | 完成图书状态查询、更新等操作后，系统保持在图书状态管理页面，方便管理员继续其他操作 |

1. 借阅管理用例规约如表1.12所示

表1.12借阅管理管理用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 借阅管理 |
| 用例说明 | 保障图书有序流通，维护借阅秩序 |
| 参与用户 | 管理员 |
| 前置条件 | 读者已完成并拥有有效账户 |
| 基本事件流 | 1. 读者出示借书证 2. 系统验证读者账户有效性 3. 检查借阅权限 |
| 其他事件流 | 读者已达到最大借阅数量上限，系统提示“您已达到最大借阅数量，无法继续借阅” |
| 后置条件 | 完成借阅、归还、续借等操作后，系统保持在“借阅管理”页面，方便管理员继续处理其他业务 |

1. 系统管理用例规约如表1.13所示

表1.13系统管理用例规约表

|  |  |
| --- | --- |
| **用例描述** | |
| 用例名称 | 系统管理 |
| 用例说明 | 保障系统稳定运行、数据安全可靠，同时优化系统功能以满足实际业务要求 |
| 参与用户 | 管理员 |
| 前置条件 | 满足系统负载要求的服务器硬件 |
| 基本事件流 | 1. 环境部署 2. 基础数据导入 3. 系统联调测试 |
| 其他事件流 | 参数配置错误，系统弹出提示框，指出错误项并要求管理员重新输入正确参数 |
| 后置条件 | 完成系统管理相关操作后，系统保持在系统管理页面，方便管理员继续进行其他操作 |

# 2. 系统设计

## 2.1 总体设计

本系统主要包括管理员端和读者端。管理员端功能模块图如图2.1所示，读者端功能模块图如图2.2所示



图2.1管理员端功能模块图



图2.2读者端功能模块图

## 2.2数据库设计

2.2.1概念结构设计

任务是在需求分析阶段产生的需求规格说明书的基础上，将需求抽象为一个不依赖于任何DBMS的数据模型，即概念模型，表现形式为E-R模型。

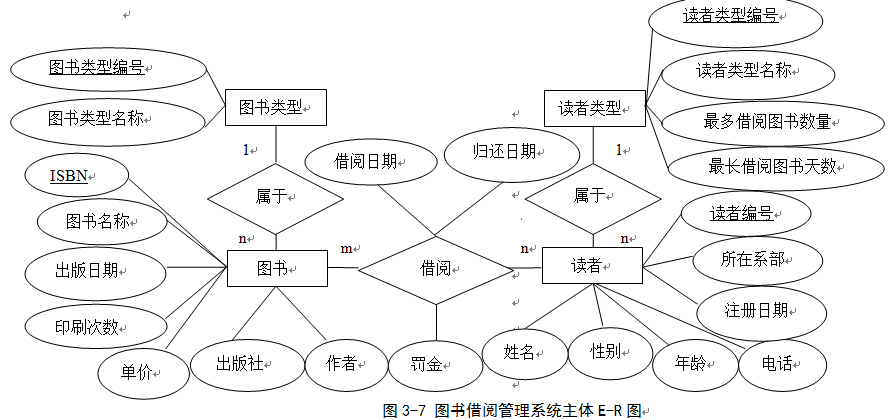


图2.3E-R图

2.2.2逻辑结构设计

任务是将概念模型转化为某个特定DBMS上的逻辑模型。具体来说，就是设计表。

图书（图书编号ISBN，图书名称，作者，出版社，出版日期，印刷次数，单价）

读者（读者编号，读者类型编号，姓名，年龄，性别，电话，所在系部，注册日期）

读者类型（读者类型编号，读者类型名称，最多可借图书数量，最长可借图书天数）

用户（用户编号，用户名，密码）

图书借阅（读者编号，图书编号，借阅日期，归还日期，罚金）

本系统所需的表如下：

1. 图书表（Book）该表结构如表2.1所示。

表2.1图书表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 主键 | 中文描述 |
| ISBN | varchar (20) | NOT NULL | PRIMARY KEY | 图书ISBN |
| bookname | varchar (50) | NOT NULL | NO | 图书名称 |
| password | varchar(50) | NOT NULL | NO | 作者 |
| email | varchar (20) | NULL | NO | 出版社 |
| avatar | timestamp | NULL | NO | 出版日期 |
| phone | int (11) | NOT NULL | NO | 印刷次数 |
| area\_id | double | NOT NULL | NO | 单价 |

参考此表，编写其他数据库表格.

## 2.3详细设计

在详细设计阶段对系统界面进行设计，包含每个功能的界面。

对每个模块进行详细设计，可以绘制流程图、时序图，或伪代码形式对程序进行详细设计。

（1）登录流程图

管理员输入用户名和密码，判断用户名密码是否为空，为空则提示“用户名密码不能为空”，判断用户名密码是否正确，正确则加载主页面，不正确则提示“用户名密码有误”。登录流程图如图2.4所示。

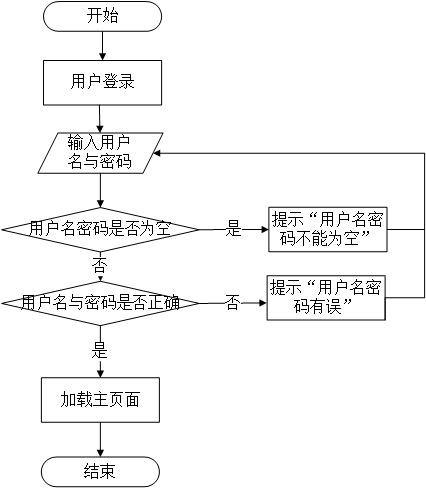


图2.4登录流程图

（2）XXX流程图

# 3. 系统实现

## 3.1 项目结构

描述项目结构，并进行截图

## 3.2功能实现

3.2.1管理员端功能实现

（1）管理员登录功能

管理员输入用户名和密码，点击登录按钮，进行登录。点击重置按钮将用户名和密码清空。登录界面如图3.1所示。

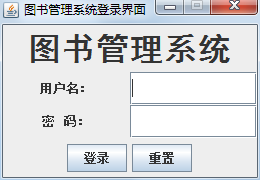


图3.1登录界面

**代码部分可以放置相关的Dao层代码或界面代码，需要放部分核心代码即可。需要包含所有界面截图。**

**public** **class** Login **extends** JFrame{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** JPanel textJP,loginJP,buttonJP;

**private** Font f1=**new** Font("黑体", Font.***BOLD***, 32);

**private** JLabel textJL,usernameJL,passwordJL;

**private** JTextField usernameJTF;

**private** JPasswordField pwdJPF;

**private** JButton confirmJB,resetJB;

**private** **static** Users *user*;

**public** Login(){

setSize(260,180);

//在整个界面的中间位置显示

**int** width=Toolkit.*getDefaultToolkit*().getScreenSize().width;

**int** height=Toolkit.*getDefaultToolkit*().getScreenSize().height;

**this**.setLocation(width/2-200, height/2-150);

setTitle("图书管理系统登录界面");

textJP=**new** JPanel();//提示信息面板

loginJP=**new** JPanel();//登录信息面板

buttonJP=**new** JPanel();//登录取消按钮面板

//提示信息面板

textJL=**new** JLabel();

textJL.setFont(f1);

textJL.setText("图书管理系统");

textJP.add(textJL);

**this**.add(textJP,BorderLayout.***NORTH***);

//登录信息面板设计

loginJP.setLayout(**new** GridLayout(2,2));

usernameJL=**new** JLabel("用户名：");

usernameJL.setHorizontalAlignment(SwingConstants.***CENTER***);

usernameJTF=**new** JTextField();

passwordJL=**new** JLabel("密 码：");

passwordJL.setHorizontalAlignment(SwingConstants.***CENTER***);

pwdJPF=**new** JPasswordField();

loginJP.add(usernameJL);

loginJP.add(usernameJTF);

loginJP.add(passwordJL);

loginJP.add(pwdJPF);

//登录取消按钮面板设计

confirmJB=**new** JButton("登录");

resetJB=**new** JButton("重置");

buttonJP.add(confirmJB);

buttonJP.add(resetJB);

**this**.add(loginJP,BorderLayout.***CENTER***);

**this**.add(buttonJP,BorderLayout.***SOUTH***);

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);//关闭窗口时，退出程序

**this**.setVisible(**true**);//设置窗体显示，否则不显示。

setResizable(**false**);//取消最大化

}

**public** **static** **void** setUser(Users user) {

Login.*user* = user;

}

**public** **static** Users getUser() {

**return** *user*;

}

**public** **static** **void** main(String[] args) {

**new** Login();

}

}

（2）图书信息添加功能

管理员输入图书ISBN，选择图书类别，填写书名作者、出版社、出版日期、印刷次数和单价等信息，点击添加按钮，实现图书添加。点击重置按钮将图书信息清空。点击关闭按钮，关闭图书信息添加界面。图书信息添加界面如图3.2所示。

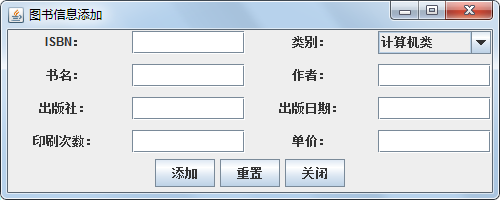


图3.2图书入库界面

添加图书界面代码如下：

**public** **class** BookAdd **extends** JFrame {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** JPanel bookAddJP,buttonJP;

**private** JLabel ISBNJL,booktypeJL,bookNameJL,authorJL,

publishJL,publishDateJL,printJL,priceJL;

**private** JTextField ISBNJTF,bookNameJTF,authorJTF,publishJTF,publishDateJTF,printJTF,

priceJTF;

**private** JComboBox booktypeJCB;

**private** JButton addJB,resetJB,closeJB;

**public** BookAdd(){

setBounds(200, 200, 500, 200);

setTitle("图书信息添加");

//图书信息添加面板设计

bookAddJP = **new** JPanel();

**final** GridLayout gridLayout = **new** GridLayout(4, 4);

gridLayout.setVgap(10);

gridLayout.setHgap(10);

bookAddJP.setLayout(gridLayout);

**this**.add(bookAddJP);

ISBNJL=**new** JLabel("ISBN：");

ISBNJL.setHorizontalAlignment(SwingConstants.***CENTER***);

bookAddJP.add(ISBNJL);

ISBNJTF=**new** JTextField();

bookAddJP.add(ISBNJTF);

booktypeJL=**new** JLabel("类别：");

booktypeJL.setHorizontalAlignment(SwingConstants.***CENTER***);

bookAddJP.add(booktypeJL);

//下拉列表

booktypeJCB = **new** JComboBox();

bookAddJP.add(booktypeJCB);

booktypeJCB.addItem("计算机类");

booktypeJCB.addItem("外语类");

bookNameJL=**new** JLabel("书名：");

bookNameJL.setHorizontalAlignment(SwingConstants.***CENTER***);

bookAddJP.add(bookNameJL);

bookNameJTF=**new** JTextField();

bookAddJP.add(bookNameJTF);

authorJL=**new** JLabel("作者：");

authorJL.setHorizontalAlignment(SwingConstants.***CENTER***);

bookAddJP.add(authorJL);

authorJTF=**new** JTextField();

bookAddJP.add(authorJTF);

publishJL=**new** JLabel("出版社：");

publishJL.setHorizontalAlignment(SwingConstants.***CENTER***);

bookAddJP.add(publishJL);

publishJTF=**new** JTextField();

bookAddJP.add(publishJTF);

publishDateJL=**new** JLabel("出版日期：");

publishDateJL.setHorizontalAlignment(SwingConstants.***CENTER***);

bookAddJP.add(publishDateJL);

publishDateJTF=**new** JTextField();

bookAddJP.add(publishDateJTF);

printJL=**new** JLabel("印刷次数：");

printJL.setHorizontalAlignment(SwingConstants.***CENTER***);

bookAddJP.add(printJL);

printJTF=**new** JTextField();

bookAddJP.add(printJTF);

priceJL=**new** JLabel("单价：");

priceJL.setHorizontalAlignment(SwingConstants.***CENTER***);

bookAddJP.add(priceJL);

priceJTF=**new** JTextField();

bookAddJP.add(priceJTF);

//按钮面板设计

buttonJP=**new** JPanel();

addJB=**new** JButton("添加");

resetJB=**new** JButton("重置");

closeJB=**new** JButton("关闭");

buttonJP.add(addJB);

buttonJP.add(resetJB);

buttonJP.add(closeJB);

**this**.add(bookAddJP,BorderLayout.***CENTER***);

**this**.add(buttonJP,BorderLayout.***SOUTH***);

**this**.setVisible(**true**);//设置窗体显示，否则不显示。

//setResizable(false);//取消最大化

}

**public** **static** **void** main(String[] args) {

**new** BookAdd();

}

}

添加图书Dao层代码如下：

**public static int** insertBook(String ISBN, String typename, String Bookname, String author, String publish, String publishdate,Integer printtime,Double price) {  
 **int** id = 0,i=0;  
 **try** {  
 String sql1=**"select \* from booktype where typename='"**+typename+**"'"**;   
 ResultSet rs = Dao.executeQuery(sql1);  
 *//BookType booktype=new BookType();* **try** {  
 **while** (rs.next()) {  
 id=rs.getInt(**"id"**);   
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 String sql = **"insert into Book(ISBN,typeId,Bookname,author,publish,publishdate,printtime,unitprice) values('"** + ISBN + **"','"** + id + **"','"** + Bookname + **"','"** + author + **"','"** + publish + **"','"** + publishdate + **"','"** + printtime + **"',"** + price + **")"**;  
 i = Dao.executeUpdate(sql);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 Dao.close();  
 **return** i;  
}

（3）图书借阅界面实现

在读者借阅信息区域中，管理员输入读者编号、读者姓名或读者类别，展示读者借阅信息。在图书借阅区域中，管理员输入图书信息，点击借阅，实现读者借阅图书功能。图书借阅借阅界面如图3.3所示。

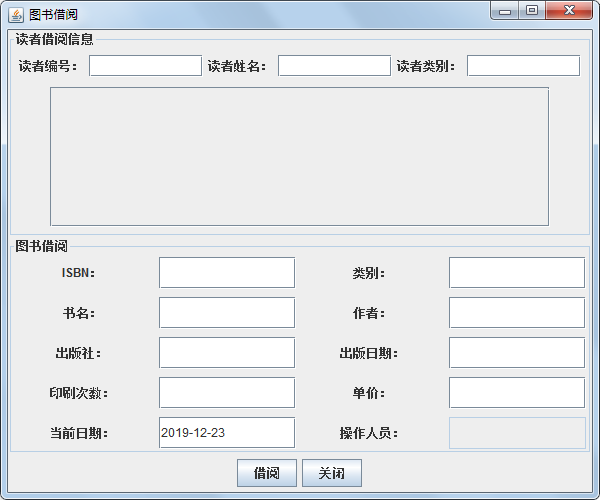


图3.3借书查阅界面

借阅功能Dao层代码如下：

**public static int** borrowBook(String readerid,String ISBN,Date borrowdate){  
 **int** i = 0;  
 **try** {  
 String sql = **"insert into borrowbook(readerid,ISBN,borrowdate) values('"**+readerid+**"','"**+ISBN+**"','"**+borrowdate+**"')"**;   
 i = Dao.executeUpdate(sql);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 Dao.close();  
 **return** i;  
}

3.2.2读者端功能实现