# 一、绪论

## 1.1项目描述

某图书管理系统的主要功能是图书管理和信息查询。对于初次借书的读者，系统自动生成读者号，并与读者基本信息（姓名、单位、地址等）一起写入读者文件。

系统的图书管理功能分为四个方面：购入新书、读者借书、读者还书以及图书注销。

（1）购入新书时需要为该书编制入库单。入库单内容包括图书分类目录号、书名、作者、价格、数量和购书日期，将这些信息写入图书目录文件并修改文件中的库存总量（表示到目前为止，购入此种图书的数量）。

（2）读者借书时需填写借书单。借书单内容包括读者号和所借图书分类目录号。系统首先检查该读者号是否有效，若无效，则拒绝借书；若有效，则进一步检查该读者已借图书是否超过最大限制数（假设每位读者能同时借阅的书不超过5本），若已达到最大限制数，则拒绝借书；否则允许借书，同时将图书分类目录号、读者号和借阅日期等信息写入借书文件中。

（3）读者还书时需填写还书单。系统根据读者号和图书分类目录号，从借书文件中读出与该图书相关的借阅记录，标明还书日期，再写回到借书文件中，若图书逾期，则处以相应的罚款。

（4）注销图书时，需填写注销单并修改图书目录文件中的库存总量。

系统的信息查询功能主要包括读者信息查询和图书信息查询。其中读者信息查询可得到读者的基本信息以及读者借阅图书的情况；图书信息查询可得到图书基本信息和图书的借出情况。

## 1.2开发环境

MySQL、JDK 1.8。

## 1.3开发工具

Eclipse、IntelliJ IDEA。

## 1.4需求分析

系统角色包含读者和管理员两个角色。

系统功能性需求：图书管理功能，包含购入新书、图书注销、读者借书、读者还书四个功能。信息查询功能，包含读者信息查询和图书信息查询功能。要求借还业务信息持久化存储，用户界面为图形界面，使用面向对象方法开发。

购入新书时，不仅要求系统中增加该书籍，而且可对历史入库的书籍信息进行查询。

图书注销时，不仅要求系统中去除该书籍，而且可对历史出库的书籍信息进行查询。

图书借书功能中，要求记录读者信息、图书信息和还书时间。为简化系统设计，没有考虑对读者在借图书数量和时间设置限制。

图书还书功能中，要求记录读者信息、图书信息和还书时间，若图书节约时间超过设定的限额，则在还书时给予提示。

UML用例图：

# 二、概要设计

**系统顶层模块设计：**

系统分为界面和数据库操作两大部分。

界面部分为每个窗口生成一个类，统一放置在View包下，数据库操作部分将对某一对象的数据库操作集中编写存放到Util包下。为了将数据库的数据转化为对象，为每张数据库表编写了类，存放在Model包下。

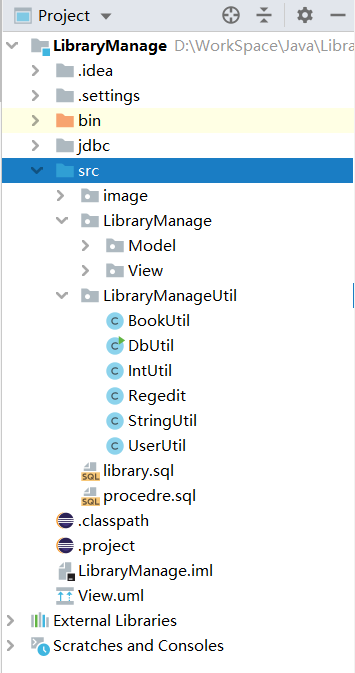
**数据库ER图设计：**



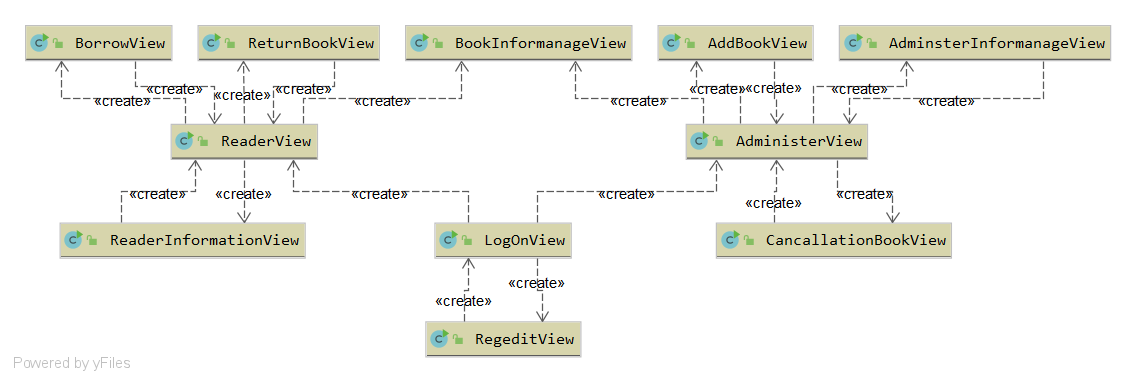
系统涉及到读者、图书、系统管理员三个实体，读者与图书间可有借阅归还关系。入库单、出库单、指定图书的库存量不存入数据库，在需要时实时生成。

# 三、详细设计及实现

**项目结构：**

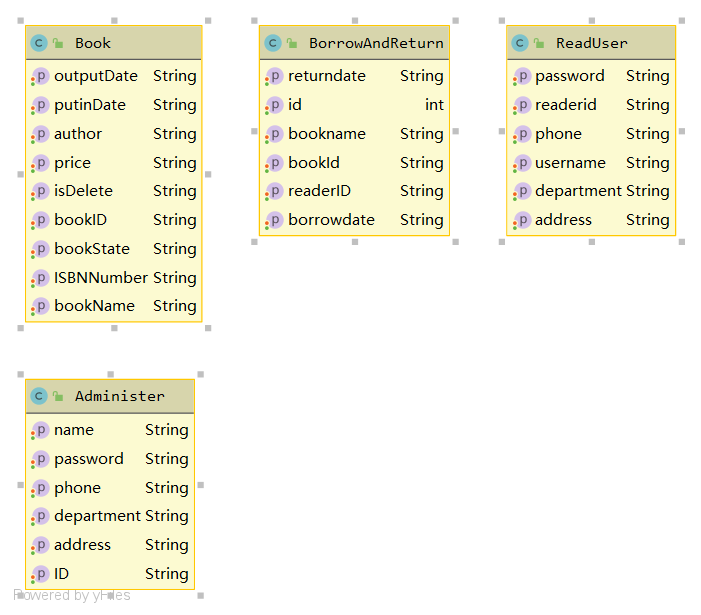
****

**系统界面类图：**

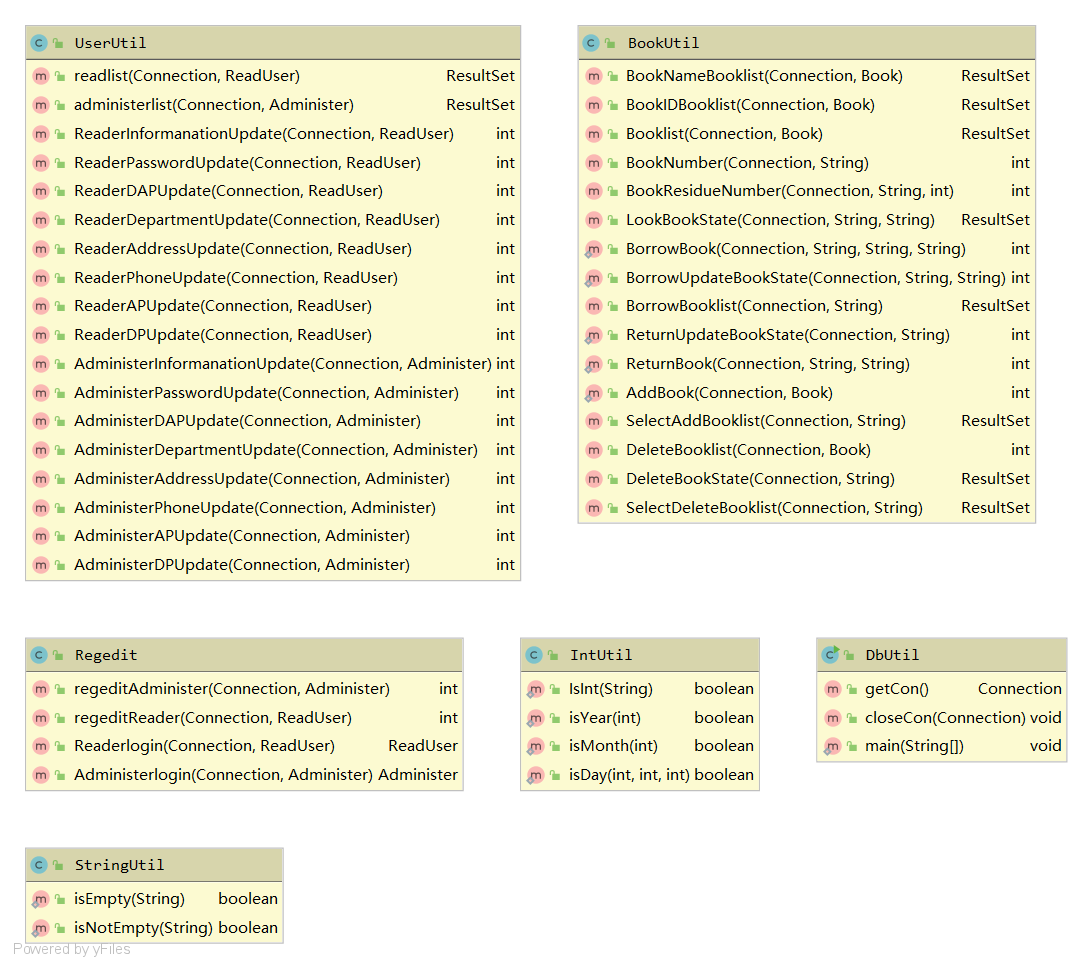


登陆界面为初始用户界面，在选择用户角色登陆后，转向对应角色的功能菜单界面，在此界面选择相应功能进入具体功能界面，可通过单击返回按钮退回到功能菜单界面。

**Model包类图**



**Util包类图：**

****

**DbUtil类部分实现：**

*/\*\*  
 \* 获取数据库连接  
 \** ***@return*** *Connection  
 \** ***@throws*** *Exception  
 \*/***public** Connection getCon()**throws** Exception{  
 Class.*forName*(**jdbacName**);  
 Connection con=DriverManager.*getConnection*(**dbUrl**, **dbUserName**, **dbpassword**);  
 **return** con;  
   
}

*/\*\*  
 \* 关闭数据库连接  
 \*/***public void** closeCon(Connection con)**throws** Exception{  
 **if**(con!=**null**) {  
 con.close();  
 }  
}

**BookUtil类部分实现：**

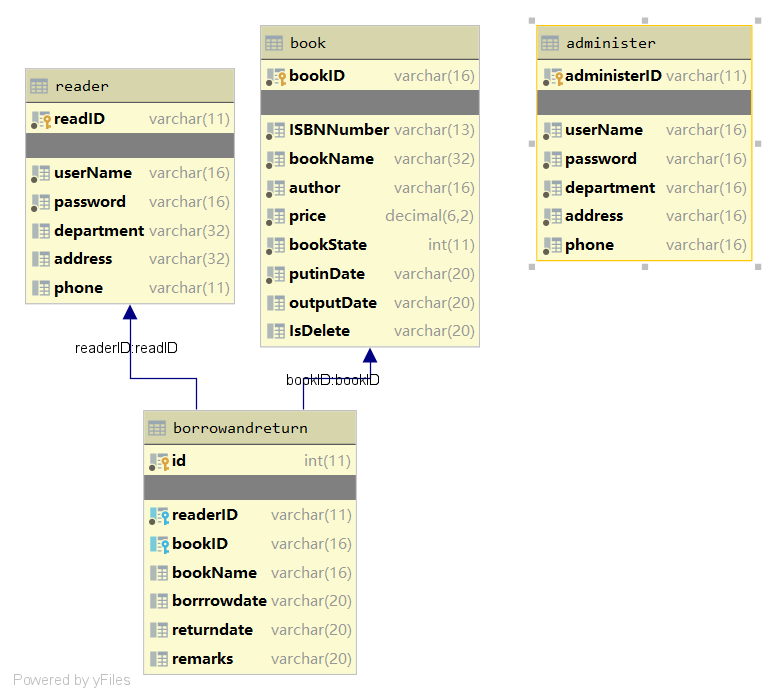
*/\*\*  
 \* 借书数据库操作  
 \*/***public static int** BorrowBook(Connection con,String bookID,String userID,String bookName) **throws** Exception{  
 String sql=**"insert into borrowandreturn values(null,?,?,?,?,null,null)"**;  
 PreparedStatement pstmt1=con.prepareStatement(sql);  
 pstmt1.setString(1, userID);  
 pstmt1.setString(2, bookID);  
 pstmt1.setString(3, bookName);  
 Date day=**new** Date();   
 SimpleDateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd"**);   
 pstmt1.setString(4, df.format(day));  
 **return** pstmt1.executeUpdate();  
}

*/\*\*  
 \* 借书更新图书的状态  
 \*/***public static int** BorrowUpdateBookState(Connection con,String bookID,String bookName) **throws** Exception{  
 String sql=**"update book set bookState=0 where bookID=?"**;  
 PreparedStatement pstmt=con.prepareStatement(sql);  
 pstmt.setString(1, bookID);  
 **return** pstmt.executeUpdate();  
}

*/\*\*  
 \* 归还图书,之后更改图书在库状态为1，标记为不可借阅  
 \*/***public static int** ReturnUpdateBookState(Connection con,String bookID) **throws** Exception{  
 String sql=**"update book set bookState=1 where bookID=?"**;  
 PreparedStatement pstmt=con.prepareStatement(sql);  
 pstmt.setString(1, bookID);  
 **return** pstmt.executeUpdate();  
}  
  
*/\*\*  
 \* 归还图书，增加归还图书时间记录  
 \*/***public static int** ReturnBook(Connection con,String bookID,String userID) **throws** Exception{  
 String sql=**"update borrowandreturn set returndate=? where bookID=? and readerID=?"** ;  
 PreparedStatement pstmt1=con.prepareStatement(sql);  
 Date day=**new** Date();  
 SimpleDateFormat df = **new** SimpleDateFormat(**"yyyy-MM-dd"**);   
 pstmt1.setString(1, df.format(day));  
 pstmt1.setString(2, bookID);  
 pstmt1.setString(3, userID);  
 **return** pstmt1.executeUpdate();  
   
}

*/\*\*  
 \* 查询借书单信息  
 \*/***public** ResultSet BorrowBooklist(Connection con,String userID)**throws** Exception{  
 StringBuffer sb=**new** StringBuffer(**"select id,readerID,bookID,bookName,borrrowdate,returndate from borrowandreturn where readerID=?"**);  
 PreparedStatement pstmt=con.prepareStatement(sb.toString());  
 pstmt.setString(1, userID);  
 **return** pstmt.executeQuery();  
}

**数据库数据模式设计：**



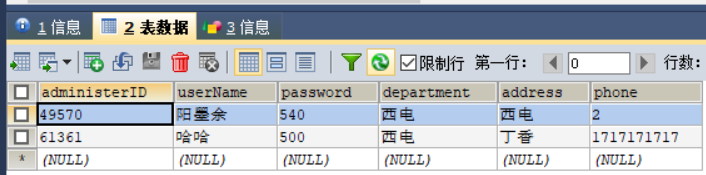
按照ER图转换为的数据模式，数据表间的关系如上图。

其中，借还表主键为流水号id，设置为自动递增，用以区分不同借书事件。图书模式中，bookState用来标记图书当前状态，区分在借（0）和在馆（1），isDelete用来标记是否已注销（1为已注销），分别作为图书借还和出库单查看时的标志变量。

# 四、系统测试

系统管理员界面：

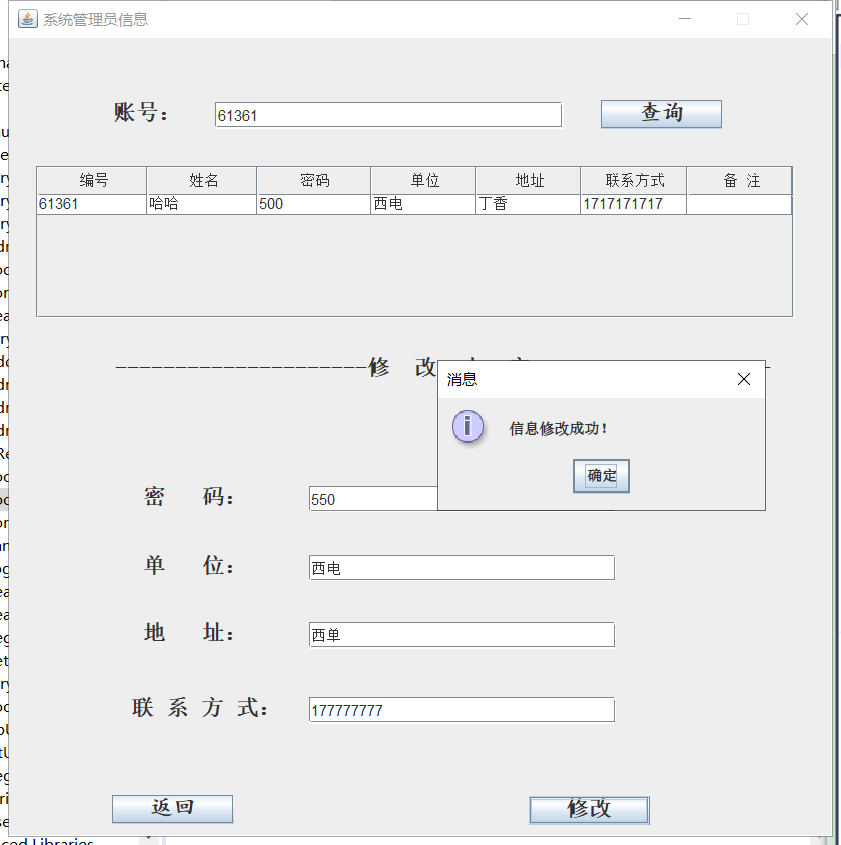


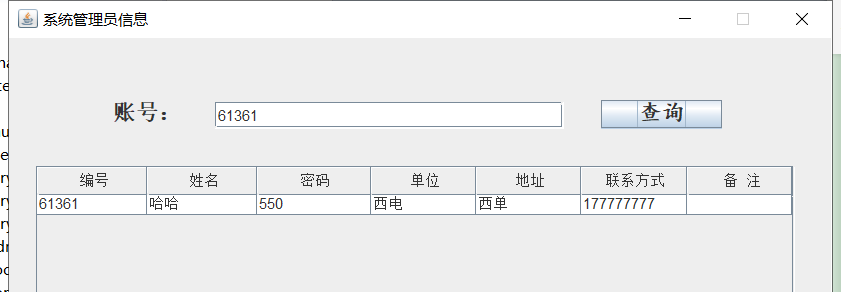


系统管理员功能界面

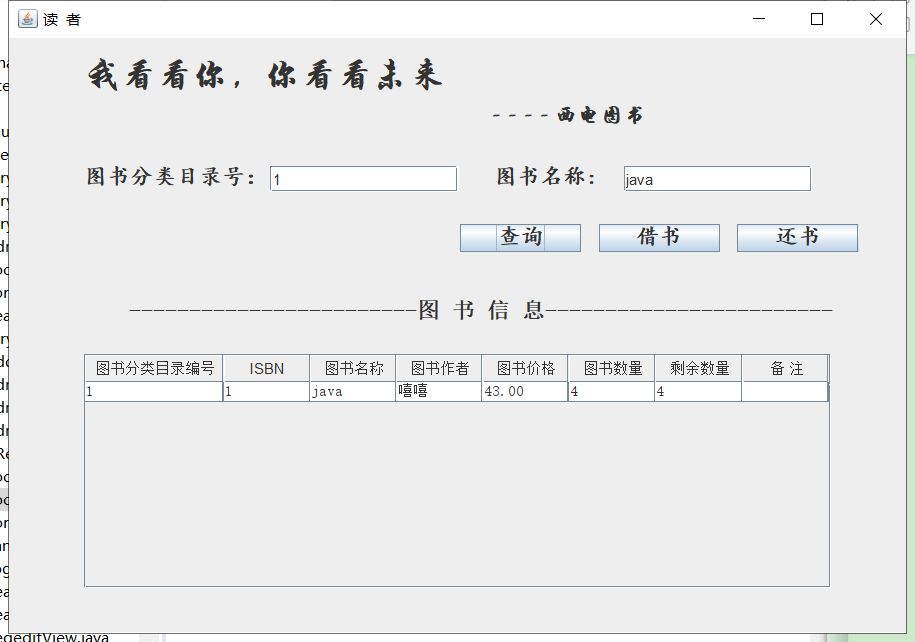


管理员个人信息修改与查询

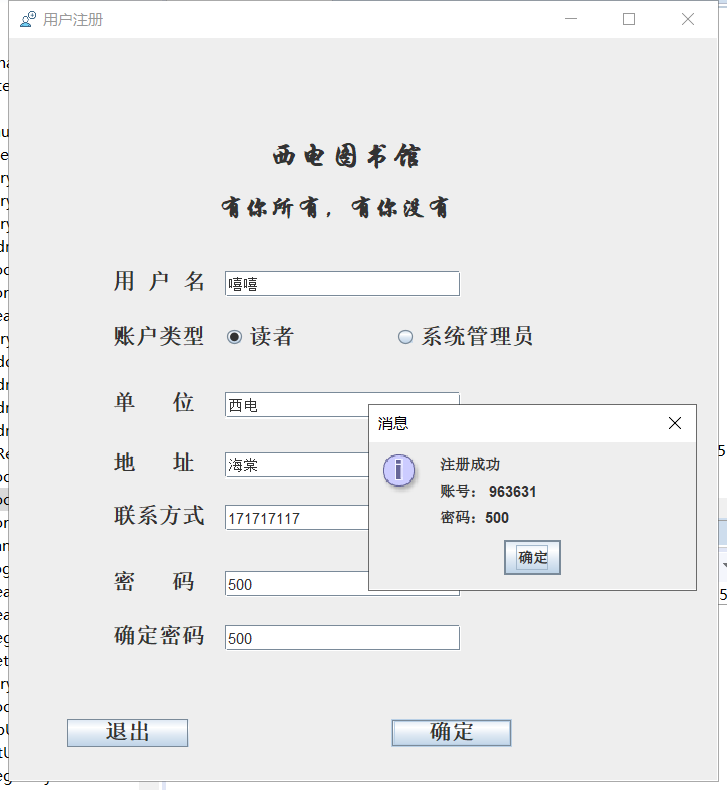


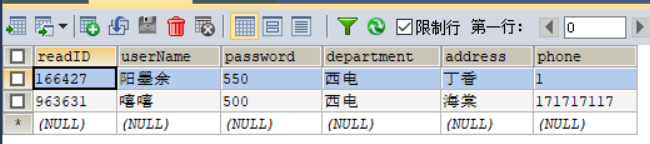


图书查询界面



读者注册界面

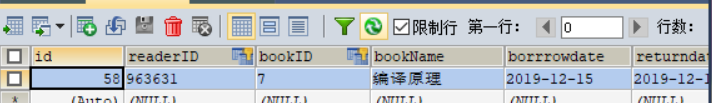




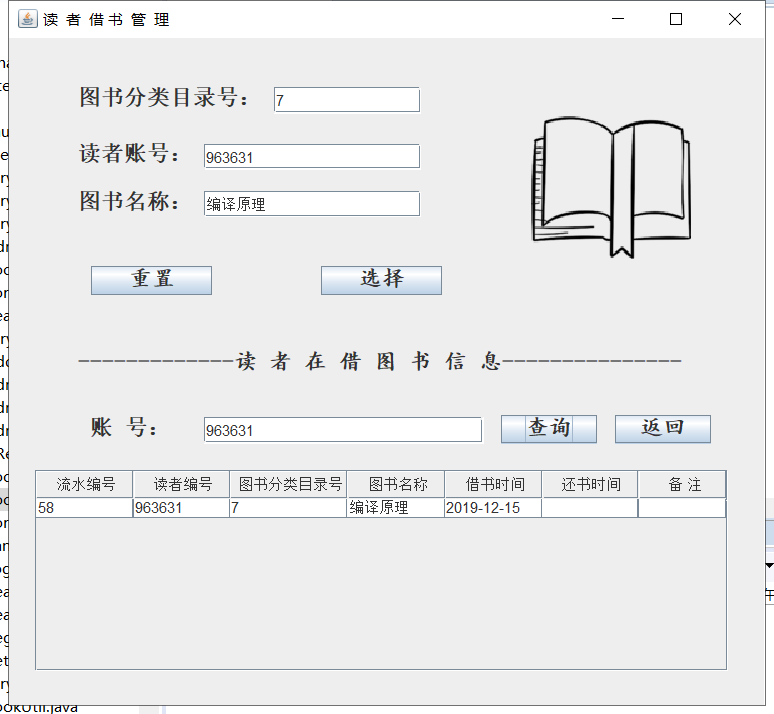
借书操作界面

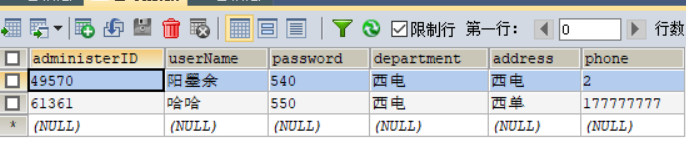


对应数据库的状态变化



借书信息查询：

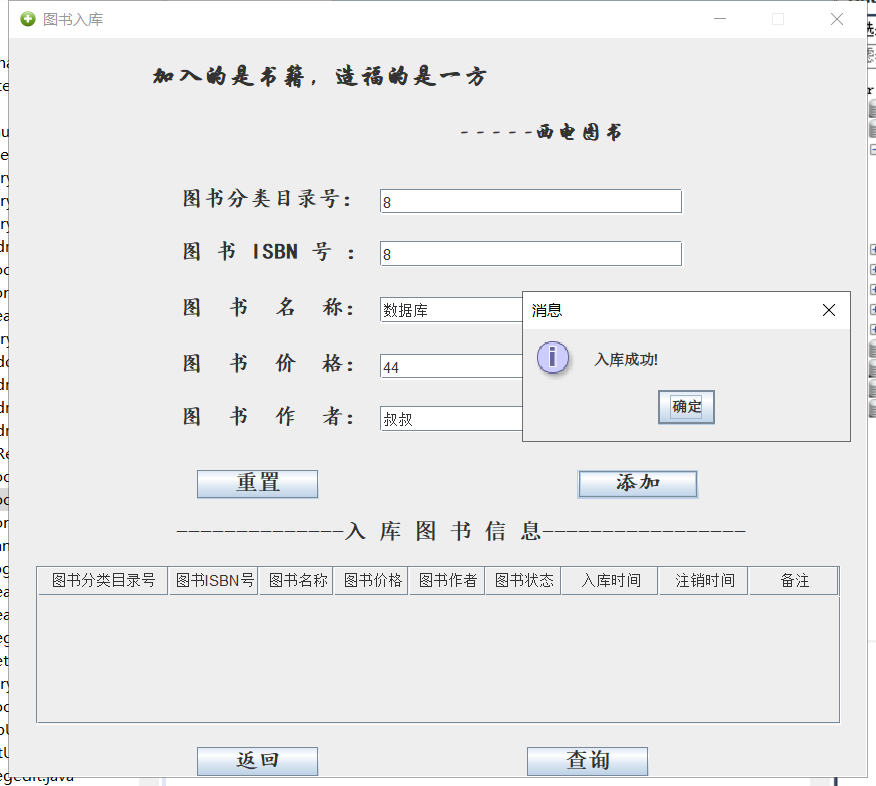


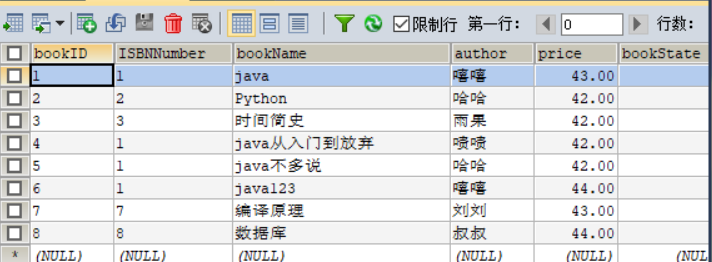


归还图书界面：



图书入库界面

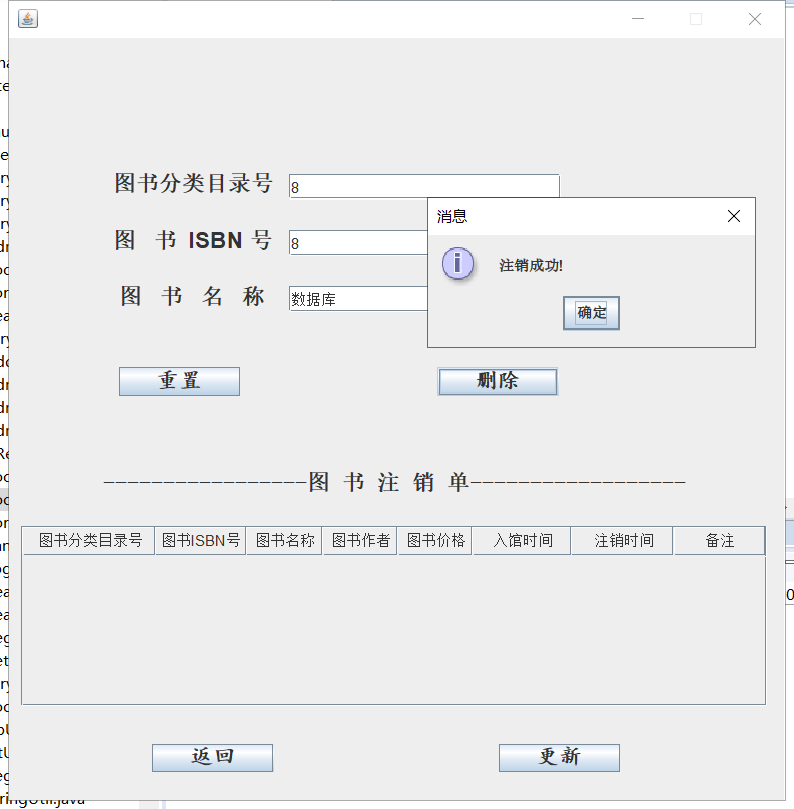




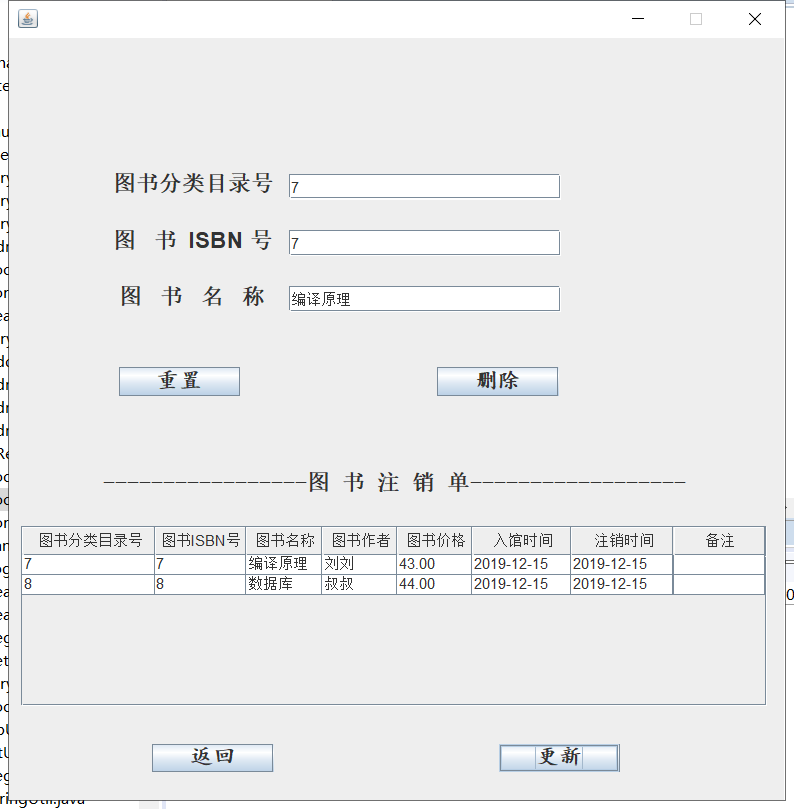
图书入库信息查询



图书注销界面：



图书注销信息查询：



# 五、总结与思考

本次程序设计是我们第一次完整的项目开发，也是第一次使用所学的知识去构架属于我们自己的系统，从根本上去理解我们日常所用的web应用或者安卓APP的底层实现，对于以前的黑箱子有了更加完整的理解，对于那些应用不再那么陌生。

其次便针对自己设计的系统而言，在实现与测试过程中，不断的去完善，不断的去更新系统，实现对自己心中构架的系统，学会如何去构架自己的构想，以及设计好的功能，对于自己构架的功能与需求进行编码翻译，从技术角度来说，我们学会怎么解决，怎么去发现问题，如何Debug，如何共同开发一项技术，相比大二的单独开发与简单程序设计，此次项目量更加多，而且流程更加繁多，同时还有自己队员的沟通，避免沟通过程中的交流不通，导致设计中的实现混乱。

之后便是我们对项目的文档记录，在开发过程中，设计了mysql数据库，对数据库概念设计做了一个完整的实现，画了一些visio图，以及系统实现流图，系统界面类图的设计，同时也学会了如何去进行需求分析，以及系统的构架，同时还要注意开发过程中的二次开发，实验必须留出再次开发的余地，以便进行增量开发。