

# 《数据库系统原理》实验报告

(实验名称： 图书管理系统实验 )

专    业	<u>信息资源管理</u>
班    级	<u>信管 1601</u>
学    号	<u>3160104494</u>
学生姓名	<u>梁鑫宇</u>
指导老师	<u>林怀忠、田若言</u>

浙江大学

2018 年 6 月 27 日

# 目录

1	实验目的 .....	2
2	实验内容: .....	2
2.1	总体设计 .....	2
2.1.1	系统架构的描述 .....	2
①	系统架构 .....	2
②	基本功能模块 .....	3
2.1.2	数据库表结构设计 .....	3
①	书籍 .....	3
②	借书证 .....	4
③	管理员 .....	4
④	借书记录 .....	4
2.2	详细设计 (流程图、界面及实现细节) .....	5
2.2.1	主菜单 .....	5
2.2.2	读者登录 .....	5
2.2.3	管理员登录 .....	6
2.2.4	图书查询 .....	7
2.2.5	图书入库 .....	9
2.2.6	借书 .....	10
2.2.7	还书 .....	12
2.2.8	添加借书证 .....	13
2.2.9	删除借书证 .....	13
3	实验环境 .....	14
4	程序源码 .....	14
5	实验总结 .....	15

## 1 实验目的

设计并实现一个精简的图书管理系统，要求具有图书入库、查询、借书、还书、借书证管理等功能。

通过该图书馆系统的设计与实现，提高系统编程能力，加深对数据库系统原理及应用的理解。

## 2 实验内容：

### 2.1 总体设计

#### 2.1.1 系统架构的描述

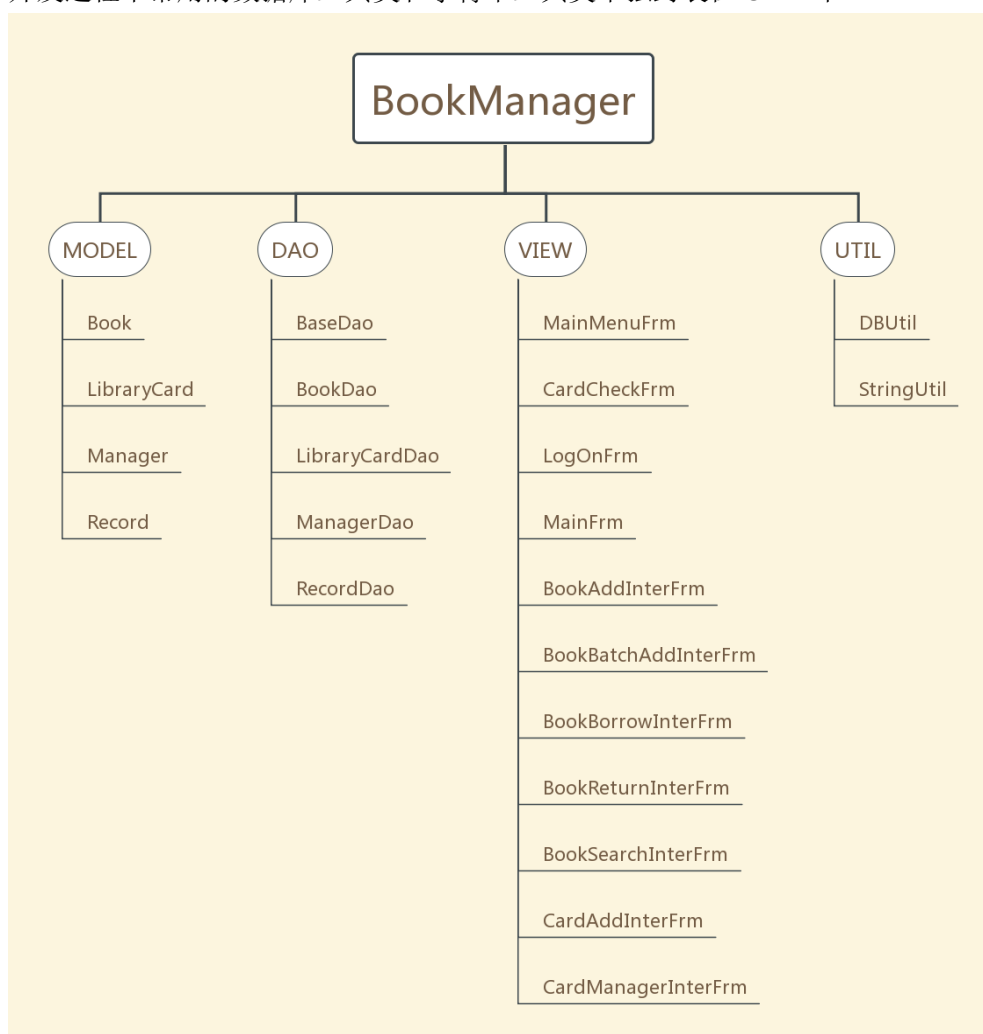
##### ① 系统架构

采用 MVC 模式设计，对应数据库封装了四类实体 Book, LibraryCard, Manager, Record。

设计数据库访问类 DAO 体系使用 JDBC 提供操作数据库的方法，其中文件读取使用了 jxl 包。DAO 体系设计了基类 BaseDao，其余 Dao 类均继承基类而来，形成较为清晰的类体系。

最后利用 Java Swing 完成图形界面的设计。

开发过程中常用的数据库工具类和字符串工具类单独封装在 UTIL 中

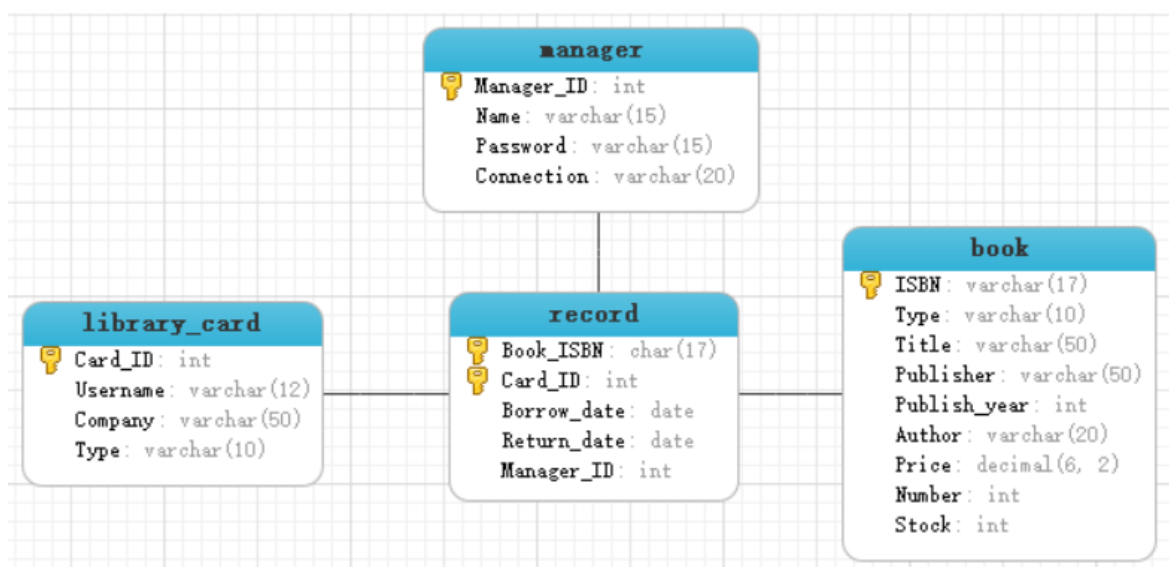


## ② 基本功能模块

模块名称	功能描述
管理员登陆	输入管理员 ID, 密码; 登入系统 或 返回 ID/密码 错误.
读者登录	输入借书证 ID, 登入图书查询界面 或 返回 借书证卡号错误
图书入库	1.单本入库 2.批量入库 (方便最后测试)
图书查询	1.精确查找 输入书号, 显示对应书籍所有信息 2.模糊查找 输入书籍相关信息, 如书名, 作者, 出版年份等, 显示符合条件的所有书籍的相关信息
借书	1.输入借书证卡号 显示该借书证所有已借书籍 2.输入书号 如果该书还有库存, 则借书成功, 同时库存数减一。 否则输出该书无库存, 且输出最近归还的时间。
还书	1.输入借书证卡号 显示该借书证所有已借书籍 (返回, 格式同查询模块) 2.输入书号 如果该书在已借书籍列表内, 则还书成功, 同时库存加一。 否则输出出错信息.
借书证管理	增加或删除一个借书证.

除图书查询功能外, 其余功能模块都由图书管理员操作。

## 2.1.2 数据库表结构设计



## ① 书籍

book		数据类型	约束
书号	ISBN	varchar(17)	primary key

类别	Type	varchar(10)	
书名	Title	varchar(50)	
出版社	Publisher	varchar(50)	
年份	Publish_year	int	
作者	Author	varchar(20)	
价格	Price	decimal(6,2)	
总藏书量	Number	int	check(Number>=0)
库存	Stock	int	check(Stock>=0)

## ② 借书证

library_card		数据类型	约束
卡号	Card_ID	int	primary key
姓名	Username	varchar(12)	
单位	Company	varchar(50)	
类别	Type	varchar(10)	check (Type in ( 'student', 'teacher' ) )

## ③ 管理员

manager		数据类型	约束
管理员 ID	Manager_ID	int	primary key
密码	Password	varchar(15)	not null
姓名	Name	varchar(12)	not null
联系方式	Connection	varchar(20)	

## ④ 借书记录

record		数据类型	约束
书号	Book_ID	int	REFERENCES book(ISBN)
卡号	Card_ID	int	REFERENCES library_card(Card_ID)
借期	Borrow_date	date	
还期	Return_date	date	
经手人	Manager_ID	int	REFERENCES manager(Manager_ID)

外键约束：

栏位	索引	外键	触发器	选项	注释	SQL 预览
名	栏位	参考数据库	参考表	参考栏位	删除时	更新时
record_ibfk_1	Book_ISBN	db_book	book	isbn	RESTRICT	CASCADE
record_ibfk_2	Card_ID	db_book	library_card	card_id	RESTRICT	CASCADE
record_ibfk_3	Manager_ID	db_book	manager	Manager_ID	RESTRICT	CASCADE

触发器：

栏位	索引	外键	触发器	选项	注释	SQL 预览
名	触发	插入	更新	删除		
AutoSetDate	Before	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
AutoBorrowStock	After	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
AutoReturnStock	After	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

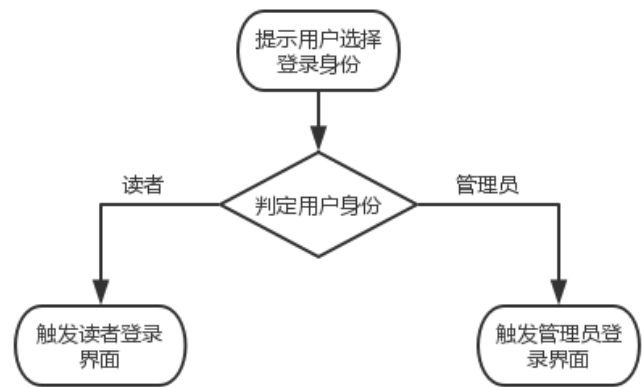
触发器名	定义
AutoSetDate	set new.Borrow_date = curdate(); set new.Return_date = date_add(new.Borrow_date,INTERVAL 1 month);

AutoBorrowStock	update book set Stock=Stock-1 where ISBN=new.Book_ISBN
AutoReturnStock	update book set Stock=Stock+1 where ISBN=old.Book_ISBN

## 2.2 详细设计（流程图、界面及实现细节）

### 2.2.1 主菜单

流程图：

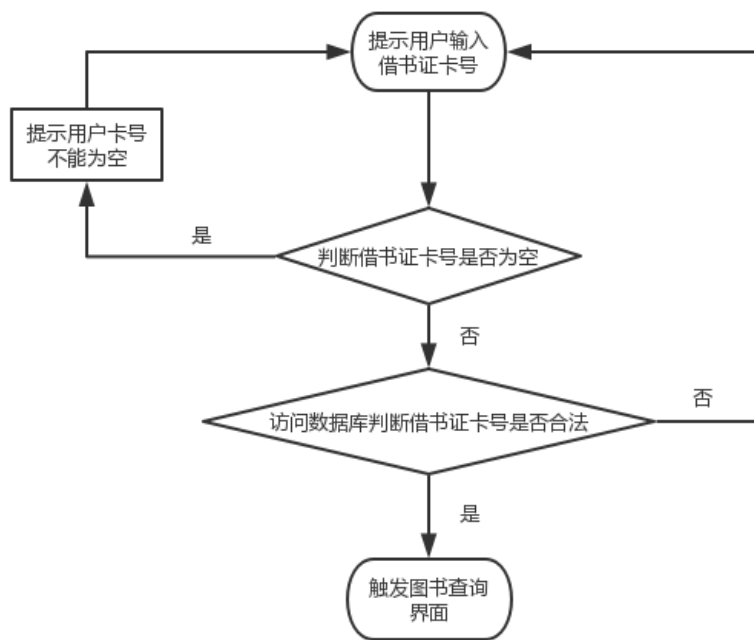


界面：



### 2.2.2 读者登录

流程图



界面

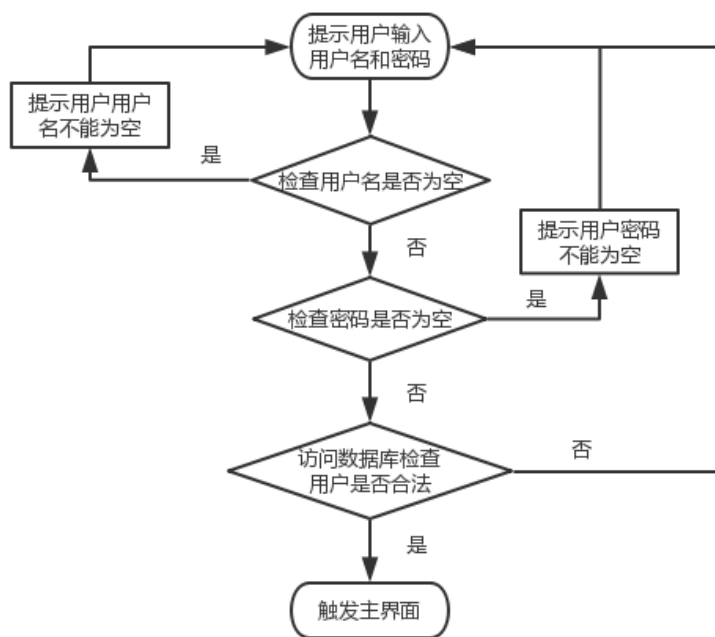


实现

接收输入的借书证卡号,通过构造函数封装在一个 LibraryCard 对象中,传给 DAO 类中的 SelectCard\_ID 函数,连接数据库,查询是否有匹配的借书证,接收一个 ResultSet 返回值,判断若不为空,则读书证存在,若为空,则提示借书证卡号错误。

### 2.2.3 管理员登录

流程图



界面

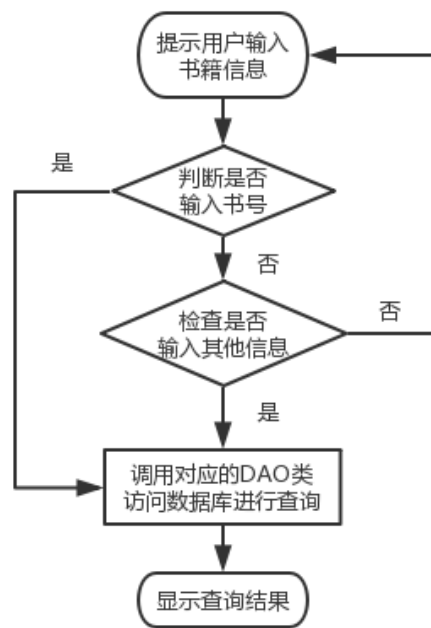
实现

接收输入的用户名和密码,通过构造函数封装在一个 **Manager** 对象中,传给 DAO 类中的 **login** 函数, **login** 函数连接数据库, 查询是否有匹配的管理员, 接收一个 **ResultSet** 返回值, 将其中的信息利用全信息构造函数封装在一个 **Manager** 对象中返回给界面事务处理函数。判断若返回值不为空, 则管理员合法, 并将此封装好的管理员信息赋值给主界面的静态变量 **manager** 以保留管理员身份信息以备后续操作; 若返回值为空, 则提示用户名或密码错误。

## 2.2.4 图书查询

流程图





## 界面

图书查询

书号：

精确查找

书名：

类别：

作者：

出版社：

年份：

至

模糊查找

价格：

至

重置

书号	类别	书名	出版社	出版年份	作者	价格	总藏书量	在架数量

实现

首先设计填表函数 `fillTable` 接收一个 `ResultSet` 做参数，并将其中的每一行的信息存入一个 `vector` 并显示在表格中。

接收输入先检查是否所有信息都为空,若是则提示用户至少输入一个检索要求。

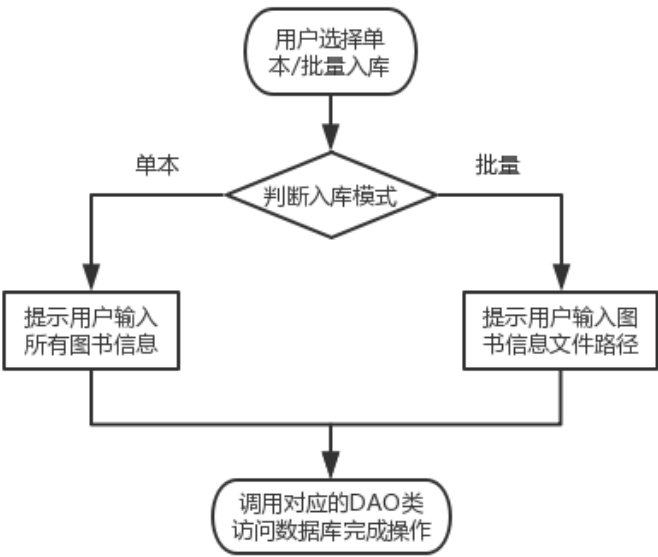
再检查是否输入了书号信息，若是则默认使用精确查找，通过 `SelectISBN` 函数传入书号并接收再数据库中查询此书号的结果集；若没有输入书号，则为模糊查找，这里使用 `StringBuffer` 来处理，逐个检查输入，若不为空，就将对应的部分 `sql` 语句压入，最后将 `StringBuffer` 转化为 `String`，通过 `PreparedStatement` 方法获得查询结果

集

将上述步骤中获得的结果集传给填表函数，显示查询结果。

### 2.2.5 图书入库

流程图



界面

The screenshot shows a window titled '图书单本入库' (Single Book入库). The window contains several input fields for book information: '书号ISBN:', '书籍类别:', '书 名:', '出版社:', '出版年份:', '作 者:', '价 格:', and '入库数量:'. Each field is followed by a text input box. At the bottom of the window, there are two buttons: '确认入库' (Confirm入库) and '重置' (Reset).

入库要求信息完整，接收输入的信息后，利用构造函数封装在一个 Book 对象中，

传给 bookDao 的 Insert 函数。Insert 函数先调用 SelectISBN 函数查询书库中是否已经有这本书，若结果集不为空，则说明这本书已存在于书库中，只需要使用 update 语句改变总藏书量，添加新的入库数量即可；若结果集为空，则说明这本书没有入过库，那么使用 insert 语句插入图书信息，并设置总藏书量即为入库数量。

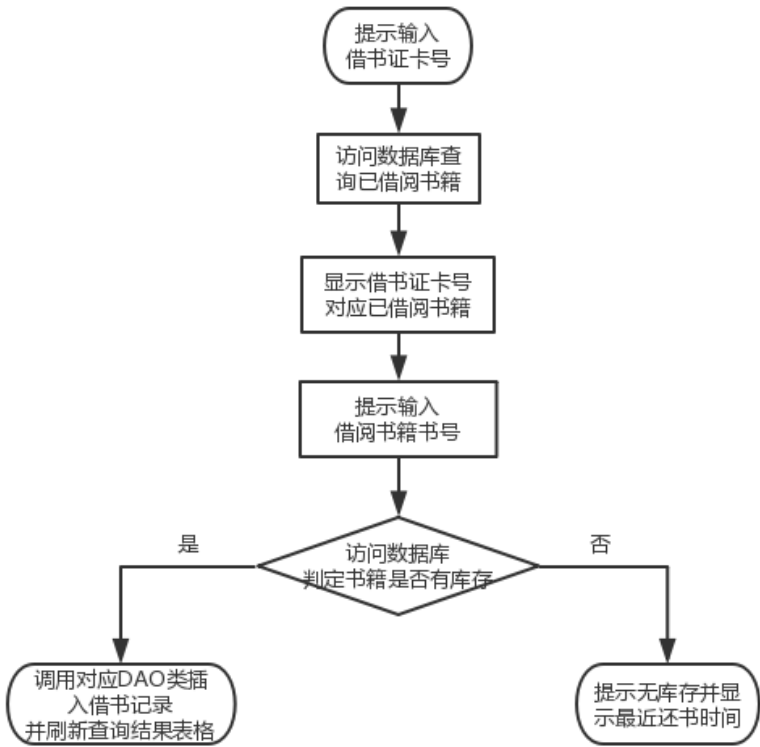


批量入库时图书信息储存在 excel 文件中，使用到 jxl 包来读取

首先检查输入的文件路径，若不合法则输出提示信息。若合法则声明一个对应的 File 对象并传给 BookBatchAdd 函数。函数接收后逐页逐行读取，把每一行的数据依次读取进来存储在一个数组 book\_information 中，调用对应的构造函数将所有信息封装在一个 book 对象中传给 bookDao 的 Insert 函数，每一行包含一本书的信息，插入过程与单本入库无异。一行处理完后再处理下一行，如此直到所有信息处理完毕。

### 2.2.6 借书

流程图



## 界面

书号	类别	书名	出版社	年份	作者	价格	借阅日期	应还日期
----	----	----	-----	----	----	----	------	------

## 实现

首先接收输入借书证卡号 `Card_ID` 传递给填表函数 `fillTable`，函数调用 `recordDao.Select` 函数查询对应借书证卡号的借书记录得到一个返回的结果集。遍历此结果集，对于每一行，提取书号信息传给 `bookDao.SelectISBN` 得到对应书籍信息的结果集。使用一个 `vector` 存储两个结果集中需要显示的信息并添加到表格中。遍历结束后得到对应借书证卡号借阅记录的表格。

接收输入的书号，调用 `bookDao.SelectISBN` 得到对应书籍信息的结果集，查询其中的在架库存数量，若大于 0，说明可以借阅，则将书号，借书证卡号和管理员 ID 封装到一个 `Record` 对象中传给 `recordDao.Insert` 函数。

这里的管理员信息不需要手动输入，而是通过在 `VIEW` 类中设置了一个静态变量，将登陆时输入的管理员信息都保留了下来，也保证了借书记录中的管理员信息与登陆的管理员相匹配。

由于在数据库中设计了触发器，当插入新的借书记录时会自动设置借书日期为当前日期，并设置默认还书日期为一个月后。且会自动为 `book` 表中对应书籍的在架库存数减一，则至此借书成功。

但若查询到的在架库存数为 0，说明此书籍所有藏书已全部借出，那么利用 `StringBuffer` 设计 `sql` 语句查询此书籍最近的归还日期。此处 `sql` 语句稍有些复杂，在此列出加强说明。

```
StringBuffer sql=new StringBuffer("select * from (select * from record where Book_ISBN='"+ISBN);
sql.append(ISBN);
sql.append("'') as t inner join (select min(Return_date) as C from record where Book_ISBN='"+ISBN);
sql.append(ISBN);
sql.append("'') as A on t.Return_date=A.C;");
PreparedStatement pstmt=conn.prepareStatement(sql.toString());
ResultSet latest_date = pstmt.executeQuery();
```

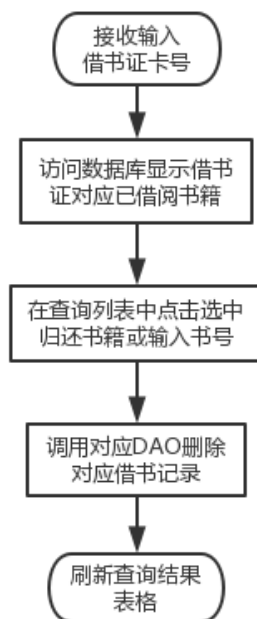
由上便得到了最近归还日期的查询结果，这里需要将得到的日期格式数据转化为

字符串向用户显示，使用了 SimpleDateFormat 来完成，代码如下。

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  
String date = sdf.format(latest_date.getDate("Return_date"));
```

## 2.2.7 还书

流程图



界面

The screenshot shows a window titled "归还书籍" (Return Book). It contains a search section with a label "借书证卡号:" followed by a text input field and a "查询" (Search) button. Below this is a table with the following headers: "书号" (Book ID), "类别" (Category), "书名" (Book Name), "出版社" (Publisher), "年份" (Year), "作者" (Author), "价格" (Price), "借阅日期" (Borrow Date), and "应还日期" (Due Date). The table body is currently empty. At the bottom, there is a label "归还书号:" followed by another text input field and a "确认归还" (Confirm Return) button.

实现

借书记录检索方法同借书部分，不再赘述。

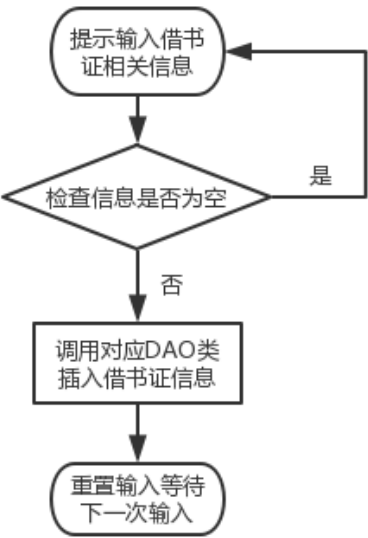
这里添加了一个表格行点击事件处理方法 RecordTableMousePressed 支持用户可

以通过点击借书记录选择归还的书籍，也可以手动输入归还书号，效果相同。但不能既不点击选中记录也不输入要归还的书号，这样会提示用户书号不可为空。

接收到要归还的书号后调用 `recordDao.Delete` 函数删除对应借书记录。数据库中设计了触发器可以自动为 `book` 表中对应书籍的在架库存数加一，则至此归还书籍成功。

### 2.2.8 添加借书证

流程图



界面

借书证添加

读者姓名:

工作单位:

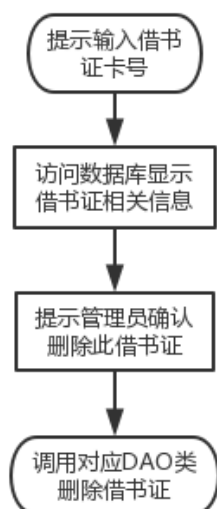
身份类别:

实现

方法与图书单本入库类似，但这里工作单位允许为空。

### 2.2.9 删除借书证

流程图



界面

实现

这里与归还图书方法类似，通过卡号查询到借书证信息并显示在表格中。支持点击选中记录，但由于必须输入卡号，不通过查询后点击选中也可以确定需要删除的记录，故也可直接删除。

### 3 实验环境

数据库平台：MySQL.

开发工具：Java

### 4 程序源码

依设计方案为表添加约束

```

ALTER TABLE book
ADD CHECK (Number>=0);
ALTER TABLE book

```

```
ADD CHECK (Stock>=0);
ALTER TABLE library_card
ADD CHECK (Type in ('student','teacher'));
```

触发器：自动设置借还书日期

```
delimiter $$
create trigger AutoSetDate
before INSERT on record
for each row
BEGIN
    set new.Borrow_date = curdate();
    set new.Return_date = date_add(new.Borrow_date,INTERVAL 1 month);
END$$
```

触发器：借书自动减少库存

```
create trigger AutoBorrowStock
after INSERT on record
for each row
update book set Stock=Stock-1 where ISBN=new.Book_ISBN;
```

触发器：还书自动增加库存

```
create trigger AutoReturnStock
after DELETE on record
for each row
update book set Stock=Stock+1 where ISBN=old.Book_ISBN;
```

JAVA 部分源码过长，故通过程序文件压缩包提交不在设计报告中体现

## 5 实验总结

### 1 . 实验中遇到的问题及解决过程

数据库表的设计中，varchar(n)表示长度为 n 个字节的可变长度且非 Unicode 的字符数据。nvarchar(n)表示 n 个字符的可变长度 Unicode 字符数据。Unicode 字符集是为了解决字符集这种不兼容的问题而产生的，它所有的字符都用两个字节表示，即英文字符也是用两个字节表示。Mysql5.0 后采用 utf-8 编码,所以无论汉字还是字母,都是一个长度的 char,所以不需要区分 nvarhcar 和 varchar,一律作 varchar

管理员登陆后的身份信息保留问题，考虑了很多方法，由于 JAVA 语言的严格封装，不能够使用全局变量等方式，最终决定采用在登陆后续的主界面及需要管理员信息的如借书界面中添加 public static 变量的方法，每次启动界面时将管理员信息传递过去。暂时未想到更好的方法，希望以后的学习中能有所进步

### 2 . 实验中产生的错误及原因分析

Java 中日期 Date 在两个包中都有定义 java.util.Date 和 java.sql.Date,使用时应该加上包名，如 private java.util.Date borrow\_date;

### 3 . 实验体会和收获。

通过图书管理系统的设计与实现，加深了对数据库系统原理及应用的理解。并通过作业过程，加强了 JAVA 语言，尤其是 JDBC 和 JAVA SWING 的学习，锻炼了系统编程能力。