**福建中医药大学**

人文与管理学院

2017年秋季学期

**《Java程序设计》课程设计**

题 目： 图书信息管理系统\_\_

专业班级： 15信管­\_\_\_\_\_\_\_\_

组 别： 第 3 组\_\_\_\_\_\_\_\_

指导教师： 黄 敏\_\_\_\_\_\_\_\_\_

成 绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2018年1月

目录

[第一章 图书馆管理系统分析 1](#_Toc3846)

[1.1 需求分析  1](#_Toc13014)

[1.2 功能分析  2](#_Toc6319)

[1.3 系统用例图设计  2](#_Toc5625)

[1.4 绘制系统流程图  3](#_Toc27018)

[1.5 系统的开发环境  4](#_Toc17592)

[第二章 数据库分析与设计 4](#_Toc18688)

[2.1 数据库分析  4](#_Toc29791)

[2.2 数据库概念设计  4](#_Toc7306)

[2.3 数据逻辑结构设计  6](#_Toc27745)

[第三章 系统设计与功能实现 8](#_Toc3654)

[3.1 数据库Dao类的构建 8](#_Toc29966)

[3.2 系统登录模块  26](#_Toc6465)

[3.2.1 登陆窗体界面设计与实现  27](#_Toc10548)

[3.3 主窗体模块  31](#_Toc12581)

[3.3.1 主窗体界面设计  31](#_Toc10661)

[3.4 图书类别管理模块  36](#_Toc10273)

[3.4.1 图书类别添加模块   36](#_Toc27114)

[3.4.2 图书类别修改模块  39](#_Toc19782)

[3.5 图书信息管理模块  45](#_Toc18361)

[3.5.1 图书信息添加界面设计  45](#_Toc20153)

[3.5.3 图书信息修改  54](#_Toc27361)

[3.6 读者信息管理模块  65](#_Toc1627)

[3.6.1 读者信息添加  65](#_Toc6602)

[3.6.2 读者信息修改与删除  77](#_Toc19676)

[3.9 系统维护模块  91](#_Toc20151)

[3.9.1 修改密码  91](#_Toc3677)

[3.9.2 管理员信息添加  92](#_Toc31861)

[3.9.3 用户信息修改与删除  100](#_Toc29952)

[第四章 系统测试 114](#_Toc9496)

[4.1 测试项目  114](#_Toc24856)

[4.2 测试用例  114](#_Toc15340)

第一章 图书馆管理系统分析

1.1 需求分析

图书馆管理系统是一个高度集成的图书信息处理系统，通过将图书馆的各种功能进行整合，从而达到显示检索信息，提高工作效率，降低管理成本等目的。一个典型的图书馆管理系统应该能够管理所有的图书种类，图书信息以及读者信息，还需要提供各种图书信息的检索查询功能。该系统还需要能够对图书的借阅，归还进行管理，并对读者的罚款进行自动计算。通过该系统的自动化管理，能够大大减少图书馆管理人员，还能减少管理人员的工作任务，从而降低管理开销和成本。

一个完整的图书馆管理系统包括前台和后台，前台主要是显示在计算机屏幕上的显示界面，有各种指令按钮，操作框以及文本框，后台主要是为前台的一些操作提供一些必要的数据，也就是一个相对于系统比较完整的数据库，操作员以及各类图书的信息等等。

一个最基本的图书馆管理系统要有如下几个重要功能：

（1）用户在借书超期的情况下得到来自管理员的提醒。

（2）管理员可以方便进行图书管理，用户管理，管理员管理。图书管理包括图书信息以及图书分类的添加，修改，删除。用户管理包括用户信息的添加，删除，修改。管理员管理包括管理员信息的添加，删除，修改等。

（3）用户和管理员可以修改自己的密码，修改前需先核实自己的原始密码。

（4）未注册用户（游客）也可以浏览所有的图书信息和分类信息。

（5）实现模糊查询，使用户得到更多的相关记录。并且考虑使用的方便性，一些经常使用的输入无须用户输入，比如进行图书查询时图书分类只须用户做选择就可以。

（6）考虑程序执行操作时可能出现的情况，比如删除图书分类时该分类下存在图书，程序自动跳转该分类图书查看。

一个图书馆管理系统要是能实现以上的各种功能，那么这个图书馆管理系统也就算是比较成功的一个系统了。

1.2 功能分析

根据以上需求分析，我所设计的图书馆管理系统有5个功能模块，分别是图书类别管理模块，图书信息管理模块，读者信息管理模块，图书借阅模块以及系统维护模块。其中各功能模块的具体说明如下：

图书类别管理模块：该模块主要负责管理图书馆的图书种类信息，如图书类型、编号等信息。

图书信息管理模块：该模块主要负责管理图书馆的图书信息，如图书的国际标准书号、书名、作者、出版社、图书类型等信息。

读者信息管理模块：该模块主要负责管理图书馆的用户信息，如用户姓名、性别、年龄、办证日期、证件号码和押金等信息。

系统维护模块：该模块主要负责图书馆的工作人员信息，包括用户管理和更改系统口令两个子模块。

1.3 系统用例图设计

图书馆管理系统是一个内部人员使用的系统，也就是说不是所有的人都能够使用它，只有图书馆的工作人员才能使用。而图书馆的工作人员也分为两类，一类是操作人员，主要负责图书的借阅和归还的工作；一类是管理员，除了操作人员的所有功能外，还能够对书籍列表，书籍信息，读者信息等进行管理。下面以管理员为例绘制其所对应的用例图如图1.1所示。

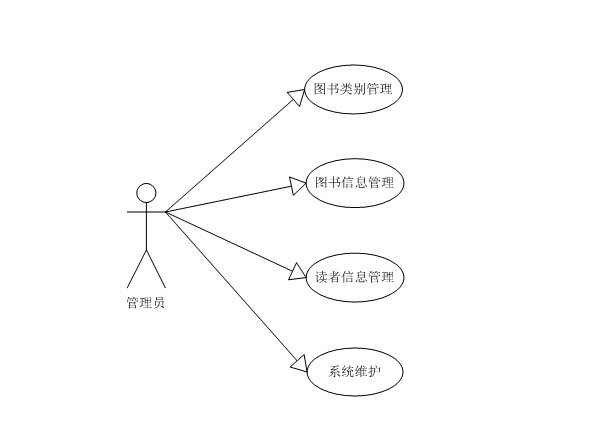
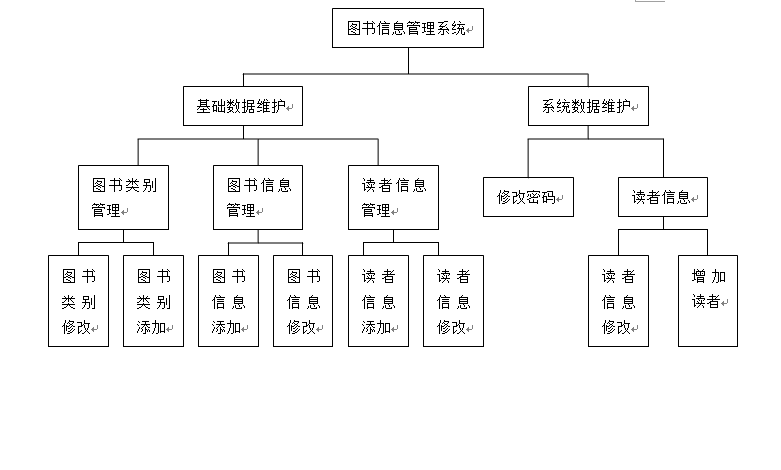


              图1.1 管理员用例图

1.4 绘制系统流程图

本系统首先需要对用户的身份进行识别，只有合法的用户才能进入系统，否则将无法进入系统。进入系统后，首先打开系统主窗体，在系统首页的菜单栏或者功能区可以选择各种导航链接来进行各种操作。由于不同权限的用户对于系统有不同的功能，下面以功能最多的管理员为例，其系统流程图如图1.2所示。

图1.2 系统流程图

1.5 系统的开发环境

图书馆管理系统的具体开发环境要求如下：

（1）系统开发平台：Eclipse 3.4。

（2）数据库管理系统软件：SQL Server 2008。

（3）运行平台：Windows 7。

（4）Java开发包：JDK 5.0以上。

（5）分辨率：800×600以上。

（6）GUI开发包：Swing。

第二章 数据库分析与设计

2.1 数据库分析

在开发图书馆管理系统时，考虑到图书量大，数据库维护大的特点，选用SQL Server 2008作为数据库管理系统。在SQL Server 2008中新增一个数据库，其数据库名为BMSBase。数据库中可以包含图书信息、图书类别信息、管理员信息及读者信息等实体，用来存储不同的信息。

2.2 数据库概念设计

本系统一共设计规划出4个实体，分别是图书类别信息实体、图书信息实体、读者信息实体、以及管理员实体。

图书的类别有很多，因此可以建立一个图书类别信息表，专门用来保存图书的类别信息。图书类别信息实体E-R图如图2.1所示。

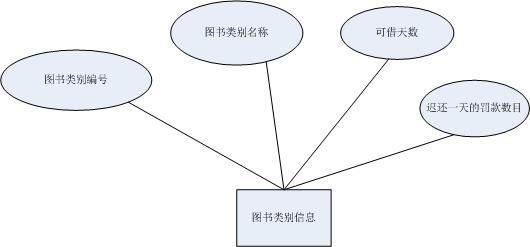


图2.1 图书类别信息实体E-R图

对于图书馆来说最重要的就是要管理其下的书籍，所以需要建立一个图书信息表，用来保存图书的所有信息。图书信息实体E-R图如图2.2所示。

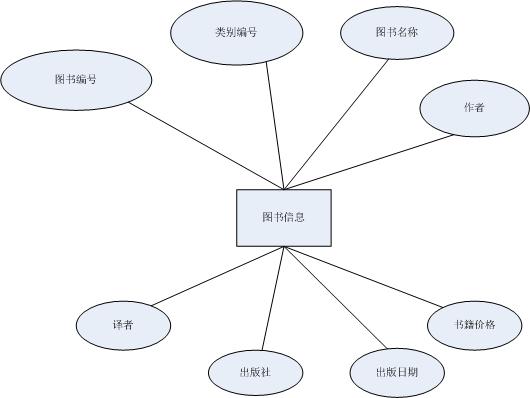


图2.2 图书信息实体E-R图

要想在图书馆借书首先需要进行登记并交付押金。领取读书卡才能借书。所以需要建立一个读者信息表来保存图书馆的所有读者的登记信息。读者信息实体E-R图如图2.3所示。

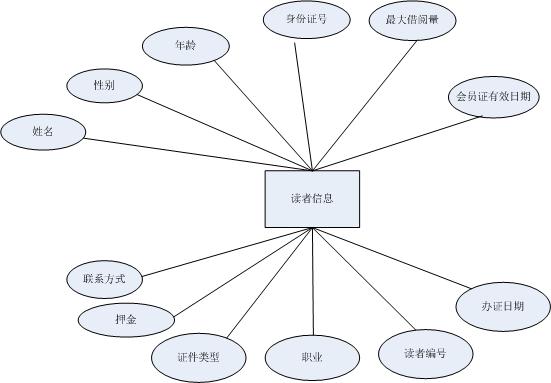


图2.3 读者信息实体E-R图

图书馆里一般有一个以上的系统管理员，需要建立一个管理员信息表，用来保存管理员的身份信息。管理员信息实体E-R图如图2.4所示。

2.3 数据逻辑结构设计

根据设计好的各实体E-R图创建数据库的逻辑结构，数据库各表的结构如下：

（1）图书类别信息采表用来储存所有的图书类别信息，包括图书类别编号、图书类别名称、可借天数以及迟还一天的罚款数目4个字段。该表的逻辑结构如表2.1所示。

表2.1 图书类别表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 说明 | 类型 | 长度及备注 |
| TypeName | 图书类型 | Varchar(20) |  |
| Id | 自动编号 | Int | 主键 |

（2）图书信息表用来储存所有的图书信息，包括图书国际标准书号、图书类型、书名、作者、译者、出版社、出版日期以及价格8个字段。该表的逻辑结构如表2.2所示。

表2.2 图书信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 说明 | 类型 | 长度及备注 |
| ISBN | 国际标准书号 | Varchar(13) | 主键 |
| typeId | 图书类型id | Int |  |
| Bookname | 书名 | Varchar(40) |  |
| Writer | 作者 | Varchar(21) |  |
| Translater | 译者 | Varchar(30) | 可以为空 |
| Publisher | 出版社 | Varchar(50) |  |
| Date | 出版日期 | Smalldatetime |  |
| Price | 价格 | money |  |

（3）读者信息表用来储存所有的读者信息，包括读者姓名、性别、年龄、证件号码、日期、最大书量、电话号码、押金、证件类型、职业、国际标准书号以及办证日期12个字段。该表的逻辑结构如表2.3所示。

表2.3 读者信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 说明 | 类型 | 长度及备注 |
| Name | 姓名 | Varchar(10) |  |
| Sex | 性别 | Varchar(2) |  |
| Age | 年龄 | int |  |
| identityCard | 证件号码 | Varchar(30) |  |
| Date | 日期 | Smalldatetime |  |
| manNum | 最大书量 | Int |  |
| Tel | 电话号码 | Varchar(15) |  |
| keepMoney | 押金 | Miney |  |
| Zj | 证件类型 | Int |  |
| Zy | 职业 | Varchar(50) |  |
| ISBN | 国际标准书号 | Varchar(13) |  |
| bztime | 办证时间 | datetime |  |

（4）管理员信息表用来保存管理员信息，包括管理员编号、用户名、性别、年龄、证件号码、工作时间、电话号码、是否为管理员、密码9个字段。该表的逻辑结构如表2.4所示。

表2.4 管理员信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 说明 | 类型 | 长度及备注 |
| Name | 姓名 | Varchar(10) |  |
| Sex | 性别 | Varchar(2) |  |
| Age | 年龄 | int |  |
| Identitycard | 证件号码 | Varchar(30) |  |
| Tel | 电话号码 | Varchar(15) |  |
| Password | 密码 | Varchar(30) |  |
| UserId | 用户编号 | Int |  |
| UserName | 用户名 | Varchar(10) |  |

第三章 系统设计与功能实现

3.1 数据库Dao类的构建

首先定义了一个类Dao，并为该类添加了4中方法，分别是构造方法Dao、查询方法executeQuery、更新方法executeUpdate以及关闭连接方法close。

public class Dao {

protected static String dbClassName = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

protected static String dbUrl = "jdbc:sqlserver://localhost:1433;" + "DatabaseName=BMSBase; SelectMethod=Cursor";

protected static String dbUser = "sa";

protected static String dbPwd = "sa";

protected static String second = null;

private static Connection conn = null;

// 创建数据库连接

private Dao() {

try {

if (conn == null) {

Class.forName(dbClassName).newInstance();

conn = DriverManager.getConnection(dbUrl, dbUser, dbPwd);

System.out.println("数据库连接成功！");

} else

return;

} catch (Exception ee) {

ee.printStackTrace();

}

}

// 执行查询

private static ResultSet executeQuery(String sql) {

try {

if (conn == null)

new Dao();

return conn.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_UPDATABLE).executeQuery(sql);

} catch (SQLException e) {

e.printStackTrace();

return null;

} finally {

}

}

// 执行更新

private static int executeUpdate(String sql) {

try {

if (conn == null)

new Dao();

return conn.createStatement().executeUpdate(sql);

} catch (SQLException e) {

System.out.println(e.getMessage());

// if(e.getMessage().equals("[Microsoft][SQLServer 2000 Driver for

// JDBC][SQLServer]DELETE 语句与 COLUMN REFERENCE 约束

// 'FK\_TB\_BORRO\_REFERENCE\_TB\_BOOKI' 冲突。该冲突发生于数据库 'db\_library'，表

// 'tb\_borrow', column 'bookISBN'。"))

return -1;

} finally {

}

}

// 关闭数据库

public static void close() {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

} finally {

conn = null;

}

}

/\*

\* 管理员登录方法

\*/

public static Operator check(String id, String password) {

int i = 0;

Operator operater = new Operator();

String sql = "select \* from tb\_operator where id='" + id + "' and password='" + password + "'and admin=1";

ResultSet rs = Dao.executeQuery(sql);

try {

while (rs.next()) {

String ids = rs.getString("id");

operater.setId(rs.getString("id"));

operater.setName(rs.getString("name"));

operater.setGrade(rs.getString("admin"));

operater.setPassword(rs.getString("password"));

if (ids != null) {

i = 1;

}

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return operater;

}

/\*

\* 查询类别方法

\*/

// 查询图书目录

public static List selectBookCategory() {

List list = new ArrayList();

String sql = "select \* from tb\_bookType";

ResultSet rs = Dao.executeQuery(sql);

try {

while (rs.next()) {

BookType bookType = new BookType();

bookType.setId(rs.getString("id"));

bookType.setTypeName(rs.getString("typeName"));

list.add(bookType);

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return list;

}

public static List selectBookCategory(String bookType) {

List list = new ArrayList();

String sql = "select days from tb\_bookType where typeName='" + bookType + "'";

ResultSet rs = Dao.executeQuery(sql);

try {

while (rs.next()) {

BookType type = new BookType();

type.setDays(rs.getString("days"));

list.add(type);

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return list;

}

/\*

\* 图书类别表相关操作

\*

\*/

// 插入图书类型

public static int InsertBookType(String bookTypeName) {

int i = 0;

try {

String sql = "insert into tb\_bookType(typeName) values('" + bookTypeName + "')";

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

return i;

}

// 更新图书类别

public static int UpdatebookType(String id, String typeName) {

int i = 0;

try {

String sql = "update tb\_bookType set typeName='" + typeName + "' where id='" + id + "'";

// System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

// 删除图书类别

public static int DelbookType(String id) {

int i = 0;

try {

String sql = "delete from tb\_bookType where id='" + id + "'";

// System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

/\*

\* 图书信息表相关操作

\*/

/\*

\* 插入图书信息方法

\*/

// 插入图书信息

public static int Insertbook(String ISBN, String typeId, String bookname, String writer, String translator,

String publisher, Date date, Double price) {

int i = 0;

try {

String sql = "insert into tb\_bookInfo(ISBN,typeId,bookname,writer,translator,publisher,date,price) values('"

+ ISBN + "','" + typeId + "','" + bookname + "','" + writer + "','" + translator + "','" + publisher

+ "','" + date + "'," + price + ")";

// System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

System.out.println(e.getMessage());

}

Dao.close();

return i;

}

/\*

\* 查询图书相关信息

\*

\*/

public static List selectBookInfo() {

List list = new ArrayList();

String sql = "select \* from tb\_bookInfo";

ResultSet rs = Dao.executeQuery(sql);

try {

while (rs.next()) {

BookInfo bookinfo = new BookInfo();

bookinfo.setISBN(rs.getString("ISBN"));

bookinfo.setTypeid(rs.getString("typeid"));

bookinfo.setBookname(rs.getString("bookname"));

bookinfo.setWriter(rs.getString("writer"));

bookinfo.setTranslator(rs.getString("translator"));

bookinfo.setPublisher(rs.getString("publisher"));

bookinfo.setDate(rs.getDate("date"));

bookinfo.setPrice(rs.getDouble("price"));

list.add(bookinfo);

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return list;

}

public static List selectBookInfo(String ISBN) {

List list = new ArrayList();

String sql = "select \* from tb\_bookInfo where ISBN='" + ISBN + "'";

ResultSet rs = Dao.executeQuery(sql);

try {

while (rs.next()) {

BookInfo bookinfo = new BookInfo();

bookinfo.setISBN(rs.getString("ISBN"));

bookinfo.setTypeid(rs.getString("typeid"));

bookinfo.setBookname(rs.getString("bookname"));

bookinfo.setWriter(rs.getString("writer"));

bookinfo.setTranslator(rs.getString("translator"));

bookinfo.setPublisher(rs.getString("publisher"));

bookinfo.setDate(rs.getDate("date"));

bookinfo.setPrice(rs.getDouble("price"));

list.add(bookinfo);

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return list;

}

/\*

\* 修改图书信息方法

\*/

// 更新图书信息

public static int Updatebook(String ISBN, String typeId, String bookname, String writer, String translator,

String publisher, Date date, Double price) {

int i = 0;

try {

String sql = "update tb\_bookInfo set ISBN='" + ISBN + "',typeId='" + typeId + "',bookname='" + bookname

+ "',writer='" + writer + "',translator='" + translator + "',publisher='" + publisher + "',date='"

+ date + "',price=" + price + " where ISBN='" + ISBN + "'";

// System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

/\*

\* 删除图书信息方法

\*/

public static int Delbook(String ISBN) {

int i = 0;

try {

String sql = "delete from tb\_bookInfo where ISBN='" + ISBN + "'";

// System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

/\*

\* 对读者信息表执行的相关操作

\*/

// 增加读者信息

public static int InsertReader(String name, String sex, String age, String identityCard, Date date, String maxNum,

String tel, Double keepMoney, String zj, String zy, Date bztime, String ISBN) {

int i = 0;

try {

String sql = "insert into tb\_reader(name,sex,age,identityCard,date,maxNum,tel,keepMoney,zj,zy,bztime,ISBN) values('"

+ name + "','" + sex + "','" + age + "','" + identityCard + "','" + date + "','" + maxNum + "','"

+ tel + "'," + keepMoney + ",'" + zj + "','" + zy + "','" + bztime + "','" + ISBN + "')";

System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

public static List selectReader() {

List list = new ArrayList();

String sql = "select \* from tb\_reader";

ResultSet rs = Dao.executeQuery(sql);

try {

while (rs.next()) {

Reader reader = new Reader();

// reader.setId(rs.getString("id"));

reader.setName(rs.getString("name"));

reader.setSex(rs.getString("sex"));

reader.setAge(rs.getString("age"));

reader.setIdentityCard(rs.getString("identityCard"));

reader.setDate(rs.getDate("date"));

reader.setMaxNum(rs.getString("maxNum"));

reader.setTel(rs.getString("tel"));

reader.setKeepMoney(rs.getDouble("keepMoney"));

reader.setZj(rs.getString("zj"));

reader.setZy(rs.getString("zy"));

reader.setISBN(rs.getString("ISBN"));

reader.setBztime(rs.getDate("bztime"));

list.add(reader);

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return list;

}

public static List selectReader(String readerISBN) {

List list = new ArrayList();

String sql = "select \* from tb\_reader where ISBN='" + readerISBN + "'";

ResultSet rs = Dao.executeQuery(sql);

try {

while (rs.next()) {

Reader reader = new Reader();

reader.setName(rs.getString("name"));

reader.setSex(rs.getString("sex"));

reader.setAge(rs.getString("age"));

reader.setIdentityCard(rs.getString("identityCard"));

reader.setDate(rs.getDate("date"));

reader.setMaxNum(rs.getString("maxNum"));

reader.setTel(rs.getString("tel"));

reader.setKeepMoney(rs.getDouble("keepMoney"));

reader.setZj(rs.getString("zj"));

reader.setZy(rs.getString("zy"));

reader.setISBN(rs.getString("ISBN"));

reader.setBztime(rs.getDate("bztime"));

list.add(reader);

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return list;

}

// 更新读者信息

public static int UpdateReader(String id, String name, String sex, String age, String identityCard, Date date,

String maxNum, String tel, Double keepMoney, String zj, String zy, Date bztime, String ISBN) {

int i = 0;

try {

String sql = "update tb\_reader set name='" + name + "',sex='" + sex + "',age='" + age + "',identityCard='"

+ identityCard + "',date='" + date + "',maxNum='" + maxNum + "',tel='" + tel + "',keepMoney="

+ keepMoney + ",zj='" + zj + "',zy='" + zy + "',bztime='" + bztime + "'where ISBN='" + ISBN + "'";

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

// 删除读者信息

public static int DelReader(String ISBN) {

int i = 0;

try {

String sql = "delete from tb\_reader where ISBN='" + ISBN + "'";

// System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

// new

public static List selectbookserch() {

List list = new ArrayList();

String sql = "select \* from tb\_bookInfo";

ResultSet s = Dao.executeQuery(sql);

try {

while (s.next()) {

BookInfo bookinfo = new BookInfo();

bookinfo.setISBN(s.getString(1));

bookinfo.setTypeid(s.getString(2));

bookinfo.setBookname(s.getString(3));

bookinfo.setWriter(s.getString(4));

bookinfo.setTranslator(s.getString(5));

bookinfo.setPublisher(s.getString(6));

bookinfo.setDate(s.getDate(7));

bookinfo.setPrice(s.getDouble(8));

list.add(bookinfo);

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return list;

}

// 添加管理员

public static int Insertoperator(String name, String sex, int age, String identityCard, Date workdate, String tel,

String password, int admin) {

int i = 0;

try {

String sql = "insert into tb\_operator(name,sex,age,identityCard,workdate,tel,password,admin) values('"+name + "','" + sex + "'," + age + ",'" + identityCard + "','" + workdate + "','" + tel + "','"+ password + "','" + admin + "')";

System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

public static List selectuser() {

List list = new ArrayList();

String sql = "select id,name,sex,age,identityCard,workdate,tel,password from tb\_operator where admin=1";

ResultSet rs = Dao.executeQuery(sql);

try {

while (rs.next()) {

User user = new User();

user.setId(rs.getInt(1));

user.setName(rs.getString(2));

user.setSex(rs.getString(3));

user.setAge(rs.getInt(4));

user.setIdentityCard(rs.getString(5));

// user.setYajin(rs.getInt(5));

user.setWorkdate(rs.getDate(6));

user.setTel(rs.getString(7));

user.setPassword(rs.getString(8));

list.add(user);

}

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return list;

}

// 删除管理员

public static int Deluser(int id) {

int i = 0;

try {

String sql = "delete from tb\_operator where id='" + id + "'";

// System.out.println(sql);

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

// 更新管理员信息

public static int Updateuser(int id, String name, String sex, int age, Date workdate, String tel, String password) {

int i = 0;

try {

String sql = "update tb\_operator set name='" + name + "',sex='" + sex + "',age=" + age + ",workdate='"+ workdate + "',tel='" + tel + "',password='" + password + "' where id='" + id + "'";

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

public static int Updatepass(String password, String name) {

int i = 0;

try {

String sql = "update tb\_operator set password='" + password + "' where name='" + name + "'";

i = Dao.executeUpdate(sql);

} catch (Exception e) {

e.printStackTrace();

}

Dao.close();

return i;

}

}

3.2 系统登录模块

对于图书馆管理系统而言，不是所有的用户都是可以使用的，所有需要添加一个登陆模块。通过该模块来对用户的合法性进行判断，只有合法的用户才能进入到系统。整个登陆模块的实现非常简单，相信读者能够快速掌握其开发过程。登录模块的运行结果如图3.1

所示。



图3.1 系统登录

3.2.1 登陆窗体界面设计与实现

登陆窗体的界面设计依然采用的是比较传统的方法，新建一个类BookLoginFrame类，用来显示登录窗体。在该窗体中需要使用到四类控件，分别是JLabel、JTextFieid、JPasswordField以及JButton。其中控件JLabel是用来显示顶部的图片以及标签文本；控件JTextFieid用来接收用户名输入信息；控件JPasswordField用来接收密码输入信息；控件JButton用来创建登陆和重置按钮。

public class LoginIFrame extends JFrame {

private class BookResetAction implements ActionListener {

public void actionPerformed(final ActionEvent e) {

username.setText("");

password.setText("");

}

}

class BookLoginAction implements ActionListener {

public void actionPerformed(final ActionEvent e) {

user = Dao.check(username.getText(), password.getText());

if (user.getName() != null) {

try {

JOptionPane.showMessageDialog(null, "登录成功！");

MainFrame frame = new MainFrame();

frame.setVisible(true);

LoginIFrame.this.setVisible(false);

} catch (Exception ex) {

ex.printStackTrace();

}

} else {

JOptionPane.showMessageDialog(null, "用户名或密码错误！");

}

}

}

private JPasswordField password;

private JTextField username;

private JButton login;

private JButton reset;

private static Operator user;

/\*\*

\* Create the frame

\*/

public LoginIFrame() {

super();

final BorderLayout borderLayout = new BorderLayout();

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

borderLayout.setVgap(10);

getContentPane().setLayout(borderLayout);

setTitle("图书信息管理系统");

setBounds(100, 100, 355, 250); // 设置窗体位置和大小

final JPanel panel = new JPanel();

panel.setLayout(new BorderLayout());

panel.setBorder(new EmptyBorder(0, 0, 10, 0));

getContentPane().add(panel);

final JPanel panel\_2 = new JPanel();

final GridLayout gridLayout = new GridLayout(0, 2);

gridLayout.setHgap(5);

gridLayout.setVgap(20);

panel\_2.setLayout(gridLayout);

panel.add(panel\_2);

final JLabel label = new JLabel();

label.setHorizontalAlignment(SwingConstants.CENTER);

label.setPreferredSize(new Dimension(0, 0));

label.setMinimumSize(new Dimension(0, 0));

panel\_2.add(label);

label.setText("用 户：");

username = new JTextField(20);

username.setPreferredSize(new Dimension(0, 0));

panel\_2.add(username);

final JLabel label\_1 = new JLabel();

label\_1.setHorizontalAlignment(SwingConstants.CENTER);

panel\_2.add(label\_1);

label\_1.setText("密 码：");

password = new JPasswordField(20);

password.setDocument(new MyDocument(6));

password.setEchoChar('\*');// 设置密码框的回显字符

password.addKeyListener(new KeyAdapter() {

public void keyPressed(final KeyEvent e) {

if (e.getKeyCode() == 10) // 回车

login.doClick();

}

});

panel\_2.add(password);

final JPanel panel\_1 = new JPanel();

panel.add(panel\_1, BorderLayout.SOUTH);

login = new JButton();

login.addActionListener(new BookLoginAction());

login.setText("登录");

panel\_1.add(login);

reset = new JButton();

reset.addActionListener(new BookResetAction());

reset.setText("重置");

panel\_1.add(reset);

final JLabel tupianLabel = new JLabel();

ImageIcon loginIcon = CreateIcon.add("login.jpg");

tupianLabel.setIcon(loginIcon);

tupianLabel.setOpaque(true);

tupianLabel.setPreferredSize(new Dimension(260, 90));

panel.add(tupianLabel, BorderLayout.NORTH);

setVisible(true);

setResizable(false);

}

public static Operator getUser() {

return user;

}

public static void setUser(Operator user) {

LoginIFrame.user = user;

}

}

3.3 主窗体模块

图书馆管理系统的主窗体模式是由两个部分组成的。第一部分是位于主窗体上组上端的系统菜单栏，用来实现链接系统功能菜单。第二部分位于菜单栏下面的工具栏，用来提供各种常用功能链接。

3.3.1 主窗体界面设计

在主窗体中可以显示其他功能窗体，所以需要在该窗体中加入一个桌面窗体。通过将其他功能窗体加入到该窗体中。从而实现多窗体显。同时在主窗口体中，还需要添加一个main方法，该方法为系统入口的方法，通过执行该方法来执行系统。其主要代码如下：

public class MainFrame extends JFrame {

private static final JDesktopPane DESKTOP\_PANE = new JDesktopPane();

public static void main(String[] args) {

try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

new LoginIFrame();// 登录窗口

} catch (Exception ex) {

ex.printStackTrace();

}

}

public static void addIFame(JInternalFrame iframe) { // 添加子窗体的方法

DESKTOP\_PANE.add(iframe);

}

public MainFrame() {

super();

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

setLocationByPlatform(true);

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize(); // 获取屏幕size

setSize(dim); // 全屏

setTitle("图书信息管理系统");

JMenuBar menuBar = createMenu(); // 调用创建菜单栏的方法

setJMenuBar(menuBar);

JToolBar toolBar = createToolBar(); // 调用创建工具栏的方法

getContentPane().add(toolBar, BorderLayout.NORTH);

final JLabel label = new JLabel();

label.setBounds(0, 0, 0, 0);

label.setIcon(null); // 窗体背景

DESKTOP\_PANE.addComponentListener(new ComponentAdapter() {

public void componentResized(final ComponentEvent e) {

Dimension size = e.getComponent().getSize();

label.setSize(e.getComponent().getSize());

label.setText("<html><img width=" + size.width + " height=" + size.height + " src='"

+ this.getClass().getResource("/backImg.jpg") + "'></html>");

}

});

DESKTOP\_PANE.add(label, new Integer(Integer.MIN\_VALUE));

getContentPane().add(DESKTOP\_PANE);

}

/\*\*

\* 创建工具栏

\*

\* @return JToolBar

\*/

private JToolBar createToolBar() {

JToolBar toolBar = new JToolBar();// 创建工具栏的方法

toolBar.setFloatable(false);

toolBar.setBorder(new BevelBorder(BevelBorder.RAISED));

// 在工具栏中添加 图书信息添加

JButton bookAddButton = new JButton(MenuActions.BOOK\_ADD);

ImageIcon icon = CreateIcon.add("bookAddtb.jpg");// 创建图标方法

bookAddButton.setIcon(icon);

bookAddButton.setHideActionText(true);

toolBar.add(bookAddButton);

// 在工具栏中添加 图书信息修改与删除图标

JButton bookModiAndDelButton = new JButton(MenuActions.BOOK\_MODIFY);

ImageIcon bookmodiicon = CreateIcon.add("bookModiAndDeltb.jpg");// 创建图标方法

bookModiAndDelButton.setIcon(bookmodiicon);

bookModiAndDelButton.setHideActionText(true);

toolBar.add(bookModiAndDelButton);

// 在工具栏中添加 图书类别添加

JButton bookTypeAddButton = new JButton(MenuActions.BOOKTYPE\_ADD);

ImageIcon bookTypeAddicon = CreateIcon.add("bookTypeAddtb.jpg");// 创建图标方法

bookTypeAddButton.setIcon(bookTypeAddicon);

bookTypeAddButton.setHideActionText(true);

toolBar.add(bookTypeAddButton);

// 在工具栏中添加 读者信息添加

JButton readerAddButton = new JButton(MenuActions.READER\_ADD);

ImageIcon readerAddicon = CreateIcon.add("readerAddtb.jpg");// 创建图标方法

readerAddButton.setIcon(readerAddicon);

readerAddButton.setHideActionText(true);

toolBar.add(readerAddButton);

// 在工具栏中添加 读者信息修改与删除

JButton readerModiAndDelButton = new JButton(MenuActions.READER\_MODIFY);

ImageIcon readerModiAndDelicon = CreateIcon.add("readerModiAndDeltb.jpg");// 创建图标方法

readerModiAndDelButton.setIcon(readerModiAndDelicon);

readerModiAndDelButton.setHideActionText(true);

toolBar.add(readerModiAndDelButton);

// 在工具栏中添加 退出系统

JButton ExitButton = new JButton(MenuActions.EXIT);

ImageIcon Exiticon = CreateIcon.add("exittb.jpg");// 创建图标方法

ExitButton.setIcon(Exiticon);

ExitButton.setHideActionText(true);

toolBar.add(ExitButton);

return toolBar;

}

/\*\*

\* 创建菜单栏

\*/

private JMenuBar createMenu() {

JMenuBar menuBar = new JMenuBar();// 创建菜单栏的方法

// 初始化基础数据维护菜单

JMenu baseMenu = new JMenu();

baseMenu.setIcon(CreateIcon.add("jcsjwhcd.jpg"));

{

JMenu readerManagerMItem = new JMenu("读者信息管理");

readerManagerMItem.add(MenuActions.READER\_ADD);

readerManagerMItem.add(MenuActions.READER\_MODIFY);

JMenu bookTypeManageMItem = new JMenu("图书类别管理");

bookTypeManageMItem.add(MenuActions.BOOKTYPE\_ADD);

bookTypeManageMItem.add(MenuActions.BOOKTYPE\_MODIFY);

JMenu menu = new JMenu("图书信息管理");

menu.add(MenuActions.BOOK\_ADD);

menu.add(MenuActions.BOOK\_MODIFY);

baseMenu.add(readerManagerMItem);

baseMenu.add(bookTypeManageMItem);

baseMenu.add(menu);

baseMenu.addSeparator();

baseMenu.add(MenuActions.EXIT);

}

// 初始化系统维护菜单

JMenu sysManageMenu = new JMenu();

sysManageMenu.setIcon(CreateIcon.add("xtwhcd.jpg"));

JMenu userManageMItem = new JMenu("管理员设置");

userManageMItem.add(MenuActions.USER\_ADD);

userManageMItem.add(MenuActions.USER\_MODIFY);

sysManageMenu.add(MenuActions.MODIFY\_PASSWORD);

sysManageMenu.add(userManageMItem);

menuBar.add(baseMenu); // 添加基础数据维护菜单到菜单栏

menuBar.add(sysManageMenu); // 添加系统维护菜单到菜单栏

return menuBar;

}

}

3.4 图书类别管理模块

图书类别管理模块包含两个子模块，分别是图书类别添加和图书类别修改。其中图书类别添加模块用来录入新的图书类别信息，图书列别修改模块用来修改已有的图书列别信息。

3.4.1 图书类别添加模块

要添加图书类别信息，首先要在Dao中添加一个InsertbookType方法，通过该方法来接受用户输入的所有书籍类别信息，然后执行插入操作。系统界面如图3.2所示。

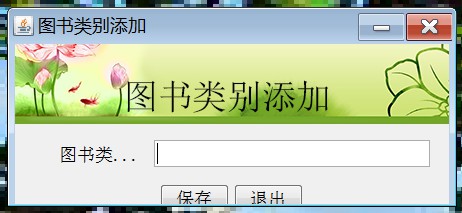


图3.2 图书类别添加

其主要方法如下：

public class BookTypeAddIFrame extends JInternalFrame {

private JTextField bookTypeName;

/\*\*

\* Create the frame

\*/

public BookTypeAddIFrame() {

super();

setIconifiable(true); // 设置窗体可最小化－－－必须

setClosable(true);

setTitle("图书类别添加");

setBounds(100, 100, 450, 200);

final JPanel panel = new JPanel();

panel.setLayout(new BorderLayout());

panel.setPreferredSize(new Dimension(360, 80));

getContentPane().add(panel, BorderLayout.NORTH);

final JLabel label\_4 = new JLabel();

ImageIcon bookTypeAddIcon = CreateIcon.add("bookTypeAdd.jpg");

label\_4.setIcon(bookTypeAddIcon);

label\_4.setPreferredSize(new Dimension(450, 80));

label\_4.setText("图书类别图片（400\*80）");

panel.add(label\_4);

final JPanel panel\_3 = new JPanel();

panel\_3.setLayout(new FlowLayout());

getContentPane().add(panel\_3, BorderLayout.CENTER);

final JLabel label\_1 = new JLabel();

label\_1.setPreferredSize(new Dimension(20, 50));

panel\_3.add(label\_1);

final JLabel label\_2 = new JLabel();

label\_2.setPreferredSize(new Dimension(90, 20));

label\_2.setText("图书类别名称：");

panel\_3.add(label\_2);

bookTypeName = new JTextField();

bookTypeName.setDocument(new MyDocument(20));

bookTypeName.setColumns(30);

panel\_3.add(bookTypeName);

final JButton button = new JButton();

button.setText("保存");

button.addActionListener(new ActionListener() {

public void actionPerformed(final ActionEvent e) {

if (bookTypeName.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "图书类别不能为空");

return;

}

int i = Dao.InsertBookType(bookTypeName.getText().trim());

if (i == 1) {

JOptionPane.showMessageDialog(null, "添加成功！");

doDefaultCloseAction();

} else {

JOptionPane.showMessageDialog(null, "图书类别名已存在，请重新输入！");

}

}

});

panel\_3.add(button);

final JButton buttonDel = new JButton();

buttonDel.setText("退出");

buttonDel.addActionListener(new ActionListener() {

public void actionPerformed(final ActionEvent e) {

doDefaultCloseAction();

}

});

panel\_3.add(buttonDel);

setVisible(true);

}

class NumberListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789." + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

}

3.4.2 图书类别修改模块

修改图书类别信息同保存图书类别信息非常类似，首先在Dao中添加一个方法UpdatebookType。通过该方法来接收图书修改表单中提交的书籍类别信息。系统运行界面如下图3.3所示。



图3.3 图书类别修改与删除

其代码如下：

public class BookTypeModiAndDelIFrame extends JInternalFrame {

private JComboBox comboBox;

private JTextField BookTypeId;

private JTable table;

private String[] array;

private String[] columnNames = { "图书类别编号", "图书类别名称" };

DefaultComboBoxModel bookTypeModel;

DefaultTableModel model;

Map map;

private Object[][] getFileStates(List list) {

Object[][] results = new Object[list.size()][columnNames.length];

for (int i = 0; i < list.size(); i++) {

BookType booktype = (BookType) list.get(i);

results[i][0] = booktype.getId();

results[i][1] = booktype.getTypeName();

}

return results;

}

/\*\*

\* Create the frame

\*/

public BookTypeModiAndDelIFrame() {

super();

setTitle("图书类别修改");

setBounds(100, 100, 500, 400);

setIconifiable(true);

setClosable(true);

final JPanel panel = new JPanel();

panel.setLayout(new BorderLayout());

getContentPane().add(panel, BorderLayout.NORTH);

final JLabel logoLabel = new JLabel();

ImageIcon bookTypeModiAndDelIcon = CreateIcon.add("booktypemodify.jpg");

logoLabel.setIcon(bookTypeModiAndDelIcon);

logoLabel.setPreferredSize(new Dimension(500, 80));

logoLabel.setText("logo");

panel.add(logoLabel);

final JPanel panel\_1 = new JPanel();

getContentPane().add(panel\_1);

final JScrollPane scrollPane = new JScrollPane();

scrollPane.setPreferredSize(new Dimension(480, 200));

panel\_1.add(scrollPane);

model = new DefaultTableModel();

Object[][] results = getFileStates(Dao.selectBookCategory());

model.setDataVector(results, columnNames);

table = new JTable();

table.setModel(model);

table.addMouseListener(new TableListener());

scrollPane.setViewportView(table);

final JPanel panel\_2 = new JPanel();

panel\_1.add(panel\_2);

final GridLayout gridLayout = new GridLayout(0, 6);

gridLayout.setVgap(5);

gridLayout.setHgap(5);

panel\_2.setLayout(gridLayout);

panel\_2.setPreferredSize(new Dimension(500, 30));

final JLabel label = new JLabel();

label.setText("类别编号：");

panel\_2.add(label);

BookTypeId = new JTextField();

BookTypeId.setFocusable(false);

panel\_2.add(BookTypeId);

final JLabel label\_1 = new JLabel();

label\_1.setText("类别名称：");

panel\_2.add(label\_1);

comboBox = new JComboBox();

bookTypeModel = (DefaultComboBoxModel) comboBox.getModel();

// 从数据库中取出图书类别

List list = Dao.selectBookCategory();

for (int i = 0; i < list.size(); i++) {

BookType booktype = (BookType) list.get(i);

Item item = new Item();

item.setId(booktype.getId());

item.setName(booktype.getTypeName());

bookTypeModel.addElement(item);

}

panel\_2.add(comboBox);

final JPanel panel\_4 = new JPanel();

panel\_1.add(panel\_4);

final JButton buttonDel = new JButton();

buttonDel.setText("删除");

buttonDel.addActionListener(new ButtonDelListener());

panel\_4.add(buttonDel);

final JButton buttonMod = new JButton();

buttonMod.setText("修改");

buttonMod.addActionListener(new ButtonAddListener());

panel\_4.add(buttonMod);

final JButton buttonExit = new JButton();

buttonExit.setText("退出");

buttonExit.addActionListener(new CloseActionListener());

panel\_4.add(buttonExit);

setVisible(true);

}

class TableListener extends MouseAdapter {

public void mouseClicked(final MouseEvent e) {

int selRow = table.getSelectedRow();

BookTypeId.setText(table.getValueAt(selRow, 0).toString().trim());

bookTypeModel.setSelectedItem(table.getValueAt(selRow, 1).toString().trim());

}

}

class ButtonAddListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

Object selectedItem = bookTypeModel.getSelectedItem();

int i = Dao.UpdatebookType(BookTypeId.getText().trim(), selectedItem.toString());

System.out.println(i);

if (i == 1) {

JOptionPane.showMessageDialog(null, "修改成功");

}

}

}

class ButtonDelListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

int i = Dao.DelbookType(BookTypeId.getText().trim());

if (i == 1) {

JOptionPane.showMessageDialog(null, "删除成功");

Object[][] results = getFileStates(Dao.selectBookCategory());

model.setDataVector(results, columnNames);

table.setModel(model);

}

}

}

class CloseActionListener implements ActionListener { // 添加关闭按钮的事件监听器

public void actionPerformed(final ActionEvent e) {

doDefaultCloseAction();

}

}

}

3.5 图书信息管理模块

图书信息管理模块包含两个子模块，分别是图书信息添加和图书信息修改。其中图书信息添加模块用来录入新的图书信息，图书信息修改模块用来修改已有的图书信息。

3.5.1 图书信息添加界面设计

图书信息添加窗体一共包含三个部分，第一部分是位于窗体最上端的图片。第二部分是位于中间的主面板，用来提供各种标签和表单元素。第三部分是位于主面板下的按钮面板，用来提供添加和关闭两个按钮。其运行界面如图3.4所示。

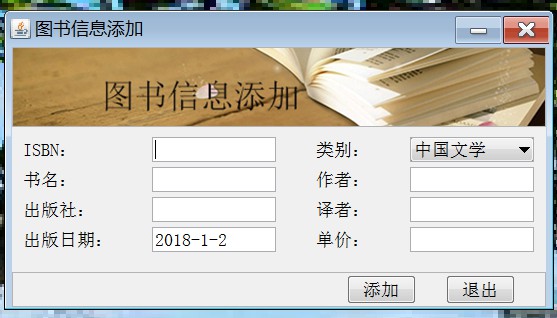


图3.4 图书信息添加

其部分实现代码如下：

public class BookAddIFrame extends JInternalFrame {

private JTextField publisher;

private JTextField price;

private JFormattedTextField pubDate;

private JTextField translator;

private JTextField writer;

private JTextField ISBN;

private JTextField bookName;

private JComboBox bookType;

private JButton buttonadd;

private JButton buttonclose;

DefaultComboBoxModel bookTypeModel;

Map map = new HashMap();

public BookAddIFrame() {

super();

final BorderLayout borderLayout = new BorderLayout();

getContentPane().setLayout(borderLayout);

setIconifiable(true); // 设置窗体可最小化

setClosable(true); // 设置窗体可关闭

setTitle("图书信息添加"); // 设置窗体标题

setBounds(100, 100, 550, 300); // 设置窗体位置和大小

final JPanel panel = new JPanel();

panel.setBorder(new EmptyBorder(10, 10, 20, 10));

final GridLayout gridLayout = new GridLayout(0, 4);

gridLayout.setVgap(5);

gridLayout.setHgap(5);

panel.setLayout(gridLayout);

getContentPane().add(panel);

final JLabel label\_2 = new JLabel();

label\_2.setText("ISBN：");

panel.add(label\_2);

ISBN = new JTextField("13位图书编号不能为空！", 13);

ISBN.setDocument(new MyDocument(13)); // 设置书号文本框最大输入值为13

ISBN.setColumns(13);

ISBN.addKeyListener(new ISBNkeyListener());

ISBN.addFocusListener(new ISBNFocusListener());

panel.add(ISBN);

final JLabel label = new JLabel();

label.setHorizontalAlignment(SwingConstants.CENTER);

label.setText("类别：");

panel.add(label);

bookType = new JComboBox();

bookTypeModel = (DefaultComboBoxModel) bookType.getModel();

// 从数据库中取出图书类别

List list = Dao.selectBookCategory();

for (int i = 0; i < list.size(); i++) {

BookType booktype = (BookType) list.get(i);

Item item = new Item();

item.setId((String) booktype.getId());

item.setName((String) booktype.getTypeName());

bookTypeModel.addElement(item);

}

panel.add(bookType);

final JLabel label\_1 = new JLabel();

label\_1.setText("书名：");

panel.add(label\_1);

bookName = new JTextField();

panel.add(bookName);

final JLabel label\_3 = new JLabel();

label\_3.setHorizontalAlignment(SwingConstants.CENTER);

label\_3.setText("作者：");

panel.add(label\_3);

writer = new JTextField();

writer.setDocument(new MyDocument(10));

panel.add(writer);

final JLabel label\_2\_1 = new JLabel();

label\_2\_1.setText("出版社：");

panel.add(label\_2\_1);

publisher = new JTextField();

publisher.setDocument(new MyDocument(10));

panel.add(publisher);

final JLabel label\_4 = new JLabel();

label\_4.setHorizontalAlignment(SwingConstants.CENTER);

label\_4.setText("译者：");

panel.add(label\_4);

translator = new JTextField();

translator.setDocument(new MyDocument(10));

panel.add(translator);

final JLabel label\_1\_1 = new JLabel();

label\_1\_1.setText("出版日期：");

panel.add(label\_1\_1);

SimpleDateFormat myfmt = new SimpleDateFormat("yyyy-MM-dd");

pubDate = new JFormattedTextField(myfmt.getDateInstance());

pubDate.setValue(new java.util.Date());

panel.add(pubDate);

final JLabel label\_3\_1 = new JLabel();

label\_3\_1.setHorizontalAlignment(SwingConstants.CENTER);

label\_3\_1.setText("单价：");

panel.add(label\_3\_1);

price = new JTextField();

price.setDocument(new MyDocument(5));

price.addKeyListener(new NumberListener());

panel.add(price);

final JPanel panel\_1 = new JPanel();

panel\_1.setBorder(new LineBorder(SystemColor.activeCaptionBorder, 1, false));

getContentPane().add(panel\_1, BorderLayout.SOUTH);

final FlowLayout flowLayout = new FlowLayout();

flowLayout.setVgap(2);

flowLayout.setHgap(30);

flowLayout.setAlignment(FlowLayout.RIGHT);

panel\_1.setLayout(flowLayout);

buttonadd = new JButton();

buttonadd.addActionListener(new addBookActionListener());

buttonadd.setText("添加");

panel\_1.add(buttonadd);

buttonclose = new JButton();

buttonclose.addActionListener(new CloseActionListener());

buttonclose.setText("退出");

panel\_1.add(buttonclose);

final JLabel label\_5 = new JLabel();

ImageIcon bookAddIcon = CreateIcon.add("newBookorderImg.jpg");

label\_5.setIcon(bookAddIcon);

label\_5.setPreferredSize(new Dimension(550, 80));

label\_5.setBorder(new LineBorder(SystemColor.activeCaptionBorder, 1, false));

getContentPane().add(label\_5, BorderLayout.NORTH);

// label\_5.setText("新书定购(LOGO图片)");

setVisible(true); // 显示窗体可关闭－－－必须在添加所有控件之后执行该语句

}

class ISBNFocusListener extends FocusAdapter {

public void focusLost(FocusEvent e) {

if (!Dao.selectBookInfo(ISBN.getText().trim()).isEmpty()) {

JOptionPane.showMessageDialog(null, "您输入的ISBN已存在！");

return;

}

}

}

class ISBNkeyListener extends KeyAdapter {

public void keyPressed(final KeyEvent e) {

if (e.getKeyCode() == 13) {

buttonadd.doClick();

}

}

}

class CloseActionListener implements ActionListener { // 添加关闭按钮的事件监听器

public void actionPerformed(final ActionEvent e) {

doDefaultCloseAction();

}

}

class addBookActionListener implements ActionListener { // 添加按钮的单击事件监听器

public void actionPerformed(final ActionEvent e) {

if (ISBN.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "图书编号不能为空！");

return;

}

if (ISBN.getText().length() != 13) {

JOptionPane.showMessageDialog(null, "请输入正确的13位图书编号！");

return;

}

if (bookName.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "书名不能为空！");

return;

}

if (writer.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "作者不能为空！");

return;

}

if (pubDate.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "出版日期不能为空！");

return;

}

if (price.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "单价不能为空！");

return;

}

String ISBNs = ISBN.getText().trim();

// 分类

Object selectedItem = bookType.getSelectedItem();

if (selectedItem == null)

return;

Item item = (Item) selectedItem;

String bookTypes = item.getId();

String translators = translator.getText().trim();

String bookNames = bookName.getText().trim();

String writers = writer.getText().trim();

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

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

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

int i = Dao.Insertbook(ISBNs, bookTypes, bookNames, writers, translators, publishers,

java.sql.Date.valueOf(pubDates), Double.parseDouble(prices));

if (i == 1) {

JOptionPane.showMessageDialog(null, "图书信息添加成功！");

doDefaultCloseAction();

}

}

}

class NumberListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789." + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

}

3.5.3 图书信息修改

要修改书籍信息，首先要在Dao中添加一个Updatebook。通过该方法来接收图书修改表单中提交的书籍信息。其运行界面如下图3.5所示：

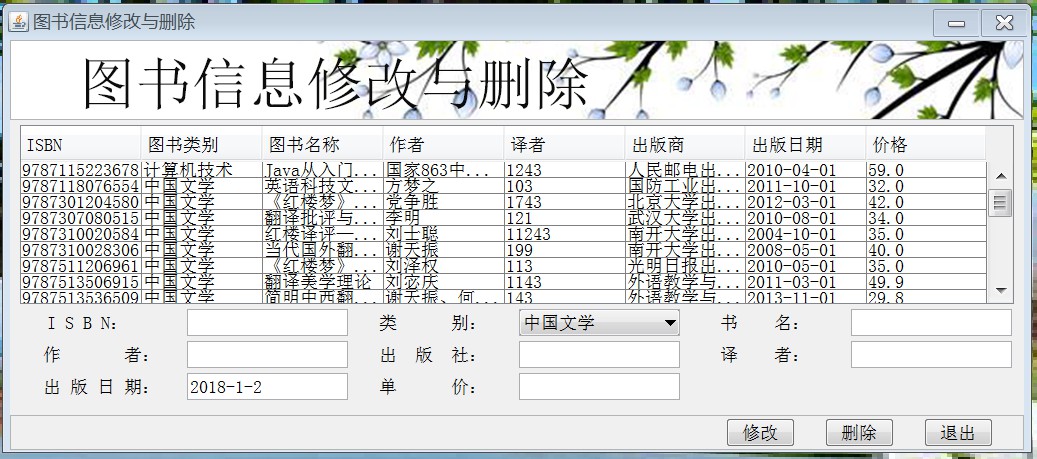


图3.5 图书信息修改与删除

其主要方法如下：

public class BookModiAndDelIFrame extends JInternalFrame {

private JTable table;

private JFormattedTextField price;

private JFormattedTextField pubDate;

private JTextField translator;

private JTextField publisher;

private JTextField writer;

private JTextField ISBN;

private JTextField bookName;

private JComboBox bookType;

DefaultComboBoxModel bookTypeModel;

private Item item;

Map map = new HashMap();

private String[] columnNames;

private Map m = MapPz.getMap();

// 取数据库中图书相关信息放入表格中

private Object[][] getFileStates(List list) {

String[] columnNames = { "ISBN", "图书类别", "图书名称", "作者", "译者", "出版商", "出版日期", "价格" };

Object[][] results = new Object[list.size()][columnNames.length];

for (int i = 0; i < list.size(); i++) {

BookInfo bookinfo = (BookInfo) list.get(i);

results[i][0] = bookinfo.getISBN();

String booktypename = String.valueOf(MapPz.getMap().get(bookinfo.getTypeid()));

results[i][1] = booktypename;

results[i][2] = bookinfo.getBookname();

results[i][3] = bookinfo.getWriter();

results[i][4] = bookinfo.getTranslator();

results[i][5] = bookinfo.getPublisher();

results[i][6] = bookinfo.getDate();

results[i][7] = bookinfo.getPrice();

}

return results;

}

public BookModiAndDelIFrame() {

super();

final BorderLayout borderLayout = new BorderLayout();

getContentPane().setLayout(borderLayout);

setIconifiable(true);

setClosable(true);

setTitle("图书信息修改与删除");

setBounds(100, 100, 1030, 450);// 移动组件并调整其大小。由 x 和 y 指定左上角的新位置，由 width 和height 指定新的大小。

final JPanel panel\_1 = new JPanel();

panel\_1.setBorder(new LineBorder(SystemColor.activeCaptionBorder, 1, false));

getContentPane().add(panel\_1, BorderLayout.SOUTH);

final FlowLayout flowLayout = new FlowLayout();

flowLayout.setVgap(2);

flowLayout.setHgap(30);

flowLayout.setAlignment(FlowLayout.RIGHT);

panel\_1.setLayout(flowLayout);

final JButton button = new JButton();

button.addActionListener(new addBookActionListener());

button.setText("修改");

panel\_1.add(button);

final JButton button\_2 = new JButton();

button\_2.addActionListener(new ActionListener() {

public void actionPerformed(final ActionEvent e) {

String ISBNs = ISBN.getText().trim();

int i = Dao.Delbook(ISBNs);

if (i == 1) {

JOptionPane.showMessageDialog(null, "删除成功");

Object[][] results = getFileStates(Dao.selectBookInfo());

// 注释代码为使用表格模型

DefaultTableModel model = new DefaultTableModel();

table.setModel(model);

model.setDataVector(results, columnNames);

}

}

});

button\_2.setText("删除");

panel\_1.add(button\_2);

final JButton button\_1 = new JButton();

button\_1.addActionListener(new ActionListener() {

public void actionPerformed(final ActionEvent e) {

doDefaultCloseAction();

}

});

button\_1.setText("退出");

panel\_1.add(button\_1);

final JLabel headLogo = new JLabel();

ImageIcon bookModiAndDelIcon = CreateIcon.add("bookmodify.jpg");

headLogo.setIcon(bookModiAndDelIcon);

headLogo.setOpaque(true);

headLogo.setPreferredSize(new Dimension(1030, 80));

headLogo.setBorder(new LineBorder(SystemColor.activeCaptionBorder, 1, false));

getContentPane().add(headLogo, BorderLayout.NORTH);

final JPanel panel\_2 = new JPanel();

final BorderLayout borderLayout\_1 = new BorderLayout();

borderLayout\_1.setVgap(5);

panel\_2.setLayout(borderLayout\_1);

panel\_2.setBorder(new EmptyBorder(5, 10, 15, 10));

getContentPane().add(panel\_2);

final JScrollPane scrollPane = new JScrollPane();

panel\_2.add(scrollPane);

Object[][] results = getFileStates(Dao.selectBookInfo());

columnNames = new String[] { "ISBN", "图书类别", "图书名称", "作者", "译者", "出版商", "出版日期", "价格" };

table = new JTable(results, columnNames);

table.setAutoResizeMode(JTable.AUTO\_RESIZE\_ALL\_COLUMNS);

// 鼠标单击表格中的内容产生事件,将表格中的内容放入文本框中

table.addMouseListener(new TableListener());

scrollPane.setViewportView(table);

final JPanel bookPanel = new JPanel();

panel\_2.add(bookPanel, BorderLayout.SOUTH);

final GridLayout gridLayout = new GridLayout(0, 6);

gridLayout.setVgap(5);

gridLayout.setHgap(5);

bookPanel.setLayout(gridLayout);

final JLabel label\_2 = new JLabel();

label\_2.setHorizontalAlignment(SwingConstants.CENTER);

label\_2.setText("I S B N： ");

bookPanel.add(label\_2);

ISBN = new JTextField();

ISBN.setDocument(new MyDocument(13));

bookPanel.add(ISBN);

final JLabel label = new JLabel();

label.setHorizontalAlignment(SwingConstants.CENTER);

label.setText("类 别：");

bookPanel.add(label);

bookType = new JComboBox();

bookTypeModel = (DefaultComboBoxModel) bookType.getModel();

List list = Dao.selectBookCategory();

for (int i = 0; i < list.size(); i++) {

BookType booktype = (BookType) list.get(i);

item = new Item();

item.setId((String) booktype.getId());

item.setName((String) booktype.getTypeName());

bookTypeModel.addElement(item);

map.put(item.getId(), item);

}

bookPanel.add(bookType);

final JLabel label\_1 = new JLabel();

label\_1.setHorizontalAlignment(SwingConstants.CENTER);

label\_1.setText("书 名：");

bookPanel.add(label\_1);

bookName = new JTextField();

bookPanel.add(bookName);

final JLabel label\_3 = new JLabel();

label\_3.setHorizontalAlignment(SwingConstants.CENTER);

label\_3.setText("作 者：");

bookPanel.add(label\_3);

writer = new JTextField();

bookPanel.add(writer);

final JLabel label\_2\_1 = new JLabel();

label\_2\_1.setHorizontalAlignment(SwingConstants.CENTER);

label\_2\_1.setText("出 版 社：");

bookPanel.add(label\_2\_1);

publisher = new JTextField();

bookPanel.add(publisher);

final JLabel label\_4 = new JLabel();

label\_4.setHorizontalAlignment(SwingConstants.CENTER);

label\_4.setText("译 者：");

bookPanel.add(label\_4);

translator = new JTextField();

bookPanel.add(translator);

final JLabel label\_1\_1 = new JLabel();

label\_1\_1.setHorizontalAlignment(SwingConstants.CENTER);

label\_1\_1.setText("出 版 日 期：");

bookPanel.add(label\_1\_1);

SimpleDateFormat myfmt = new SimpleDateFormat("yyyy-MM-dd");

pubDate = new JFormattedTextField(myfmt.getDateInstance());

pubDate.setValue(new java.util.Date());

bookPanel.add(pubDate);

final JLabel label\_3\_1 = new JLabel();

label\_3\_1.setHorizontalAlignment(SwingConstants.CENTER);

label\_3\_1.setText("单 价：");

bookPanel.add(label\_3\_1);

price = new JFormattedTextField();

price.addKeyListener(new NumberListener());

bookPanel.add(price);

setVisible(true);

}

class TableListener extends MouseAdapter {

public void mouseClicked(final MouseEvent e) {

String ISBNs, typeids, bookNames, writers, translators, publishers, dates, prices;

int selRow = table.getSelectedRow();

ISBNs = table.getValueAt(selRow, 0).toString().trim();

typeids = table.getValueAt(selRow, 1).toString().trim();

bookNames = table.getValueAt(selRow, 2).toString().trim();

writers = table.getValueAt(selRow, 3).toString().trim();

translators = table.getValueAt(selRow, 4).toString().trim();

publishers = table.getValueAt(selRow, 5).toString().trim();

dates = table.getValueAt(selRow, 6).toString().trim();

prices = table.getValueAt(selRow, 7).toString().trim();

ISBN.setText(ISBNs);

// bookTypeModel.setSelectedItem(item);

bookTypeModel.setSelectedItem(typeids);

bookName.setText(bookNames);

writer.setText(writers);

translator.setText(translators);

publisher.setText(publishers);

pubDate.setText(dates);

price.setText(prices);

}

}

class addBookActionListener implements ActionListener {

public void actionPerformed(final ActionEvent e) {

// 修改图书信息表

if (ISBN.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "您输入的ISBN号码无效！");

return;

}

if (ISBN.getText().length() != 13) {

JOptionPane.showMessageDialog(null, "请输入有效的13位ISBN号码！");

return;

}

if (bookName.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "书名不能为空！");

return;

}

if (writer.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "作者不能为空！");

return;

}

if (publisher.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "出版社不能为空！");

return;

}

// 日期与单价进行数字验证代码？

if (pubDate.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "出版日期不能为空！");

return;

}

if (price.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "单价不能为空！");

return;

}

String ISBNs = ISBN.getText().trim();

// 分类

Object selectedItem = bookTypeModel.getSelectedItem();

if (selectedItem == null)

return;

Item item = (Item) selectedItem;

String bookTypes = item.getId();

String translators = translator.getText().trim();

String bookNames = bookName.getText().trim();

String writers = writer.getText().trim();

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

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

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

int i = Dao.Updatebook(ISBNs, bookTypes, bookNames, writers, translators, publishers,

Date.valueOf(pubDates), Double.parseDouble(prices));

System.out.println(i);

if (i == 1) {

JOptionPane.showMessageDialog(null, "修改成功");

Object[][] results = getFileStates(Dao.selectBookInfo());

// 注释代码为使用表格模型

DefaultTableModel model = new DefaultTableModel();

// DefaultTableModel model = (DefaultTableModel)

// table.getModel();

table.setModel(model);

model.setDataVector(results, columnNames);

}

}

}

class NumberListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789." + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

}

3.6 读者信息管理模块

读者信息管理模块包含两个子模块，分别是读者信息添加和读者信息修改与删除。其中读者信息添加模块用来录入新的读者信息，读者信息修改与删除模块用来修改和删除已有的读者信息。

3.6.1 读者信息添加

要添加读者信息，首先要在Dao中添加一个InsertReader方法，通过该方法来接受用户输入的所有读者信息，然后执行插入操作。其系统运行界面如图3.6所示：

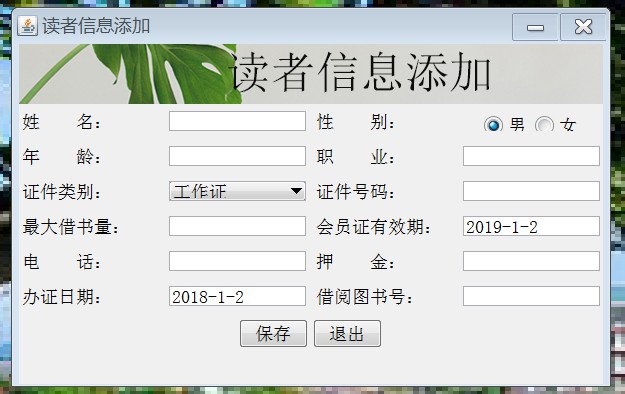


图3.6 读者信息添加

其主要方法如下：

public class ReaderAddIFrame extends JInternalFrame {

private JTextField ISBN;

private ButtonGroup buttonGroup = new ButtonGroup();

private JFormattedTextField keepmoney;

private JTextField tel;

private JFormattedTextField date;

private JFormattedTextField maxnumber;

private JFormattedTextField bztime;

private JTextField zjnumber;

private JComboBox comboBox;

private JTextField zy;

private JTextField age;

private JTextField readername;

DefaultComboBoxModel comboBoxModel;

String[] array;

/\*\*

\* Create the frame

\*/

public ReaderAddIFrame() {

super();

setTitle("读者信息添加"); // 设置窗体标题－－－必须

setIconifiable(true); // 设置窗体可最小化－－－必须

setClosable(true); // 设置窗体可关闭－－－必须

setBounds(100, 100, 600, 380);

final JLabel logoLabel = new JLabel();

ImageIcon readerAddIcon = CreateIcon.add("readerAdd.jpg");

logoLabel.setIcon(readerAddIcon);

logoLabel.setOpaque(true);

logoLabel.setBackground(Color.CYAN);

logoLabel.setPreferredSize(new Dimension(600, 60));

getContentPane().add(logoLabel, BorderLayout.NORTH);

final JPanel panel = new JPanel();

panel.setLayout(new FlowLayout());

getContentPane().add(panel);

final JPanel panel\_1 = new JPanel();

final GridLayout gridLayout = new GridLayout(0, 4);

gridLayout.setVgap(15);

gridLayout.setHgap(10);

panel\_1.setLayout(gridLayout);

panel\_1.setPreferredSize(new Dimension(580, 200));

panel.add(panel\_1);

final JLabel label\_2 = new JLabel();

label\_2.setText("姓 名：");

panel\_1.add(label\_2);

readername = new JTextField();

readername.setDocument(new MyDocument(10));

panel\_1.add(readername);

final JLabel label\_3 = new JLabel();

label\_3.setText("性 别：");

panel\_1.add(label\_3);

final JPanel label\_13 = new JPanel();

final FlowLayout flowLayout = new FlowLayout();

flowLayout.setHgap(0);

flowLayout.setVgap(0);

label\_13.setLayout(flowLayout);

panel\_1.add(label\_13);

final JRadioButton radioButton1 = new JRadioButton();

label\_13.add(radioButton1);

radioButton1.setSelected(true);

buttonGroup.add(radioButton1);

radioButton1.setText("男");

final JRadioButton radioButton2 = new JRadioButton();

label\_13.add(radioButton2);

buttonGroup.add(radioButton2);

radioButton2.setText("女");

final JLabel label\_4 = new JLabel();

label\_4.setText("年 龄：");

panel\_1.add(label\_4);

age = new JTextField();

age.setDocument(new MyDocument(2));// 设置书号文本框最大输入值为2

age.addKeyListener(new NumberListener());

panel\_1.add(age);

final JLabel label\_5 = new JLabel();

label\_5.setText("职 业：");

panel\_1.add(label\_5);

zy = new JTextField();

zy.setDocument(new MyDocument(30));

panel\_1.add(zy);

final JLabel label\_6 = new JLabel();

label\_6.setText("证件类别：");

panel\_1.add(label\_6);

comboBox = new JComboBox();

// comboBoxModel=(DefaultComboBoxModel)comboBox.getModel();

array = new String[] { "身份证", "军人证", "学生证", "工作证" };

comboBox.setModel(new DefaultComboBoxModel(array));

for (int i = 1; i < array.length; i++) {

comboBox.setSelectedIndex(i);

comboBox.setSelectedItem(array);

}

panel\_1.add(comboBox);

final JLabel label\_7 = new JLabel();

label\_7.setText("证件号码：");

panel\_1.add(label\_7);

zjnumber = new JTextField();

zjnumber.setDocument(new MyDocument(18));

zjnumber.addKeyListener(new NumberListener());

panel\_1.add(zjnumber);

final JLabel label\_9 = new JLabel();

label\_9.setText("最大借书量：");

panel\_1.add(label\_9);

maxnumber = new JFormattedTextField();

maxnumber.setDocument(new MyDocument(2));

maxnumber.addKeyListener(new NumberListener());

panel\_1.add(maxnumber);

final JLabel label\_10 = new JLabel();

label\_10.setText("会员证有效期：");

panel\_1.add(label\_10);

SimpleDateFormat myfmt = new SimpleDateFormat("yyyy-MM-dd");

date = new JFormattedTextField(myfmt.getDateInstance());

java.util.Date date2 = new java.util.Date();

date2.setDate(date2.getDate() + 365);

date.setValue(date2);

date.addKeyListener(new DateListener());

panel\_1.add(date);

final JLabel label\_11 = new JLabel();

label\_11.setText("电 话：");

panel\_1.add(label\_11);

tel = new JTextField();

tel.addKeyListener(new TelListener());

tel.setDocument(new MyDocument(11));

panel\_1.add(tel);

final JLabel label\_12 = new JLabel();

label\_12.setText("押 金：");

panel\_1.add(label\_12);

keepmoney = new JFormattedTextField();

keepmoney.addKeyListener(new KeyAdapter() {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789" + (char) 8;// 只允许输入数字与退格键

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

if (keepmoney.getText().length() > 2 || keepmoney.getText().length() < 0) {

e.consume();

}

}

});

panel\_1.add(keepmoney);

final JLabel label = new JLabel();

label.setText("办证日期：");

panel\_1.add(label);

bztime = new JFormattedTextField(myfmt.getDateInstance());

bztime.setValue(new java.util.Date());

bztime.addKeyListener(new DateListener());

panel\_1.add(bztime);

final JLabel label\_1 = new JLabel();

label\_1.setText("借阅图书号：");

panel\_1.add(label\_1);

ISBN = new JTextField();

ISBN.setDocument(new MyDocument(13));

panel\_1.add(ISBN);

final JPanel panel\_2 = new JPanel();

panel\_2.setPreferredSize(new Dimension(450, 100));

panel.add(panel\_2);

final JButton save = new JButton();

panel\_2.add(save);

save.setText("保存");

save.addActionListener(new ButtonAddListener(radioButton1));

final JButton back = new JButton();

panel\_2.add(back);

back.setText("退出");

back.addActionListener(new CloseActionListener());

setVisible(true);

}

class DateListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

if (bztime.getText().isEmpty()) {

JOptionPane.showMessageDialog(null, "时间格式请使用如\"2007-05-10\"格式");

}

}

}

class NumberListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789" + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

class ButtonAddListener implements ActionListener {

private final JRadioButton button1;

ButtonAddListener(JRadioButton button1) {

this.button1 = button1;

}

public void actionPerformed(final ActionEvent e) {

if (readername.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "姓名不能为空！");

return;

}

if (age.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "年龄不能为空！");

return;

}

if (zjnumber.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "证件号码不能为空！");

return;

}

if (zjnumber.getText().length() != 13) {

JOptionPane.showMessageDialog(null, "请输入有效的13位证件号码！");

return;

}

if (keepmoney.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "押金不能为空！");

return;

}

if (zy.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "职业不能为空！");

return;

}

if (zy.getText().length() > 20) {

JOptionPane.showMessageDialog(null, "职业不能超过20个字符");

return;

}

if (ISBN.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "借阅图书号不能为空");

return;

}

if (ISBN.getText().length() != 13) {

JOptionPane.showMessageDialog(null, "借阅图书号必须为13位");

return;

}

if (tel.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "联系电话不能为空！");

return;

}

if (tel.getText().length() > 11 || tel.getText().length() < 0) {

JOptionPane.showMessageDialog(null, "联系电话位数小于11位");

return;

}

if (maxnumber.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "最大借书量不能为空！");

return;

}

if (maxnumber.getText().length() > 2 || tel.getText().length() < 0) {

JOptionPane.showMessageDialog(null, "最大借书量为2位数字");

return;

}

if (bztime.getText().isEmpty() || date.getText().isEmpty()) {

JOptionPane.showMessageDialog(null, "时间格式请使用\"2007-05-10\"格式");

return;

}

String sex = "1";

if (!button1.isSelected()) {

sex = "2";

}

String zj = String.valueOf(comboBox.getSelectedIndex());

System.out.println(comboBox.getSelectedIndex());

int i = Dao.InsertReader(readername.getText().trim(), sex.trim(), age.getText().trim(),

zjnumber.getText().trim(), Date.valueOf(date.getText().trim()), maxnumber.getText().trim(),

tel.getText().trim(), Double.valueOf(keepmoney.getText().trim()), zj, zy.getText().trim(),

Date.valueOf(bztime.getText().trim()), ISBN.getText().trim());

System.out.println(i);

if (i == 1) {

JOptionPane.showMessageDialog(null, "添加成功！");

doDefaultCloseAction();

}

}

}

class TelListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789-" + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

class CloseActionListener implements ActionListener { // 添加关闭按钮的事件监听器

public void actionPerformed(final ActionEvent e) {

doDefaultCloseAction();

}

}

}

3.6.2 读者信息修改与删除

修改与删除读者信息同保存读者信息非常类似，首先在Dao中添加两个方法UpdateReader及DelReader。通过这两个方法来接收图读者修改表单中提交的读者信息。其运行界面如下图3.7所示：



图3.7 读者信息修改与删除

其主要代码如下：

public class ReaderModiAndDelIFrame extends JInternalFrame {

private JTextField keepmoney;

private ButtonGroup buttonGroup = new ButtonGroup();

private JTable table;

private JTextField ISBN;

private JTextField zy;

private JTextField tel;

private JTextField date;

private JTextField maxnumber;

private JTextField bztime;

private JTextField zjnumber;

private JComboBox comboBox;

private JTextField age;

private JTextField readername;

private JRadioButton JRadioButton1;

private JRadioButton JRadioButton2;

private String[] columnNames = { "读者名称", "读者性别", "读者年龄", "证件号码", "会员证有效日期", "最大借书量", "电话", "押金", "证件", "职业",

"借阅图书号", "读者办证时间" };

private String[] array = new String[] { "身份证", "军人证", "学生证", "工作证" };

String id;

/\*\*

\* Create the frame

\*/

private Object[][] getFileStates(List list) {

Object[][] results = new Object[list.size()][columnNames.length];

for (int i = 0; i < list.size(); i++) {

Reader reader = (Reader) list.get(i);

// results[i][0]=reader.getId();

results[i][0] = reader.getName();

String sex;

if (reader.getSex().equals("1")) {

sex = "男";

} else

sex = "女";

results[i][1] = sex;

results[i][2] = reader.getAge();

results[i][3] = reader.getIdentityCard();

results[i][4] = reader.getDate();

results[i][5] = reader.getMaxNum();

results[i][6] = reader.getTel();

results[i][7] = reader.getKeepMoney();

results[i][8] = reader.getZj();

results[i][9] = reader.getZy();

results[i][10] = reader.getISBN();

results[i][11] = reader.getBztime();

}

return results;

}

public ReaderModiAndDelIFrame() {

super();

setIconifiable(true);

setClosable(true);

setTitle("读者信息修改与删除");

setBounds(50, 0, 600, 500);

final JPanel panel = new JPanel();

panel.setLayout(new BorderLayout());

panel.setPreferredSize(new Dimension(400, 80));

getContentPane().add(panel, BorderLayout.NORTH);

final JLabel logoLabel = new JLabel();

ImageIcon readerModiAndDelIcon = CreateIcon.add("readerModiAndDel.jpg");

logoLabel.setIcon(readerModiAndDelIcon);

logoLabel.setBackground(Color.CYAN);

logoLabel.setOpaque(true);

logoLabel.setPreferredSize(new Dimension(600, 80));

panel.add(logoLabel);

logoLabel.setText("读者信息修改logo（400\*80）");

final JPanel panel\_1 = new JPanel();

panel\_1.setLayout(new BorderLayout());

getContentPane().add(panel\_1);

final JScrollPane scrollPane = new JScrollPane();

scrollPane.setPreferredSize(new Dimension(0, 100));

panel\_1.add(scrollPane, BorderLayout.NORTH);

final DefaultTableModel model = new DefaultTableModel();

Object[][] results = getFileStates(Dao.selectReader());

model.setDataVector(results, columnNames);

table = new JTable();

table.setModel(model);

scrollPane.setViewportView(table);

table.setAutoResizeMode(JTable.AUTO\_RESIZE\_OFF);

table.addMouseListener(new TableListener());

final JPanel panel\_2 = new JPanel();

final GridLayout gridLayout = new GridLayout(0, 4);

gridLayout.setVgap(9);

panel\_2.setLayout(gridLayout);

panel\_2.setPreferredSize(new Dimension(0, 220));

panel\_1.add(panel\_2, BorderLayout.SOUTH);

final JLabel label\_1 = new JLabel();

label\_1.setText(" 姓 名：");

panel\_2.add(label\_1);

readername = new JTextField();

readername.setDocument(new MyDocument(10));

panel\_2.add(readername);

final JLabel label\_2 = new JLabel();

label\_2.setText(" 性 别：");

panel\_2.add(label\_2);

final JPanel panel\_3 = new JPanel();

final FlowLayout flowLayout\_1 = new FlowLayout();

flowLayout\_1.setVgap(0);

panel\_3.setLayout(flowLayout\_1);

panel\_2.add(panel\_3);

JRadioButton1 = new JRadioButton();

JRadioButton1.setSelected(true);

buttonGroup.add(JRadioButton1);

panel\_3.add(JRadioButton1);

JRadioButton1.setText("男");

JRadioButton2 = new JRadioButton();

buttonGroup.add(JRadioButton2);

panel\_3.add(JRadioButton2);

JRadioButton2.setText("女");

final JLabel label\_3 = new JLabel();

label\_3.setText(" 年 龄：");

panel\_2.add(label\_3);

age = new JTextField();

age.setDocument(new MyDocument(2));

age.addKeyListener(new NumberListener());

panel\_2.add(age);

final JLabel label\_5 = new JLabel();

label\_5.setText(" 职 业：");

panel\_2.add(label\_5);

zy = new JTextField();

zy.setDocument(new MyDocument(30));

panel\_2.add(zy);

final JLabel label = new JLabel();

label.setText(" 有效证件：");

panel\_2.add(label);

comboBox = new JComboBox();

comboBox.setModel(new DefaultComboBoxModel(array));

for (int i = 1; i < array.length; i++) {

comboBox.setSelectedIndex(i);

comboBox.setSelectedItem(array);

}

panel\_2.add(comboBox);

final JLabel label\_6 = new JLabel();

label\_6.setText(" 证件号码：");

panel\_2.add(label\_6);

zjnumber = new JTextField();

zjnumber.setDocument(new MyDocument(18));

zjnumber.addKeyListener(new NumberListener());

panel\_2.add(zjnumber);

final JLabel label\_7 = new JLabel();

label\_7.setText(" 办证日期：");

panel\_2.add(label\_7);

SimpleDateFormat myfmt = new SimpleDateFormat("yyyy-MM-dd");

bztime = new JFormattedTextField(myfmt.getDateInstance());

panel\_2.add(bztime);

final JLabel label\_9 = new JLabel();

label\_9.setText(" 最大借书量：");

panel\_2.add(label\_9);

maxnumber = new JTextField();

maxnumber.addKeyListener(new NumberListener());

panel\_2.add(maxnumber);

final JLabel label\_13 = new JLabel();

label\_13.setText(" 会员证有效日期：");

panel\_2.add(label\_13);

date = new JFormattedTextField(myfmt.getDateInstance());

panel\_2.add(date);

final JLabel label\_8 = new JLabel();

label\_8.setText(" 电 话：");

panel\_2.add(label\_8);

tel = new JFormattedTextField();

tel.addKeyListener(new TelListener());

tel.setDocument(new MyDocument(11));

panel\_2.add(tel);

final JLabel label\_14 = new JLabel();

label\_14.setText(" 押 金：");

panel\_2.add(label\_14);

keepmoney = new JTextField();

keepmoney.addKeyListener(new KeepmoneyListener());

panel\_2.add(keepmoney);

final JLabel label\_4 = new JLabel();

label\_4.setText(" 借阅图书号：");

panel\_2.add(label\_4);

ISBN = new JTextField();

ISBN.setEditable(false);

ISBN.setDocument(new MyDocument(13));

panel\_2.add(ISBN);

final JPanel panel\_4 = new JPanel();

panel\_4.setMaximumSize(new Dimension(0, 0));

final FlowLayout flowLayout = new FlowLayout();

flowLayout.setVgap(0);

flowLayout.setHgap(4);

panel\_4.setLayout(flowLayout);

panel\_2.add(panel\_4);

final JButton button = new JButton();

button.setHorizontalTextPosition(SwingConstants.CENTER);

panel\_4.add(button);

button.setText("修改");

button.addActionListener(new ModiButtonListener(model));

final JButton buttonDel = new JButton();

panel\_4.add(buttonDel);

buttonDel.setText("删除");

buttonDel.addActionListener(new DelButtonListener(model));

setVisible(true);

}

class TableListener extends MouseAdapter {

public void mouseClicked(final MouseEvent e) {

int selRow = table.getSelectedRow();

// id=table.getValueAt(selRow, 0).toString().trim();

readername.setText(table.getValueAt(selRow, 0).toString().trim());

if (table.getValueAt(selRow, 1).toString().trim().equals("男"))

JRadioButton1.setSelected(true);

else

JRadioButton2.setSelected(true);

age.setText(table.getValueAt(selRow, 2).toString().trim());

zjnumber.setText(table.getValueAt(selRow, 3).toString().trim());

date.setText(table.getValueAt(selRow, 4).toString().trim());

maxnumber.setText(table.getValueAt(selRow, 5).toString().trim());

tel.setText(table.getValueAt(selRow, 6).toString().trim());

keepmoney.setText(table.getValueAt(selRow, 7).toString().trim());

comboBox.setSelectedItem(table.getValueAt(selRow, 8).toString().trim());

zy.setText(table.getValueAt(selRow, 9).toString().trim());

ISBN.setText(table.getValueAt(selRow, 10).toString().trim());

bztime.setText(table.getValueAt(selRow, 11).toString().trim());

}

}

final class NumberListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789" + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

private final class DelButtonListener implements ActionListener {

private final DefaultTableModel model;

private DelButtonListener(DefaultTableModel model) {

this.model = model;

}

public void actionPerformed(final ActionEvent e) {

int i = Dao.DelReader(ISBN.getText().trim());

if (i == 1) {

JOptionPane.showMessageDialog(null, "删除成功");

Object[][] results = getFileStates(Dao.selectReader());

model.setDataVector(results, columnNames);

table.setModel(model);

}

}

}

class ModiButtonListener implements ActionListener {

private final DefaultTableModel model;

ModiButtonListener(DefaultTableModel model) {

this.model = model;

}

public void actionPerformed(final ActionEvent e) {

if (readername.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "读者姓名不能为空");

return;

}

if (age.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "读者年龄不能为空");

return;

}

if (zjnumber.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "证件号码不能为空");

return;

}

if (keepmoney.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "押金不能为空");

return;

}

if (zy.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "职业不能为空");

return;

}

if (ISBN.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "借阅图书号不能为空");

return;

}

if (ISBN.getText().length() != 13) {

JOptionPane.showMessageDialog(null, "借阅图书号必须为13位");

return;

}

if (bztime.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "办证时间不能为空");

return;

}

if (tel.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "电话号码不能为空");

return;

}

if (tel.getText().length() > 11 || tel.getText().length() < 0) {

JOptionPane.showMessageDialog(null, "电话号码位数小于11位");

return;

}

if (maxnumber.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "最大借书量不能为空");

return;

}

if (maxnumber.getText().length() > 2 || tel.getText().length() < 0) {

JOptionPane.showMessageDialog(null, "最大借书量为两位数字");

return;

}

String sex = "1";

if (!JRadioButton1.isSelected()) {

sex = "2";

}

String zj = String.valueOf(comboBox.getSelectedIndex());

System.out.println(comboBox.getSelectedIndex());

int i = Dao.UpdateReader(id, readername.getText().trim(), sex, age.getText().trim(),

zjnumber.getText().trim(), Date.valueOf(date.getText().trim()), maxnumber.getText().trim(),

tel.getText().trim(), Double.valueOf(keepmoney.getText().trim()), zj, zy.getText().trim(),

Date.valueOf(bztime.getText().trim()), ISBN.getText().trim());

System.out.println(i);

if (i == 1) {

JOptionPane.showMessageDialog(null, "修改成功");

Object[][] results = getFileStates(Dao.selectReader());

model.setDataVector(results, columnNames);

table.setModel(model);

}

}

}

class TelListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789-" + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

class KeepmoneyListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789" + (char) 8;// 只允许输入数字与退格键

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

if (keepmoney.getText().length() > 2 || keepmoney.getText().length() < 0) {

e.consume();

}

}

}

}

3.9 系统维护模块

系统维护模块包含两个子模块，分别是修改密码和用户管理。其中修改密码模块用来更改管理员登陆系统密码，用户管理模块用来添加新的用户及对用户进行修改和删除。

3.9.1 修改密码

本图书馆管理系统是需要密码才能进入的，而且每个操作员都可以根据自己的喜好更改自己的密码，此时，需要在Dao中添加Updatepass方法，通过该方法来进行密码的更改。其系统运行界面如图3.8所示：

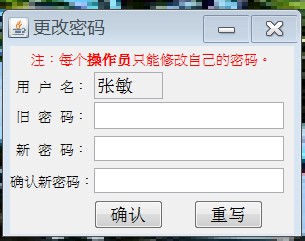


图3.8 更改密码

3.9.2 管理员信息添加

要想添加新的管理员时，需要将新的管理员信息录入到用户表中。此时，需要在Dao中添加Insertoperator方法，通过该方法录入新的用户信息。其运行界面如图3.9所示。



图3.9 管理员增加

其主要方法如下：

public class UserAddIFrame extends JInternalFrame {

// private JTextField textField\_5;

private JTextField textField\_4;

private JFormattedTextField textField\_3;

private JTextField textField\_2;

private JTextField textField\_1;

private JTextField textField;

private JPasswordField textField\_6;

private JButton button;

private ButtonGroup buttonGroup = new ButtonGroup();

final JRadioButton radioButton1, radioButton2;

/\*\*

\* Create the frame

\*/

public UserAddIFrame() {

super();

setIconifiable(true);

setClosable(true);

setTitle("管理员信息添加");

setBounds(120, 120, 450, 350);

final JPanel panel = new JPanel();

getContentPane().add(panel);

button = new JButton();

button.setText("保存");

panel.add(button);

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

if (arg0.getSource() == button) {

if (textField.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "管理员姓名不能为空！");

return;

}

if (textField.getText().length() > 12) {

JOptionPane.showMessageDialog(null, "管理员姓名不能大于12位！");

return;

}

if (textField\_1.getText().length() > 18) {

JOptionPane.showMessageDialog(null, "证件号码必须小于18位！");

return;

}

if (textField\_2.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "年龄不能为空！");

return;

}

if (textField\_3.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "办证日期不能为空！");

return;

}

if (textField\_4.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "联系电话不能为空！");

return;

}

if (textField\_4.getText().length() != 11) {

JOptionPane.showMessageDialog(null, "电话号必须是11位！");

return;

}

if (textField\_6.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "密码不能为空！");

return;

}

if (textField\_6.getText().length() > 15) {

JOptionPane.showMessageDialog(null, "密码不能大于15位！");

return;

}

String username = textField.getText();

String identityCard = textField\_1.getText();

int age = Integer.parseInt(textField\_2.getText());

String date = textField\_3.getText();

String tel = textField\_4.getText();

// int yajin = Integer.parseInt(textField\_5.getText());

String password = textField\_6.getText();

String sex = "1";

if (!radioButton1.isSelected()) {

sex = "2";

}

int admin = 1;

int i = Dao.Insertoperator(username, sex, age, identityCard, java.sql.Date.valueOf(date), tel,

password, admin);

if (i == 1) {

JOptionPane.showMessageDialog(null, "添加成功！");

doDefaultCloseAction();

}

}

}

});

final JButton button\_1 = new JButton();

button\_1.setText("退出");

panel.add(button\_1);

button\_1.addActionListener(new CloseActionListener());

setVisible(true);

final JPanel panel\_1 = new JPanel();

panel\_1.setPreferredSize(new Dimension(0, 250));

getContentPane().add(panel\_1, BorderLayout.NORTH);

final JPanel panel\_2 = new JPanel();

final GridLayout gridLayout = new GridLayout(0, 2);

gridLayout.setVgap(10);

panel\_2.setLayout(gridLayout);

panel\_2.setPreferredSize(new Dimension(280, 220));

panel\_1.add(panel\_2);

final JLabel label = new JLabel();

label.setText("用户姓名：");

panel\_2.add(label);

textField = new JTextField();

panel\_2.add(textField);

final JLabel label\_1 = new JLabel();

label\_1.setText("性 别：");

panel\_2.add(label\_1);

final JPanel label\_8 = new JPanel();

final FlowLayout flowLayout = new FlowLayout();

flowLayout.setHgap(0);

flowLayout.setVgap(0);

label\_8.setLayout(flowLayout);

panel\_1.add(label\_8);

radioButton1 = new JRadioButton();

label\_8.add(radioButton1);

radioButton1.setSelected(true);

buttonGroup.add(radioButton1);

radioButton1.setText("男");

radioButton1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

}

});

radioButton2 = new JRadioButton();

label\_8.add(radioButton2);

radioButton2.setSelected(true);

buttonGroup.add(radioButton2);

radioButton2.setText("女");

panel\_2.add(label\_8);

final JLabel label\_9 = new JLabel();

label\_9.setText("证件号码：");

panel\_2.add(label\_9);

textField\_1 = new JTextField();

textField\_1.setDocument(new MyDocument(18));

textField\_1.setColumns(18);

textField\_1.addKeyListener(new NumberListener());

panel\_2.add(textField\_1);

final JLabel label\_2 = new JLabel();

label\_2.setText("年 龄：");

panel\_2.add(label\_2);

textField\_2 = new JTextField();

textField\_2.setDocument(new MyDocument(2));

textField\_2.setColumns(2);

textField\_2.addKeyListener(new NumberListener());

panel\_2.add(textField\_2);

final JLabel label\_3 = new JLabel();

label\_3.setText("办证日期：");

panel\_2.add(label\_3);

SimpleDateFormat myfmt = new SimpleDateFormat("yyyy-MM-dd");

textField\_3 = new JFormattedTextField(myfmt.getDateInstance());

textField\_3.setValue(new java.util.Date());

panel\_2.add(textField\_3);

final JLabel label\_4 = new JLabel();

label\_4.setText("联系电话：");

panel\_2.add(label\_4);

textField\_4 = new JTextField("电话号必须是11位", 11);

textField\_4.setDocument(new MyDocument(11));

textField\_4.setColumns(11);

textField\_4.addKeyListener(new NumberListener());

panel\_2.add(textField\_4);

// textField\_5 = new JTextField();

// textField\_5.addKeyListener(new NumberListener());

// panel\_2.add(textField\_5);

// textField\_5.setColumns(20);

final JLabel label\_5 = new JLabel();

label\_5.setText("密 码：");

panel\_2.add(label\_5);

textField\_6 = new JPasswordField();

panel\_2.add(textField\_6);

setVisible(true);

}

class CloseActionListener implements ActionListener { // 添加关闭按钮的事件监听器

public void actionPerformed(final ActionEvent e) {

doDefaultCloseAction();

}

}

}

class NumberListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789." + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

3.9.3 用户信息修改与删除

要想实现用户信息修改与删除，需要在Dao中添加Updateuser和Deluser方法，通过这两个方法实现用户信息的修改与删除。其运行结果如下图3.10所示：



图3.10 管理员信息修改与删除

其方法如下：

public class ReaderModiAndDelIFrame extends JInternalFrame {

private JTextField keepmoney;

private ButtonGroup buttonGroup = new ButtonGroup();

private JTable table;

private JTextField ISBN;

private JTextField zy;

private JTextField tel;

private JTextField date;

private JTextField maxnumber;

private JTextField bztime;

private JTextField zjnumber;

private JComboBox comboBox;

private JTextField age;

private JTextField readername;

private JRadioButton JRadioButton1;

private JRadioButton JRadioButton2;

private String[] columnNames = { "读者名称", "读者性别", "读者年龄", "证件号码", "会员证有效日期", "最大借书量", "电话", "押金", "证件", "职业",

"借阅图书号", "读者办证时间" };

private String[] array = new String[] { "身份证", "军人证", "学生证", "工作证" };

String id;

/\*\*

\* Create the frame

\*/

private Object[][] getFileStates(List list) {

Object[][] results = new Object[list.size()][columnNames.length];

for (int i = 0; i < list.size(); i++) {

Reader reader = (Reader) list.get(i);

// results[i][0]=reader.getId();

results[i][0] = reader.getName();

String sex;

if (reader.getSex().equals("1")) {

sex = "男";

} else

sex = "女";

results[i][1] = sex;

results[i][2] = reader.getAge();

results[i][3] = reader.getIdentityCard();

results[i][4] = reader.getDate();

results[i][5] = reader.getMaxNum();

results[i][6] = reader.getTel();

results[i][7] = reader.getKeepMoney();

results[i][8] = reader.getZj();

results[i][9] = reader.getZy();

results[i][10] = reader.getISBN();

results[i][11] = reader.getBztime();

}

return results;

}

public ReaderModiAndDelIFrame() {

super();

setIconifiable(true);

setClosable(true);

setTitle("读者信息修改与删除");

setBounds(50, 0, 600, 500);

final JPanel panel = new JPanel();

panel.setLayout(new BorderLayout());

panel.setPreferredSize(new Dimension(400, 80));

getContentPane().add(panel, BorderLayout.NORTH);

final JLabel logoLabel = new JLabel();

ImageIcon readerModiAndDelIcon = CreateIcon.add("readerModiAndDel.jpg");

logoLabel.setIcon(readerModiAndDelIcon);

logoLabel.setBackground(Color.CYAN);

logoLabel.setOpaque(true);

logoLabel.setPreferredSize(new Dimension(600, 80));

panel.add(logoLabel);

logoLabel.setText("读者信息修改logo（400\*80）");

final JPanel panel\_1 = new JPanel();

panel\_1.setLayout(new BorderLayout());

getContentPane().add(panel\_1);

final JScrollPane scrollPane = new JScrollPane();

scrollPane.setPreferredSize(new Dimension(0, 100));

panel\_1.add(scrollPane, BorderLayout.NORTH);

final DefaultTableModel model = new DefaultTableModel();

Object[][] results = getFileStates(Dao.selectReader());

model.setDataVector(results, columnNames);

table = new JTable();

table.setModel(model);

scrollPane.setViewportView(table);

table.setAutoResizeMode(JTable.AUTO\_RESIZE\_OFF);

table.addMouseListener(new TableListener());

final JPanel panel\_2 = new JPanel();

final GridLayout gridLayout = new GridLayout(0, 4);

gridLayout.setVgap(9);

panel\_2.setLayout(gridLayout);

panel\_2.setPreferredSize(new Dimension(0, 220));

panel\_1.add(panel\_2, BorderLayout.SOUTH);

final JLabel label\_1 = new JLabel();

label\_1.setText(" 姓 名：");

panel\_2.add(label\_1);

readername = new JTextField();

readername.setDocument(new MyDocument(10));

panel\_2.add(readername);

final JLabel label\_2 = new JLabel();

label\_2.setText(" 性 别：");

panel\_2.add(label\_2);

final JPanel panel\_3 = new JPanel();

final FlowLayout flowLayout\_1 = new FlowLayout();

flowLayout\_1.setVgap(0);

panel\_3.setLayout(flowLayout\_1);

panel\_2.add(panel\_3);

JRadioButton1 = new JRadioButton();

JRadioButton1.setSelected(true);

buttonGroup.add(JRadioButton1);

panel\_3.add(JRadioButton1);

JRadioButton1.setText("男");

JRadioButton2 = new JRadioButton();

buttonGroup.add(JRadioButton2);

panel\_3.add(JRadioButton2);

JRadioButton2.setText("女");

final JLabel label\_3 = new JLabel();

label\_3.setText(" 年 龄：");

panel\_2.add(label\_3);

age = new JTextField();

age.setDocument(new MyDocument(2));

age.addKeyListener(new NumberListener());

panel\_2.add(age);

final JLabel label\_5 = new JLabel();

label\_5.setText(" 职 业：");

panel\_2.add(label\_5);

zy = new JTextField();

zy.setDocument(new MyDocument(30));

panel\_2.add(zy);

final JLabel label = new JLabel();

label.setText(" 有效证件：");

panel\_2.add(label);

comboBox = new JComboBox();

comboBox.setModel(new DefaultComboBoxModel(array));

for (int i = 1; i < array.length; i++) {

comboBox.setSelectedIndex(i);

comboBox.setSelectedItem(array);

}

panel\_2.add(comboBox);

final JLabel label\_6 = new JLabel();

label\_6.setText(" 证件号码：");

panel\_2.add(label\_6);

zjnumber = new JTextField();

zjnumber.setDocument(new MyDocument(18));

zjnumber.addKeyListener(new NumberListener());

panel\_2.add(zjnumber);

final JLabel label\_7 = new JLabel();

label\_7.setText(" 办证日期：");

panel\_2.add(label\_7);

SimpleDateFormat myfmt = new SimpleDateFormat("yyyy-MM-dd");

bztime = new JFormattedTextField(myfmt.getDateInstance());

panel\_2.add(bztime);

final JLabel label\_9 = new JLabel();

label\_9.setText(" 最大借书量：");

panel\_2.add(label\_9);

maxnumber = new JTextField();

maxnumber.addKeyListener(new NumberListener());

panel\_2.add(maxnumber);

final JLabel label\_13 = new JLabel();

label\_13.setText(" 会员证有效日期：");

panel\_2.add(label\_13);

date = new JFormattedTextField(myfmt.getDateInstance());

panel\_2.add(date);

final JLabel label\_8 = new JLabel();

label\_8.setText(" 电 话：");

panel\_2.add(label\_8);

tel = new JFormattedTextField();

tel.addKeyListener(new TelListener());

tel.setDocument(new MyDocument(11));

panel\_2.add(tel);

final JLabel label\_14 = new JLabel();

label\_14.setText(" 押 金：");

panel\_2.add(label\_14);

keepmoney = new JTextField();

keepmoney.addKeyListener(new KeepmoneyListener());

panel\_2.add(keepmoney);

final JLabel label\_4 = new JLabel();

label\_4.setText(" 借阅图书号：");

panel\_2.add(label\_4);

ISBN = new JTextField();

ISBN.setEditable(false);

ISBN.setDocument(new MyDocument(13));

panel\_2.add(ISBN);

final JPanel panel\_4 = new JPanel();

panel\_4.setMaximumSize(new Dimension(0, 0));

final FlowLayout flowLayout = new FlowLayout();

flowLayout.setVgap(0);

flowLayout.setHgap(4);

panel\_4.setLayout(flowLayout);

panel\_2.add(panel\_4);

final JButton button = new JButton();

button.setHorizontalTextPosition(SwingConstants.CENTER);

panel\_4.add(button);

button.setText("修改");

button.addActionListener(new ModiButtonListener(model));

final JButton buttonDel = new JButton();

panel\_4.add(buttonDel);

buttonDel.setText("删除");

buttonDel.addActionListener(new DelButtonListener(model));

setVisible(true);

}

class TableListener extends MouseAdapter {

public void mouseClicked(final MouseEvent e) {

int selRow = table.getSelectedRow();

// id=table.getValueAt(selRow, 0).toString().trim();

readername.setText(table.getValueAt(selRow, 0).toString().trim());

if (table.getValueAt(selRow, 1).toString().trim().equals("男"))

JRadioButton1.setSelected(true);

else

JRadioButton2.setSelected(true);

age.setText(table.getValueAt(selRow, 2).toString().trim());

zjnumber.setText(table.getValueAt(selRow, 3).toString().trim());

date.setText(table.getValueAt(selRow, 4).toString().trim());

maxnumber.setText(table.getValueAt(selRow, 5).toString().trim());

tel.setText(table.getValueAt(selRow, 6).toString().trim());

keepmoney.setText(table.getValueAt(selRow, 7).toString().trim());

comboBox.setSelectedItem(table.getValueAt(selRow, 8).toString().trim());

zy.setText(table.getValueAt(selRow, 9).toString().trim());

ISBN.setText(table.getValueAt(selRow, 10).toString().trim());

bztime.setText(table.getValueAt(selRow, 11).toString().trim());

}

}

final class NumberListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789" + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

private final class DelButtonListener implements ActionListener {

private final DefaultTableModel model;

private DelButtonListener(DefaultTableModel model) {

this.model = model;

}

public void actionPerformed(final ActionEvent e) {

int i = Dao.DelReader(ISBN.getText().trim());

if (i == 1) {

JOptionPane.showMessageDialog(null, "删除成功");

Object[][] results = getFileStates(Dao.selectReader());

model.setDataVector(results, columnNames);

table.setModel(model);

}

}

}

class ModiButtonListener implements ActionListener {

private final DefaultTableModel model;

ModiButtonListener(DefaultTableModel model) {

this.model = model;

}

public void actionPerformed(final ActionEvent e) {

if (readername.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "读者姓名不能为空");

return;

}

if (age.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "读者年龄不能为空");

return;

}

if (zjnumber.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "证件号码不能为空");

return;

}

if (keepmoney.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "押金不能为空");

return;

}

if (zy.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "职业不能为空");

return;

}

if (ISBN.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "借阅图书号不能为空");

return;

}

if (ISBN.getText().length() != 13) {

JOptionPane.showMessageDialog(null, "借阅图书号必须为13位");

return;

}

if (bztime.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "办证时间不能为空");

return;

}

if (tel.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "电话号码不能为空");

return;

}

if (tel.getText().length() > 11 || tel.getText().length() < 0) {

JOptionPane.showMessageDialog(null, "电话号码位数小于11位");

return;

}

if (maxnumber.getText().length() == 0) {

JOptionPane.showMessageDialog(null, "最大借书量不能为空");

return;

}

if (maxnumber.getText().length() > 2 || tel.getText().length() < 0) {

JOptionPane.showMessageDialog(null, "最大借书量为两位数字");

return;

}

String sex = "1";

if (!JRadioButton1.isSelected()) {

sex = "2";

}

String zj = String.valueOf(comboBox.getSelectedIndex());

System.out.println(comboBox.getSelectedIndex());

int i = Dao.UpdateReader(id, readername.getText().trim(), sex, age.getText().trim(),

zjnumber.getText().trim(), Date.valueOf(date.getText().trim()), maxnumber.getText().trim(),

tel.getText().trim(), Double.valueOf(keepmoney.getText().trim()), zj, zy.getText().trim(),

Date.valueOf(bztime.getText().trim()), ISBN.getText().trim());

System.out.println(i);

if (i == 1) {

JOptionPane.showMessageDialog(null, "修改成功");

Object[][] results = getFileStates(Dao.selectReader());

model.setDataVector(results, columnNames);

table.setModel(model);

}

}

}

class TelListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789-" + (char) 8;

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

}

}

class KeepmoneyListener extends KeyAdapter {

public void keyTyped(KeyEvent e) {

String numStr = "0123456789" + (char) 8;// 只允许输入数字与退格键

if (numStr.indexOf(e.getKeyChar()) < 0) {

e.consume();

}

if (keepmoney.getText().length() > 2 || keepmoney.getText().length() < 0) {

e.consume();

}

}

}

}

第四章 系统测试

4.1 测试项目

页面测试：逐个测试每个页面是否可以正确显示，显示页面是否美观，易懂，测试各菜单能否正常运行。

经测试，所有界面均可以正确显示，显示页面比较美观，易懂，各菜单均能正常运行。

功能测试：对每个菜单下的功能逐一测试，尽最大可能发现潜在的错误。

经测试，每个菜单下的功能均能实现，暂未发现其潜在的错误。

需求测试：根据需求分析的内容，测试系统是否和当初的设计一样。

经测试，所设计的图书馆管理系统基本满足当初的设计需求。

4.2 测试用例

根据本系统的测试项目需求，设计了部分测试用例，见表4.1。

表4.1 测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试模块 | 输入 | 输出 | 预期结果 | 与预期结果 |
| 1 | 登陆模块 | 用户名：1  密码：6666 | 进入主系统界面 | 进入主系统界面 | 相同 |
| 2 | 图书信息添加 | ISBN：9787513506915  书名：翻译美学理论  出版社：外语教学与研究出版社  出版日期：2001-03-01  类别：中国文学  作者：刘宓庆  译者：1143  单价：49.9 | “添加成功！” | “添加成功！” | 相同 |
| 3 | 读者信息添加 | 姓名：夏季  性别：女  年龄：41  职业：老师  证件类别：身份证  证件号码：352188197809091234  最大借书量：8  会员证有效期：2017-09-27  电话：15324563456  押金：15.0  办证日期：2010-10-21  借阅图书号：9787301204580 | “添加成功！” | “添加成功！” | 相同 |
| 4 | 图书类别添加 | 图书类别名称：中国文学 | “添加成功！” | “添加成功！” | 相同 |
| 5 | 修改密码 | 用户名：4  旧密码：4444  新密码：6666  确认新密码：6666 | “修改成功！” | “修改成功！” | 相同 |
| 6 | 管理员信息添加 | 姓名：李靖  性别：女  年龄：18  注册日期：2012-11-11  联系电话：44444444444  密码：6666 | “添加成功！” | “添加成功！” | 相同 |