## 1.概述

## 1.1需求分析

图书管理信息系统需实现的功能如下：

（1）用户管理：包括用户的注册于登录。

（2）图书管理：包括录入、查询、修改和删除图书信息。

（3）借书：包括借阅图书和查看借书记录。

（4）还书：包括还书和查看还书记录。

## 1.2 设计思想

（1）该系统的设计分成一些相对独立的模块，这些模块都进行集中式管理。

（2）分层的模块化程序设计思想。

（3）合理的设计了数据库，使应用程序的数据显示为有效的信息。

# **2 总体设计**

## 2.1 数据库设计

图书信息表（books）结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| id | int | 图书顺序号，主键，自增 |
| isbn | varchar | ISBN号 |
| name | varchar | 书名 |
| author | varchar | 作者 |
| publisher | varchar | 出版社 |
| price | varchar | 价格 |
| pubDate | varchar | 出版日期 |
| deposit | varchar | 存放位置 |
| quantify | int | 数量 |
| lend | int | 借出数量 |

借书记录表（lendrecord）结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| id | int | 借书记录顺序号，主键，自增 |
| bookId | int | 图书顺序号 |
| userId | bigint | 借书人学号 |
| borrowName | varchar | 借书人姓名 |
| borrowDate | varchar | 借书日期 |
| state | varchar | 还书状态，已还：true |

还书记录表（returnrecord）结构

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| id | int | 还书记录顺序号，主键，自增 |
| bookId | int | 图书顺序号 |
| userId | bigint | 还书人学号 |
| returnName | varchar | 还书人姓名 |
| returnDate | varchar | 借书日期 |

用户信息表（user）结构

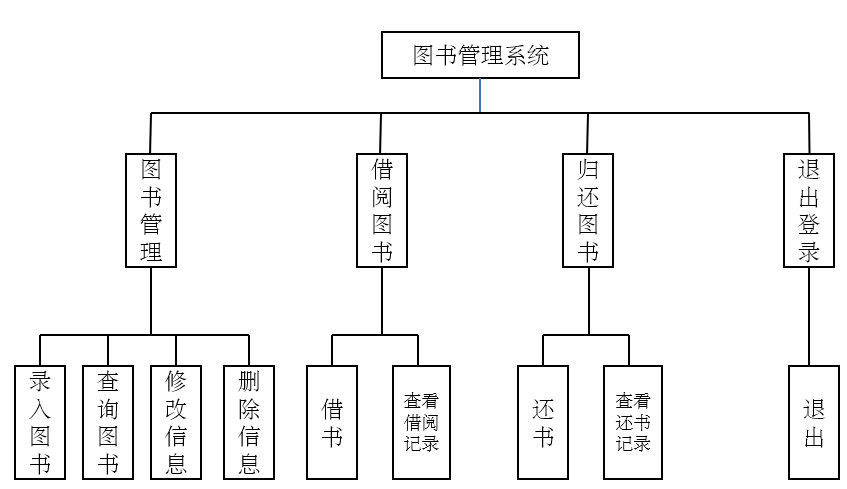
|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| id | int | 系统操作用户编号，主键自增 |
| userName | bigint | 用户学号，主键 |
| setPassword | varchar | 设置密码 |
| surePassword | varchar | 确认密码 |
| name | varchar | 用户姓名 |
| sex | varchar | 用户性别 |
| brithday | varchar | 出生日期 |

## 2.2 系统功能设计

根据图书管理系统需求分析，设计系统功能如下：

（1）注册：显示注册界面，用户输入自己的学号、密码、姓名、性别、出生日期，单击“注册”按钮后，系统为该用户进行注册，存入数据库中。

（2）登录：显示登录界面，用户输入用户名和密码，单击“登录”按钮后，系统验证用户名和密码是否正确，如正确则进入主界面。

（3）主界面：显示菜单，菜单结构如图所示

图书管理系统主菜单结构

1. 图书管理：共4项子功能，分别说明如下。

录入图书：用户在主界面菜单中选择此功能时，进入录入界面。用户在录入界面中输入各项图书信息后，单击“提交”按钮，系统将采集界面中的图书信息，将其存入数据库的图书信息表中。

查询图书：用户在主界面菜单中选择此功能时，进入查询界面。系统为用户提供完全信息查询和模糊查询功能。用户在查询界面中输入书名、作者和出版社信息后，选择查询条件为“完全一致”或“模糊查询”，最后单击“查询”按钮，系统按要求查询图书信息表，查询出图书信息显示于列表中。

修改图书：用户在主界面菜单中选择此功能时，进入修改界面1，首先根据用户输入信息完成图书信息的查询，用户在列表中选择需修改的图书后，进入修改界面2。用户在修改界面2中完成图书信息的修改后，单击“修改”按钮，系统采集界面2中的信息后，修改数据库中图书信息表中的相应记录。

删除图书：用户在主界面菜单中选择此功能时，进入删除界面。首先根据用户输入信息完成图书信息的查询，用户在列表中选择需删除的图书后，单击“删除”按钮，系统删除数据库中图书信息表中的相应记录。

1. 借书：共两项子功能,分别说明如下。

借书：用户在主界面菜单中选择此功能时，进入借阅图书界面。首先根据用户输入信息完成图书信息的查询，用户在列表中选择需借阅的图书后，单击“借书”按钮，系统修改图书信息表中的相应记录，并在借书记录表中登记一条记录。

查询借书记录：用户在主界面菜单中选择此功能时，进入借书记录界面。系统根据用户输人的信息完成借书记录的查询，如用户在查出的借书记录列表中选择一条借书记录，单击“查看详细信息”按钮，系统为用户查出所借图书的详细信息。

1. 还书：共两项子功能，分别说明如下。

还书：用户在主界面菜单中选择此功能时，进入还书界面。首先根据用户输入信息完成借书记录的查询，用户在列表中选择需还的图书后，单击“还书”按钮，系统修改图书信息表中的相应记录

查询还书记录：用户在主界面菜单中选择此功能时，进入还书记录界面。系统根据用户输入信息完成还书记录的查询，如用户在查出的还书记录列表中选择一条还书记录，单击“查看详细信息”按钮，系统为用户查出所还图书的详细信息。

（7）退出登录：用户在主界面菜单中选择此功能时，单击“退出”按钮后，系统会结束主菜单界面，退回到初始时的登录界面。

## 2.3 系统实现思路

数据操作层

数据库

用户

界面层

业务层

本系统采用的三层结构模型

用户通过界面输人信息或单击按钮，界面层通过事件处理控制转向业务层的不同处理方法进行处理，当需要对数据库进行操作时，业务层调用数据操作层的相应数据操作方法完成；之后数据操作层将结果返回业务层继续处理，业务层处理完毕，则再将结果返回到界面层，界面层显示结果，用户由此可进一步在界面上进行操作。

# **3 详细设计**

## 3.1 登录、注册功能的实现

“登录”功能运行效果 “注册”功能运行效果

3. 核心代码如下：

**public** **class** loginListener **implements** ActionListener {

@Override

**public** **void** actionPerformed(ActionEvent e) {

String userName = userField.getText().trim();

String password = MD5.*GetMD5Code*(passwordField.~~getText~~().trim());

**long** id = Service.*login*(userName, password); // 核对登录信息

**if** (id > 0) {

BooksManager app = **new** BooksManager(id); // 创建主界面

jframe1.dispose();

} **else** **if** (id == 0) { // 与数据库没有完全匹配

JOptionPane.*showMessageDialog*(jframe1, "您输入的用户名或密码不正确");

} **else** { // 与数据库完全不匹配

JOptionPane.*showMessageDialog*(jframe1, "查询用户表出错！");

}

}

}

@Override

**public** **void** actionPerformed(ActionEvent e) {

jframe2.setResizable(**false**);

jframe2.getContentPane().setBackground(Color.***PINK***);

jframe2.setVisible(**true**);

jframe1.dispose();

userField.setText("");

passwordField.setText("");

Service.*windowClose*(jframe2);

jframe2.setDefaultCloseOperation(JFrame.***DISPOSE\_ON\_CLOSE***);

regisiteButton.addActionListener(**new** regisiteListener2());

}

**public** **class** regisiteListener2 **implements** ActionListener {

@Override

**public** **void** actionPerformed(ActionEvent e) {

**if** (text1.getText().trim().equals("")) {

JOptionPane.*showMessageDialog*(jframe2, "学号项为空，请您重新填写！");

} **else** **if** (text2.~~getText~~().trim().equals("")) {

JOptionPane.*showMessageDialog*(jframe2, "设置密码项为空，请您重新填写！");

} **else** **if** (text3.~~getText~~().trim().equals("")) {

JOptionPane.*showMessageDialog*(jframe2, "确认密码项为空，请您重写填写！");

} **else** **if** (text4.getText().trim().equals("")) {

JOptionPane.*showMessageDialog*(jframe2, "姓名项为空，请您重写填写！");

} // else if (text5.getText().trim().equals("")) {

**else** **if** (str.equals("")) {

JOptionPane.*showMessageDialog*(jframe2, "性别项为空，请您重写填写！");

} **else** **if** (text6.getText().trim().equals("")) {

JOptionPane.*showMessageDialog*(jframe2, "出生日期项为空，请您重写填写！");

} **else** {

// 注册用户信息录入数据库

Service.*regisiteUser*(text1.getText().trim(), text2.~~getText~~().trim(), text3.~~getText~~().trim(),

text4.getText().trim(),str, text6.getText().trim());

}

}

}

## 3.2 主界面功能的实现

3. 核心代码如下：

@Override

**public** **void** actionPerformed(ActionEvent e) {

Object source = e.getSource();

**if** (source == insertMenuItem) { // 如果点击的是录入图书菜单项

InsertFrame insertFrame = **new** InsertFrame(userId); // 创建录入图书界面

**this**.dispose();

}

**if** (source == deleteMenuItem) { // 如果点击的是删除图书菜单项

DeleteFrame deleteFrame = **new** DeleteFrame(userId); // 删除图书

**this**.dispose();

}

**if** (source == updateMenuItem) { // 如果点击的是修改图书信息菜单项

UpdateFrame1 updateFrame = **new** UpdateFrame1(userId); // 修改图书信息

**this**.dispose();

}

**if** (source == selectMenuItem) { // 如果点击的是检索图书菜单项

SelectFrame selectFrame = **new** SelectFrame(userId);// 查询图书

**this**.dispose();

}

**if** (source == lendMenuItem) { // 如果点击的是开始借书菜单项

LendFrame lendFrame = **new** LendFrame(userId);// 开始借书

**this**.dispose();

}

**if** (source == lendRecordMenuItem) {// 如果点击的是借书记录菜单项

LendRecordFrame lendRecordFrame = **new** LendRecordFrame(userId); // 借书记录

**this**.dispose();

}

**if** (source == returnMenuItem) { // 如果点击的开始还书菜单项

ReturnFrame returnFrame = **new** ReturnFrame(userId); // 开始还书

**this**.dispose();

}

**if** (source == returnRecordMenuItem) { // 如果点击的是还书记录菜单项

ReturnRecordFrame returnRecordFrame = **new** ReturnRecordFrame(userId);

// 还书记录

**this**.dispose();

}

**if**(source == exitloadMenuItem) {

LoginManager lm = **new** LoginManager();

**this**.dispose();

}

}

## 3.3 录入功能的实现

“录入图书”功能运行效果

3. 核心代码如下：

**public** **class** SubmitListener **implements** ActionListener {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// 获取界面中的录入文本内容

String isbn = isbnField.getText().trim();

String name = nameField.getText().trim();

String author = authorField.getText().trim();

String publisher = publisherField.getText().trim();

String price = priceField.getText().trim();

String pubDate = pubDateField.getText().trim();

String deposit = depositField.getText().trim();

String quantify = quantifyField.getText().trim();

**if** (isbn.equals("")) {

JOptionPane.*showMessageDialog*(jframe, "'书号'项为空，请您重新填写！");

} **else** **if**(name.equals("")) {

JOptionPane.*showMessageDialog*(jframe, "'书名'项为空，请您重新填写！");

} **else** **if**(author.equals("")) {

JOptionPane.*showMessageDialog*(jframe, "'作者;项为空，请您重新填写！");

} **else** **if**(publisher.equals("")) {

JOptionPane.*showMessageDialog*(jframe, "'出版社'项为空，请您重新填写！");

} **else** **if**(price.equals("")) {

JOptionPane.*showMessageDialog*(jframe, "'价格'项为空，请您重新填写！");

} **else** **if**(pubDate.equals("")) {

JOptionPane.*showMessageDialog*(jframe, "'出版时间'项为空，请您重新填写！");

} **else** **if**(deposit.equals("")) {

JOptionPane.*showMessageDialog*(jframe, "'存放位置'为空，请您重新填写！");

} **else** **if**(quantify.equals("")||quantify.trim().equals("0")) {

JOptionPane.*showMessageDialog*(jframe, "'数量'项为空(0)，请您重新填写！");

} **else** { //文本框内容都不为空时

Vector<String> bookInfo = **new** Vector<String>(); // Vector类可以实现可增长的对象数组。

// 将获得得各文本框的内容依次存放到bookInfo字符串数组中

bookInfo.add(isbn);

bookInfo.add(name);

bookInfo.add(author);

bookInfo.add(publisher);

bookInfo.add(price);

bookInfo.add(pubDate);

bookInfo.add(deposit);

bookInfo.add(quantify);

**if** (Service.*addBook*(bookInfo) == 0) { // 当录入信息成功存储到数据库中

clearField(); // 清空界面文本框内容

JOptionPane.*showMessageDialog*(jframe, "添加图书成功！");

} **else** { // 录入信息添加到数据库中失败

JOptionPane.*showMessageDialog*(jframe, "添加图书失败，请重新确认信息是否填写正确！");

}

}

}

}

**public** **class** CancelListener **implements** ActionListener {

@Override

**public** **void** actionPerformed(ActionEvent e) {

clearField(); // 当点击取消时，清空界面文本框内容

}

}

**public** **class** GetbackListener **implements** ActionListener {

@Override

**public** **void** actionPerformed(ActionEvent e) {

JFrame app = **new** BooksManager(userId); // 创建主界面

jframe.dispose(); // 关闭录入图书界面

}

}

## 3.4 借书功能的实现

“借书”功能运行效果

3. 核心代码如下：

**public** **class** LendListener **implements** ActionListener {

@Override

**public** **void** actionPerformed(ActionEvent e) {

String bookValue = (String) list.getSelectedValue();

**if** (bookValue == **null**) {

JOptionPane.*showMessageDialog*(**null**, "请先进行检索，然后再选一本图书！");

**return**;

}

Vector<String> result = **new** Vector<String>();

String str = bookValue;

**int** index = str.indexOf(':');

result.add(str.substring(2, index));

**int** index1 = 0;

str = str.substring(index + 1);

**for** (**int** i = 1; i <= 9; i++) {

index = str.indexOf(':');

index1 = str.indexOf(';');

result.add(str.substring(index + 1, index1).trim());

str = str.substring(index1 + 1);

}

String book[] = **new** String[10];

**for** (**int** i = 0; i < 10; i++) {

book[i] = result.elementAt(i).trim();

}

// 在馆数目

**int** remainder = Integer.*parseInt*(book[8]) - Integer.*parseInt*(book[9]);

//System.out.println(remainder);

**if** (remainder > 0) { // 还可以进行借此书

**int** confirm = JOptionPane.*showConfirmDialog*(**null**,

"您决定要借阅的一本书的信息如下: \n书号: " + book[1] + "\n书名: " + book[2] + "\n作者:" + book[3] + "\n出版社: "

+ book[4] + "\n价格: " + book[5] + "\n出版时间: " + book[6] + "\n存放位置: " + book[7]

+ "\n在馆数量: " + remainder + "\n" + "确定借阅吗?");

**if** (confirm > 0) {

JOptionPane.*showMessageDialog*(**null**, "您已取消了这次借书！");

**return**;

}

**int** id = Integer.*parseInt*(book[0]);

**int** lendQuantify = Integer.*parseInt*(book[9]) + 1;

// id:数的序号，userId:用户名（学号）lendQuantify：已借走此书的数量

**if** (Service.*lendBook*(id, userId, lendQuantify) == -1) {

JOptionPane.*showMessageDialog*(**null**, "此次操作失败，请重新操作！");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "已成功借走此书！");

}

} **else** {

JOptionPane.*showMessageDialog*(**null**, "此书目前在馆数量为0，请重新选择！");

}

}

## 4.2软件调试

在Intellij IDEA 2023.2环境下(数据库：MySQL 8.0)，创建一个新的工程，在项目结构中加载外部包mysql-connector-java-8.0.33-bin.jar。然后运行主类LoginManager即可。