# 1 数据库设计

## 1.1 需求分析

## 1.1.1系统简要分析

本系统的用户为图书馆工作人员，系统用户分为管理员和普通用户两种。管理员为系统的高级用户，普通用户为图书馆工作人员。管理员账号和密码预先写入数据库中，账号是每位同学的学号，密码是学号后三位。

系统为不同用户提供不同的功能：

## 1.1.2基本需求功能点分析

系统为不同用户提供不同的功能：

管理员功能：

登陆：输入管理员帐号和密码，验证通过后可登陆系统。

用户管理：

添加系统普通用户

为普通用户重置登陆密码。

图书类别管理：

添加图书类别，如：政治，经济，军事，医药等。

修改图书类别名称。

类别查询。

图书管理：

添加一本图书。图书信息包括：书名，第一作者，第一出版社，出版年份，状态（正常，报废），类别等。

修改图书信息。

图书查询。

读者类别管理：

添加读者类别，类别信息包括：类别名称，最长借阅天数，最大借阅本数等。

修改读者类别名称。

读者类别查询。

读者管理：

添加读者。读者信息包括：姓名，读者类别，身份证号，联系电话等等。

修改读者信息。

读者查询。

普通用户功能：

(1)系统登陆和密码修改

(2)图书馆工作人员凭用户号和密码登陆系统。

(3)图书借阅

为读者借书。如果该读者的借书量未达到最大借阅本书，且无超期欠费情况，则可以为其登记借书信息；如有欠费，则需先缴费还书。

为读者还书。如果该读者所还图书未超期欠费，则可以为其还书；如欠费，则先计算欠费金额，付费后再还书。

## 1.1.3 系统运行条件分析

此软件系统需要至少一台计算机为服务器

Windows操作系统

SQL Server 2008 或更高版本数据库服务器

## 1.1.4 数据字典

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段序号 | 字段名 | 主键 | 类型 | 占用字节 | 长度 | 允许空 |
| t\_borrowing | 1 | readerId | √ | int | 4 | 10 |  |
|  | 2 | bookId | √ | int | 4 | 10 |  |
|  | 3 | borrowingTime | √ | date | 3 | 10 |  |
| t\_return | 1 | readerId | √ | int | 4 | 10 |  |
|  | 2 | bookId | √ | int | 4 | 10 |  |
|  | 3 | returnTime | √ | date | 3 | 10 |  |
| t\_reader\_bookRelation | 1 | readerId | √ | int | 4 | 10 |  |
|  | 2 | bookId | √ | int | 4 | 10 |  |
| t\_user | 1 | id | √ | int | 4 | 10 |  |
|  | 2 | password |  | varchar | 50 | 50 |  |
|  | 3 | username |  | varchar | 50 | 50 |  |
|  | 4 | usertype |  | varchar | 20 | 20 |  |
| t\_bookType | 1 | id | √ | int | 4 | 10 |  |
|  | 2 | bookTypeName |  | varchar | 10 | 10 |  |
|  | 3 | bookTypeDesc |  | varchar | 1000 | 1000 | √ |
| t\_book | 1 | id | √ | int | 4 | 10 |  |
|  | 2 | bookName |  | varchar | 20 | 20 |  |
|  | 3 | author |  | varchar | 20 | 20 |  |
|  | 4 | bookTypeid |  | int | 4 | 10 |  |
|  | 5 | publisher |  | varchar | 20 | 20 |  |
|  | 6 | publishtime |  | varchar | 10 | 10 |  |
|  | 7 | state |  | varchar | 10 | 10 |  |
|  | 8 | bookDesc |  | varchar | 100 | 100 | √ |
| t\_readerType | 1 | id | √ | int | 4 | 10 |  |
|  | 2 | readerTypeName |  | varchar | 20 | 20 |  |
|  | 3 | theLongestBorrowingDay |  | int | 4 | 10 |  |
|  | 4 | maximumBorrowingNumber |  | int | 4 | 10 |  |
| t\_reader | 1 | id | √ | int | 4 | 10 |  |
|  | 2 | IDNumber |  | varchar | 20 | 20 |  |
|  | 3 | name |  | varchar | 20 | 20 |  |
|  | 4 | readerTypeid |  | int | 4 | 10 |  |
|  | 5 | tel |  | varchar | 11 | 11 | √ |

表1-1 数据字典

Tab.1-1 Data dictionary

## 

## 1.2 概念结构设计

系统E-R图如图1-1：



图1-1 图书管理系统E-R图

Fig.1-1 Book management system E-R diagram

## 1.3 逻辑结构设计

此部分需要列出由E-R图转化得到的数据表。

1.用户表，具体数据如表1-2所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 约束 | 说明 |
| id | int | 4 | 主码 | 用户id |
| password | varchar | 50 | 非空 | 用户登录密码 |
| username | varchar | 50 | 非空 | 用户名 |
| usertype | varchar | 20 | 非空，管理员或普通用户 | 用户类型 |

表1-2 用户表

Tab.1-2 User table

2.图书类别表，具体数据如表1-3所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 约束 | 说明 |
| id | int | 4 | 主码 | 图书类别id |
| bookTypeName | varchar | 10 | 非空 | 图书类别名 |
| bookTypeDesc | varchar | 1000 | 无 | 图书类别描述 |

表1-3图书类别表

Tab.1-3 BookType Table

3.图书表，具体数据如表1-4所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 约束 | 说明 |
| id | int | 4 | 主码 | 图书id |
| bookName | varchar | 20 | 非空 | 图书名 |
| author | varchar | 20 | 非空 | 作者 |
| bookTypeid | int | 4 | 非空，外键 | 图书类别id |
| publisher | varchar | 20 | 非空 | 出版社 |
| publishtime | varchar | 10 | 非空 | 出版时间 |
| state | varchar | 10 | 正常或报废 | 图书状态 |
| bookDesc | varchar | 100 | 无 | 图书描述 |

表1-4图书表

Tab.1-4 Book Table

4.读者类别表，具体数据如表1-5所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 约束 | 说明 |
| id | int | 4 | 主码 | 图书id |
| readerTypeName | varchar | 20 | 非空 | 读者类型名 |
| theLongestBorrowingDays | int | 4 | 非空 | 最长借阅时间 |
| maximumBorrowingNumber | int | 4 | 非空 | 最大借阅数量 |

表1-5读者类别表

Tab.1-5 ReaderType Table

5.读者表，具体数据如表1-6所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 约束 | 说明 |
| id | int | 4 | 主码 | 读者id |
| IDNumber | varchar | 20 | 非空 | 身份证号 |
| name | varchar | 20 | 非空 | 姓名 |
| readerTypeid | int | 4 | 非空，外键 | 读者类别号 |
| tel | varchar | 11 | 无 | 联系方式 |

表1-6读者表

Tab.1-6 Reader Table

6.借书表，具体数据如表1-7所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 约束 | 说明 |
| readerId | int | 4 | 主码 | 读者id |
| bookId | int | 4 | 主码 | 图书id |
| borrowingTime | date | 8 | 主码 | 借阅时间 |

表1-7借书表

Tab.1-7 Borrowing Table

7还书表，具体数据如表1-8所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 约束 | 说明 |
| readerId | int | 4 | 主码 | 读者id |
| bookId | int | 4 | 主码 | 图书id |
| returnTime | date | 8 | 主码 | 归还时间 |

表1-8还书表

Tab.1-8 Return Table

8.读者-图书关系表，具体数据如表1-9所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 约束 | 说明 |
| readerId | int | 4 | 主码 | 读者id |
| bookId | int | 4 | 主码 | 图书id |

表1-9读者-图书关系表

Tab.1-9Reader\_bookRelation Table

## 1.4 物理结构设计

数据库管理系统：SQL Server 2014

## 1.5 数据库实施

## 1.5.1 数据库表和视图清单

如图1-2所示：

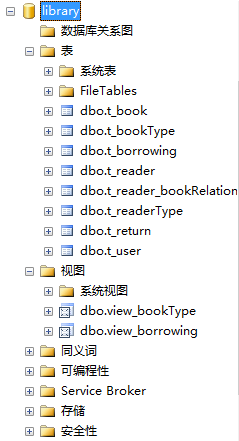


图1-2数据库表和视图清单

Fig.1-2 Table and view list

## 1.5.2 数据库基本表

1. 用户表

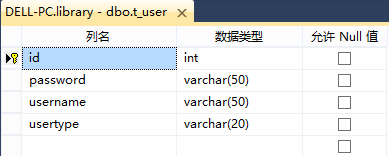


图1-3 用户表

Fig.1-3 User Table

1. 图书类别表

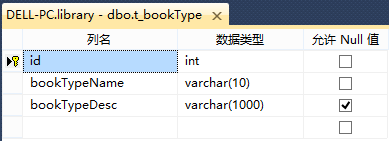


图1-4 图书类别表

Fig.1-4 BookType Table

1. 图书表

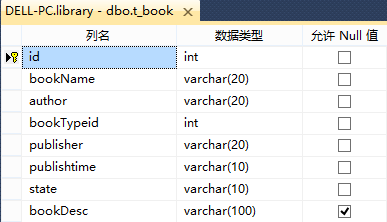


图1-5 图书表

Fig.1-5 Book Table

1. 读者类别表

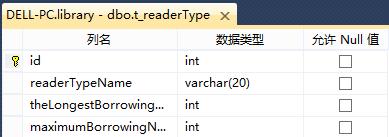


图1-6 读者类别表

Fig.1-6 ReaderType Table

1. 读者表

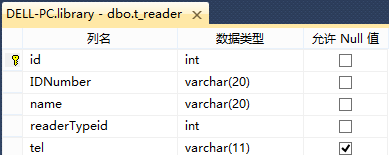


图1-7 读者表

Fig.1-7 Reader Table

1. 借书表

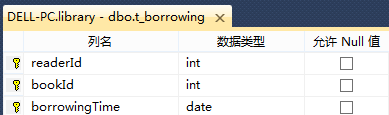


图1-8 借书表

Fig.1-8 Borrowing Table

1. 还书表

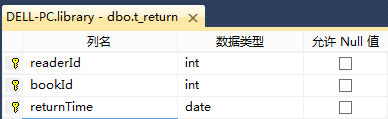


图1-9 还书表

Fig.1-9 Return Table

1. 读者-图书关系表

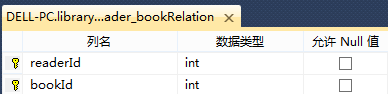


图1-10 读者-图书关系表

Fig.1-10 Reader\_bookRelation Table

## 1.5.3 视图

1. view\_bookType

创建视图代码：

create view view\_bookType

as

select\* from t\_bookType

作用：对图书类别全部属性的查询语句，通过java代码实现可显示在图书类别维护界面的表格中。

1. view\_borrowing

创建视图代码：

create view view\_borrowing

as

select readerId,IDNumber,name,bookId,borrowingTime

from t\_reader,t\_borrowing

where t\_reader.id=t\_borrowing.readerId;

作用：对借书表和读者表部分属性的查询语句，通过java代码实现可显示在图书借阅记录界面的表格中。

1. view\_userlist

创建试图代码：

create view view\_userlist

as

select \* from t\_user where usertype='普通用户'

作用：查询用户类别为普通用户的用户的全部属性，通过java代码显示在重置用户密码界面的表格中。

## 1.5.4 存储过程

1. existBookByBookTypeId存储过程

创建存储过程代码：

create procedure existBookByBookTypeId

(@ID int)

as

(select\*

from t\_book

where bookTypeId=@ID)

作用：判断指定图书类别下是否有图书

2.existReaderByReaderTypeId存储过程

create procedure existReaderByReaderTypeId

(@ID int)

as

(select\*

from t\_reader

where readerTypeId=@ID)

作用：判断指定读者类别下是否有读者

## 1.5.5 触发器

删除读者触发器

create trigger triger\_deletereader

on t\_reader

instead of delete

as

begin

EXEC sp\_msforeachtable 'ALTER TABLE ? NOCHECK CONSTRAINT ALL'--禁用约束

delete from t\_borrowing where readerId=(select id from deleted)

delete from t\_return where readerId=(select id from deleted)

delete from t\_reader where id=(select id from deleted)

EXEC sp\_msforeachtable 'ALTER TABLE ? CHECK CONSTRAINT ALL' --启用约束

end

作用：删除读者时有外键约束，触发器取消外键约束，删除借书还书记录中该读者的记录，再删除该读者，再启用约束。

# 

# 2系统设计

## 2.1 系统功能模块设计

此部分主要描述各个模块具体实现的各项功能点以及模块之间的关系，模块结构划分如图2-1所示：



图2-1 系统模块结构图图

Fig.2-1 System module structure diagram

### 

### 2.1.1 管理员用户模块

该模块具体功能是管理员对图书系统进行管理，主要包括图书类别管理模块，图书管理模块，读者类别管理模块，读者管理模块，用户管理模块。

1. 图书类别管理模块：此模块下可进行图书类别的添加和维护，维护包括查询、删除和修改，可以对图书类别名和图书类别描述等属性进行操作。
2. 图书管理模块：此模块下可进行图书的添加和维护操作，维护包括查询删除和修改，可对图书名称、图书作者、图书类别、出版社、出版时间、图书状态、图书描述等属性进行操作。
3. 读者类别管理模块：此模块下可进行读者类别的添加和维护，维护包括查询、删除和修改，可对读者类别名称、最长借阅天数和最大借阅本数等属性进行操作。
4. 读者管理模块：此模块下可进行读者的添加和维护，维护包括查询、修改和删除，可对读者身份证号、姓名、联系方式、读者类别等属性进行操作。
5. 用户管理模块：此模块下可进行对普通用户的添加和修改普通用户密码操作。

### 2.1.2普通用户具体功能模块

该模块具体功能是普通用户登录系统后进行借书和还书操作，主要包括图书借阅和图书归还模块。

1. 图书借阅模块：此模块下可进行图书借阅操作和对借阅记录的查询操作。图书借阅可添加图书借阅记录。
2. 图书归还模块：此模块下可进行图书归还操作和对归还记录的查询操作。图书归还可添加图书归还记录。

## 

## 2.2 系统功能设计

本部分按系统主要功能绘制流程图，说明系统功能将如何现。如图2-2所示：



图2-2 系统流程图

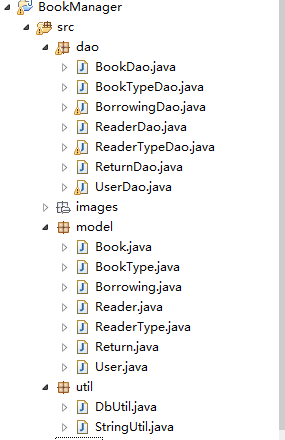
Fig.2-2 System module structure diagram

# 3系统实现

## 3.1系统项目清单

系统项目具体所含如图3-1所示：

model包包含各种实体类，dao包包含各种数据库操作的类，util包包含数据库工具类和字符串工具类，model包包含各种界面类。



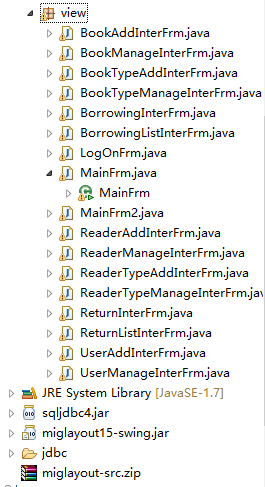


图3-1 系统项目清单

Fig.3-1 System project list

数据库工具类DbUtil代码：

public class DbUtil {

private String dbUrl="jdbc:sqlserver://127.0.0.1:1433;DatabaseName=library";//数据库连接地址

private String dbUserName="root";

private String dbPassword="123456";

private String jdbcName="com.microsoft.sqlserver.jdbc.SQLServerDriver";

/\*\*

\* 获取数据库连接

\* @return

\* @throws Exception

\*/

public Connection getCon()throws Exception{

Class.forName(jdbcName);

Connectioncon=DriverManager.getConnection(dbUrl,dbUserName, dbPassword);

return con;

}

/\*\*

\* 关闭数据库连接

\* @param con

\* @throws Exception

\*/

public void closeCon(Connection con)throws Exception

{

if(con!=null)

{

con.close();

}

}

public static void main(String[] args){

DbUtil dbUtil=new DbUtil();

try {

dbUtil.getCon();

System.out.println("数据库连接成功！");

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

System.out.println("数据库连接失败");

}

}

}

## 3.2登录界面

登录界面由LogOnFrm.java实现，输入用户名和密码，选择用户类型为管理员或普通用户，验证成功即可进入主界面。用户名或密码错误或未输入则弹出窗体提示。如图3-2所示。

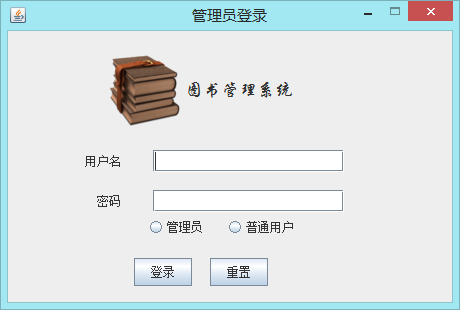


图3-2 登录界面

Fig.3-2 Login Interface

## 3.3管理员主界面

管理员主界面由MainFrm.java实现，选中菜单栏中选项选择要显示的功能窗体。如图3-3所示：

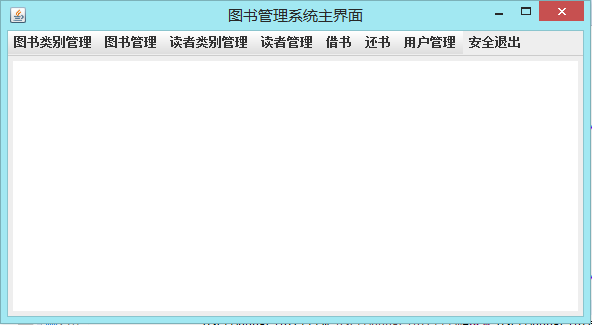


图3-3 管理员主界面

Fig.3-3 Main Interface

### 3.3.1图书类别维护界面

图书类别维护界面如图3-4可对图书进行查询、修改、删除操作。删除操作调用存储过程existBookByBookTypeId判断该图书类别下是否有图书，有图书的错误提示见图3-5：

图书类别删除事件程序：

private void bookTypeDeleteAcitonEvent(ActionEvent evt) {

String id=idTxt.getText();

if(StringUtil.isEmpty(id))

{

JOptionPane.showMessageDialog(null,"请选择要删除的记录");

return;

}

int n=JOptionPane.showConfirmDialog(null, "确定要删除该记录吗？");

if(n==0){

Connection con=null;

try{

con=dbUtil.getCon();

boolean flag=bookDao.existBookByBookTypeId(con, id);

if(flag){

JOptionPane.showMessageDialog(null,"当前图书类别下有图书，不能删除此类别");

return;

}

int deleteNum=bookTypeDao.delete(con, id);

if(deleteNum==1){

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

this.resetValue();

this.filltable(new BookType());

}else{

JOptionPane.showMessageDialog(null,"删除失败");

}

}catch(Exception e){

e.printStackTrace();

JOptionPane.showMessageDialog(null,"删除失败");

}finally{

try {

dbUtil.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

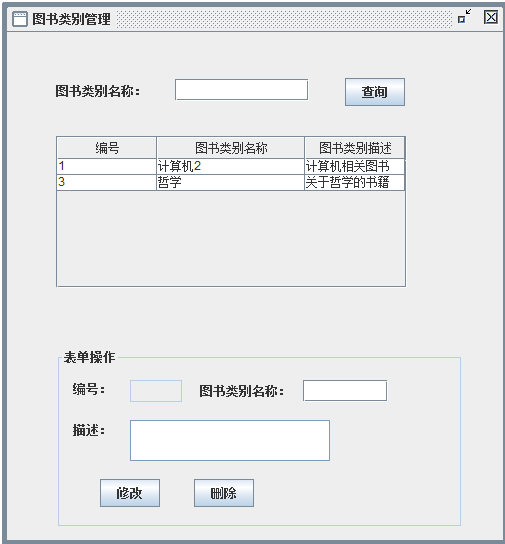


图3-4图书类别维护界面

Fig3-4 BookType Maintain Interface

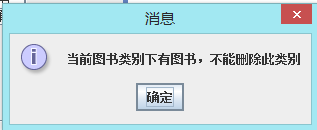


图3-5 图书类别下有图书的错误提示

Fig3-5.Error exist book

### 3.3.2图书添加

图书的添加是管理员向数据库中加入一条图书信息，若图书名称、图书作者等非空数据为空，则弹出提示图书不能为空等。图书类别为一个下拉框，其中显示图书类别表中的图书类别名称。图书状态只有正常和报废两种，用单选按钮实现。图书添加界面如图3-6所示。

图书添加数据库操作函数：

public int add(Connection con,Book book)throws Exception{

String sql="insert into t\_book values(?,?,?,?,?,?,?)";

PreparedStatement pstmt= con.prepareStatement(sql);

pstmt.setString(1, book.getBookName());

pstmt.setString(2, book.getAuthor());

pstmt.setInt(3, book.getBookTypeid());

pstmt.setString(4, book.getPublisher());

pstmt.setString(5, book.getPublishtime());

pstmt.setString(6, book.getState());

pstmt.setString(7, book.getBookDesc());

return pstmt.executeUpdate();

}

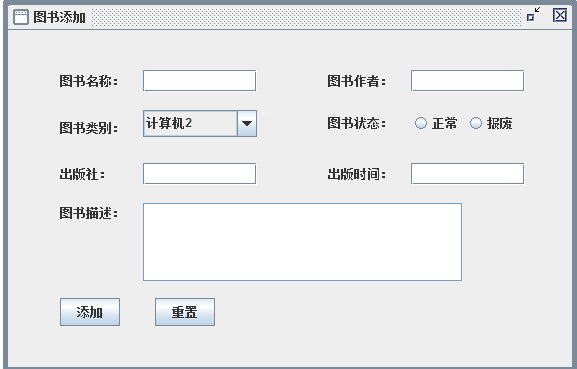


图3-6图书添加界面

Fig.3-6 Book Add Interface

### 3.3.3图书维护

图书维护界面可实现图书的查询、修改删除。没有操作时中间表格bookTable显示所有图书。进行查询操作时表格中显示查询到的结果。添加了表格行点击事件，点击指定行时，下面的表单操作中的文本框被选中内容填充，修改文本框的内容后点击修改按钮，对数据库执行update操作，修改了数据库中数据，然后界面中bookTable表刷新，显示更新后的数据。点击删除按钮时执行sql的delete操作。图书维护界面如图3-7所示。

图书信息表中返回结果集函数：

public ResultSet list(Connection con,Book book)throws Exception{

StringBuffer sb=new StringBuffer("select \* from t\_book b,t\_booktype bt where b.bookTypeid=bt.id");

if(StringUtil.isNotEmpty(book.getBookName())){

sb.append(" and b.bookName like '%"+book.getBookName()+"%'");

}

if(StringUtil.isNotEmpty(book.getAuthor())){

sb.append(" and b.author like '%"+book.getAuthor()+"%'");

}

if(book.getBookTypeid()!=null && book.getBookTypeid()!=-1){

sb.append(" and b.bookTypeid="+"book.getBookTypeid()");

}

PreparedStatement pstmt= con.prepareStatement(sb.toString());

return pstmt.executeQuery();

}



图3-7 图书维护界面

Fig.3-7 Book Maintain Interface

### 3.3.4添加普通用户

在管理员主界面选择用户管理-添加普通用户弹出菜单如图3-8，输入要添加的用户名和密码，点击添加按钮，提示用户添加成功。如用户名和密码为空则弹出错误提示。重置按钮可清空两个文本框。

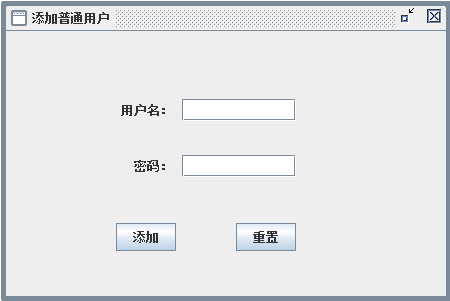


图3-8添加普通用户界面

Fig.3-8 Add User Interface

### 3.3.5重置用户密码菜单

在管理员主界面选择用户管理-重置用户密码菜单如图3-9，可进行查询和修改密码操作。在查询框中输入要查询的用户名，中间的userTable中将显示查询出的信息，运用视图view\_userlist。点击要修改的用户所在行，该用户密码将出现在密码修改框的密码文本框中，修改内容，点击修改按钮，弹出用户密码修改成功则完成用户密码修改。选中表中用户点击删除按钮，弹出删除成功提示框，则完成了删除该普通用户操作。

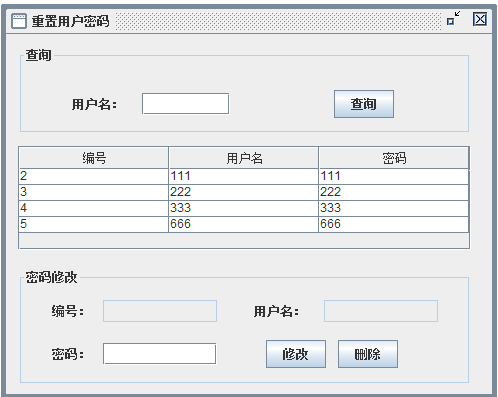


图3-9 重置用户密码界面

Fig.3-9 Reset Password Interface

### 3.3.6 读者维护界面

在管理员主界面选择读者管理-读者维护显示读者维护界面如图3-10。可对读者进行查询、修改、删除操作。查询结果显示在表格中。选择表格中数据可进行修改、删除操作。修改、删除完成后提示修改成功或删除成功。删除操作运用触发器triger\_deletereader同时删除被删除读者的借书记录和还书记录。

读者删除事件代码：

private void readerDeleteAcitonPerformed(ActionEvent evt) {

String id=idTxt.getText();

if(StringUtil.isEmpty(id))

{

JOptionPane.showMessageDialog(null,"请选择要删除的记录");

return;

}

int n=JOptionPane.showConfirmDialog(null, "确定要删除该记录吗？");

if(n==0){

Connection con=null;

try{

con=dbUtil.getCon();

if(readerDao.existBookNotReturn(con, id)){

JOptionPane.showMessageDialog(null,"该读者有书未还，不能删除！");

return;

}else{

int deleteNum=readerDao.delete(con, id);

if(deleteNum==1){

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

this.resetValue();

this.fillTable(new Reader());

}else{

JOptionPane.showMessageDialog(null,"删除失败");

}}

}catch(Exception e){

e.printStackTrace();

JOptionPane.showMessageDialog(null,"删除失败");

}finally{

try {

dbUtil.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

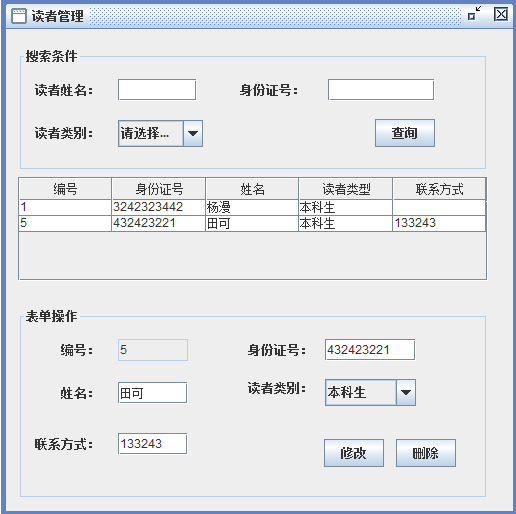
****

图3-10读者维护界面

Fig.3-10 Reader Maintain Interface

### 3.3.7 读者类别维护界面

在管理员主界面选择读者类别管理-读者类别维护显示读者类别维护界面如图3-11。读者类别管理界面可实现对读者类别的查询、修改、删除操作，其中删除操作调用存储过程existReaderByReaderTypeId判断该读者类别下是否有读者。有读者时的错误提示如图3-12

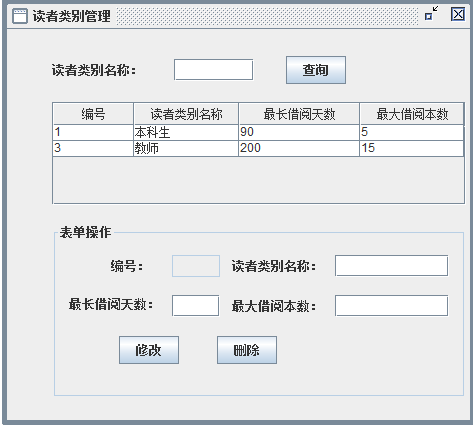


图3-11 读者类别管理界面

Fig.3-11 ReaderType Manage Interface

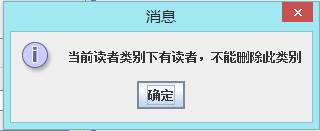


图3-12 读者类别下有读者错误提示

Fig.3-12 Error exist Reader

## 3.4普通用户主界面

普通用户主界面由MainFrm2.java实现，选中菜单栏中选项选择要显示的功能窗体。如图3-13所示

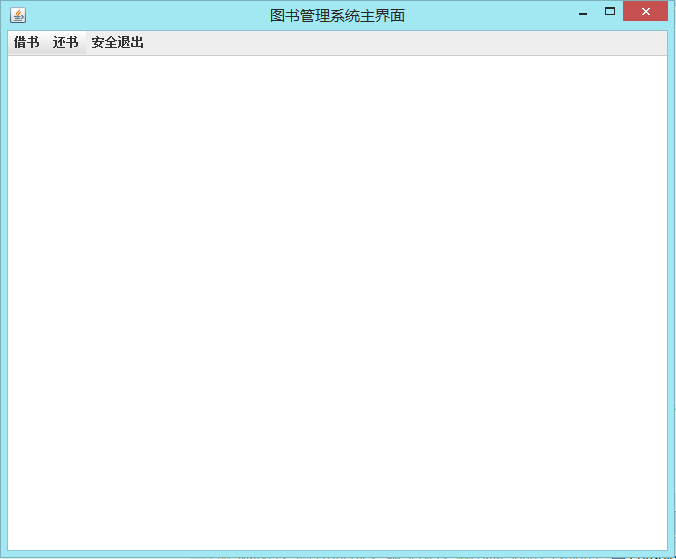


图3-13普通用户主界面

Fig.3-13 Main2 Interface

### 3.4.1图书借阅界面

在普通用户主界面点击借书-图书借阅显示图书借阅界面如图3-14。输入读者编号和图书编号点击确定，若读者编号和图书编号在t\_reader和t\_book表中不存在则弹出错误提示框。点击重置两文本框重新为空。借书本数超限提示如图3-15，借书时间超限提示如图3-16

借书事件代码：

private void borrowingAddActionPerformed(ActionEvent evt) {

String readerId=this.readerIdTxt.getText();

String bookId=this.bookIdTxt.getText();

if(StringUtil.isEmpty(readerId)){

JOptionPane.showMessageDialog(null,"读者编号不能为空！");

return;

}

if(StringUtil.isEmpty(bookId)){

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

return;

}

Connection con=null;

try{

con=dbUtil.getCon();

int flag=1;

java.sql.Date date=new java.sql.Date(new java.util.Date().getTime());

if(!borrowingDao.existInReader(con, readerId)){

JOptionPane.showMessageDialog(null,"不存在此读者！");

flag=0;

}

if(!borrowingDao.existInBook(con, bookId)){

JOptionPane.showMessageDialog(null,"不存在此图书！");

flag=0;

}

if(flag==1&&borrowingDao.numberOfbook(con, readerId)>=borrowingDao.maxBorrowingNumber(con, readerId))

{

JOptionPane.showMessageDialog(null,"该读者借阅图书数量已达到最大限制");

flag=0;

}

if(flag==1&&((date.getTime()-borrowingDao.farthestDay(con, readerId).getTime())/86400000)>borrowingDao.theLongestBorrowingDays(con, readerId)){

JOptionPane.showMessageDialog(null,"该读者有图书未还，超过最长借阅天数");

flag=0;

}

if(flag==1){

int addNum=borrowingDao.add(con, readerId, bookId, date);

if(addNum==1){

if(borrowingDao.addInR(con, readerId, bookId)==1){

JOptionPane.showMessageDialog(null,"借阅成功！");

resetValue();

return;

}

}else{

JOptionPane.showMessageDialog(null,"借阅失败！");

}}

}catch(Exception e){

e.printStackTrace();

JOptionPane.showMessageDialog(null,"借阅失败！");

}finally{

try {

dbUtil.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

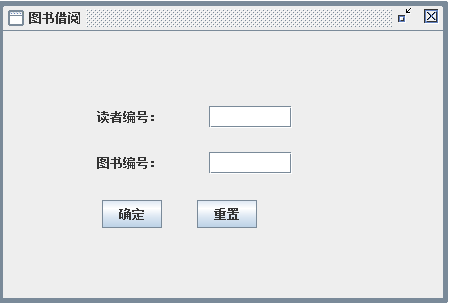


图3-14 图书借阅界面

Fig.3-14 Borrowing Interface

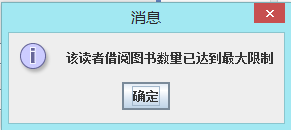


图3-15 图书数量最大限制错误提示

Fig.3-15 Error MaxNumber

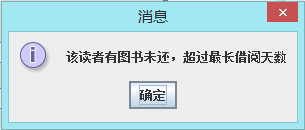


图3-16 最长借阅天数错误提示

Fig3-16 Error Longest Days

### 3.4.2图书借阅记录查询

在普通用户主界面选择借书-借阅记录查询显示界面如图3-17。在查询框中输入读者身份证号和图书编号，点击查询按钮，可进行查询操作。中间表格运用视图view\_borrowing显示查询出来的记录，查询之前显示所有的借书记录。



图3-17 借书记录查询界面

Fig.3-17 Borrowing List Interface

# 4总结

图书管理系统运用java语言，连接数据库，主要用eclipse的windowsbuilder插件完成图形化界面，数据存储在SQL server数据库中，通过SQL语句完成对数据库的操作。

运用了视图、存储过程、触发器来完善系统。

系统完成了对图书、读者、用户等对象的增删改查操作。实现了大部分图书管理系统要求的功能。

不足：

系统虽然完成了图书借阅超过期限的提示，但没有写有关的缴费界面和操作。