

浙江大学实验报告

课程名称：_____数据库系统原理_____实验类型：_____

实验项目名称：_____图书管理系统_____

实验地点：_____图书馆_____实验日期：_____2019_____年_____5_____月_____2_____日

一、 实验目的和要求

1. 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。
2. 通过该图书馆系统的设计与实现，提高学生的系统编程能力，加深对数据库系统原理及应用的理解。

二、 实验内容和原理

1. 基本数据对象

对象名称	包含属性
书	书号，类别，书名，出版社，年份，作者，价格，总藏书量，库存
借书证	卡号，姓名，单位，类别（教师 学生等）
管理员	管理员 ID，密码，姓名，联系方式
借书记录	卡号，借书证号，借期，还期，经手人（管理员 ID）

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.
图书入库	1. 单本入库 2. 批量入库（方便最后测试） 图书信息存放在文件中，每条图书信息为一行，一行中的内容如下 （书号，类别，书名，出版社，年份，作者，价格，数量） Note：其中 年份、数量是整数类型；价格是两位小数类型；其余为字符串类型

	<p>Sample: (book_no_1, Computer Science, Computer Architecture, xxx, 2004, xxx, 90.00, 2)</p>
图书查询	<p>要求可以对书的 类别, 书名, 出版社, 年份 (年份区间), 作者, 价格 (区间) 进行查询. 每条图书信息包括以下内容: (书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存)</p> <p>可选要求: 可以按用户指定属性对图书信息进行排序. (默认是书名)</p>
借书	<p>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)</p> <p>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。</p>
还书	<p>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)</p> <p>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.</p>
借书证管理	增加或删除一个借书证.

除图书查询功能外, 其余功能模块都应该由图书管理员操作。

3. 用户界面

可采用图形界面或字符界面。如果采用图形或网页界面, 酌情加分。

4. 主要仪器设备

- (1) 操作系统: Windows
- (2) 数据库管理系统: SQL Server
- (3) 开发工具: QT 5.12.0

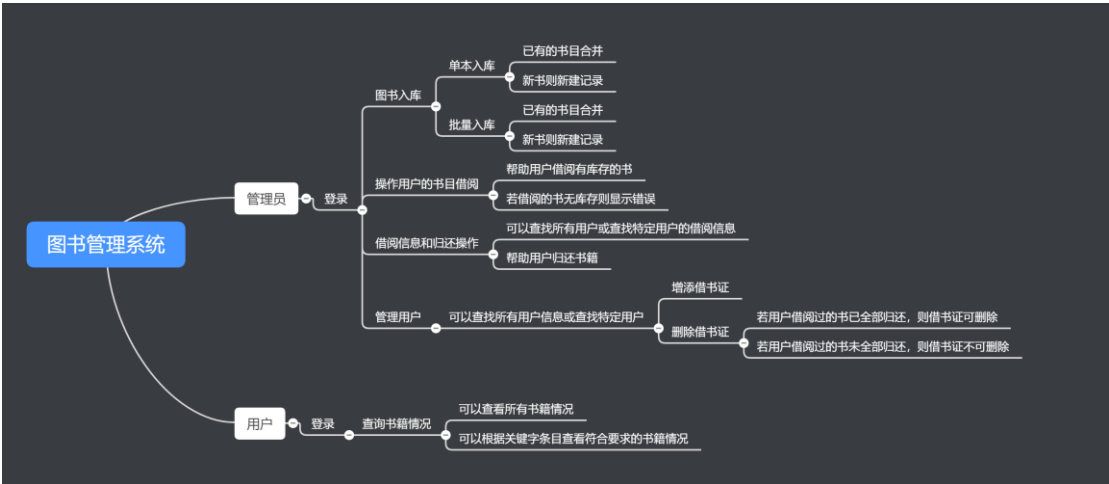
三、 实验过程与结果

1. 图书管理系统总体架构

图书管理系统中有两个角色, 一个是管理员, 一个是普通用户。本

图书管理系统最开始的界面是登录界面，管理员和普通用户通过 ID/卡号和密码可以进行登录。在登录之后，管理员可以进行图书入库、帮助用户进行书籍的借阅、查看借阅信息和归还图书、管理用户四个操作。其中图书入库操作中，图书可以单本入库，也可以批量入库，其中考虑到了图书系统中添加已有图书的情况，如果添加的图书为图书馆内已有图书，则合并到已有信息中，对应书籍的数量增多，如果是新的图书，则建立心得记录；操作用户的书目借阅中，管理员可以帮助用户借阅有库存的书，如果节约的书无库存则显示错误，无法借书，同时可以查看所有的书籍情况，也可以根据关键字条目搜索查看符合要求的书记情况；管理用户的借阅信息和归还操作时，可以查找所有用户或特定用户的借阅信息并帮助用户归还书籍；管理员管理用户时，可以查看所有用户或查找特定用户的借书证信息，可以增添借书证，也可以删除借书证，其中需要注意的是，如果要删除的借书证用户借阅过的书未全部归还，则借书证不可删除。用户可以进行对图书的搜索，可以查看所有的书籍情况，也可以根据关键字条目搜索查看符合要求的书记情况。

图书管理系统总体架构图如下：



2. 数据库表的设计

(1) 书籍信息表 book

字段名	类型	主键	外键依赖
书号 (bno)	char (8)	√	×
类别 (category)	char (10)	×	×
书名 (title)	varchar (40)	×	×
出版社 (press)	varchar (30)	×	×

年份(year)	int	×	×
作者(author)	varchar(20)	×	×
价格(price)	decimal(7,2)	×	×
总藏书量(total)	int	×	×
库存(stock)	int	×	×

(2) 管理员信息表 admin

字段名	类型	主键	外键依赖
管理员 ID(ID)	varchar(10)	√	×
密码(password)	varchar(16)	×	×
姓名(name)	varchar(10)	×	×
联系方式(contact)	varchar(20)	×	×

(3) 借书证信息表 card

字段名	类型	主键	外键依赖
卡号(cno),	char(10)	√	×
密码(password)	char(16)	×	×
姓名(name)	varchar(10)	×	×
单位(department)	varchar(40)	×	×
类别 (type)	char(1)	×	×

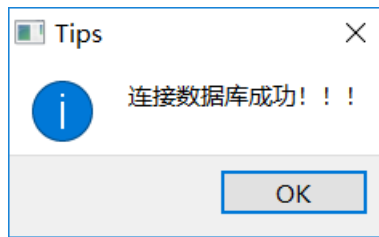
(4) 借阅信息表 borrow

字段名	类型	主键	外键依赖
卡号(cno)	char(10)	×	card 表
书号(bno)	char(8)	×	book 表
经手人 (管理员 ID) (ID)	char(10)	×	admin 表
借期(borrow_date)	char(20)	×	×
还期(return_date)	char(20)	×	×

3. 图书管理系统分功能的设计及实现

(1) 连接数据库

首先连接到数据库，实现界面如下：

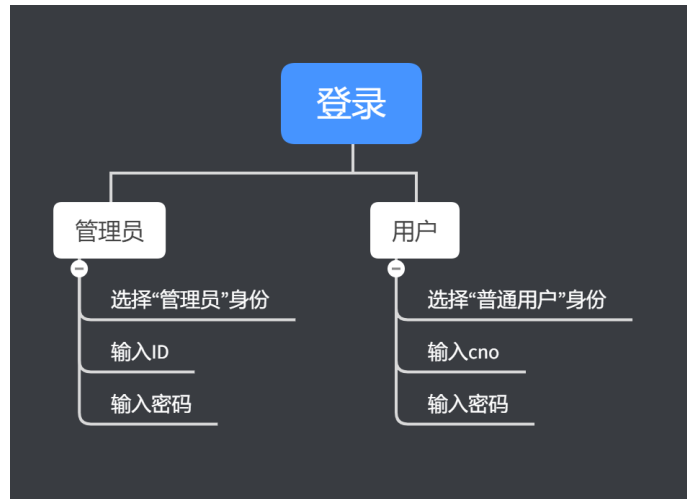


具体实现代码如下：

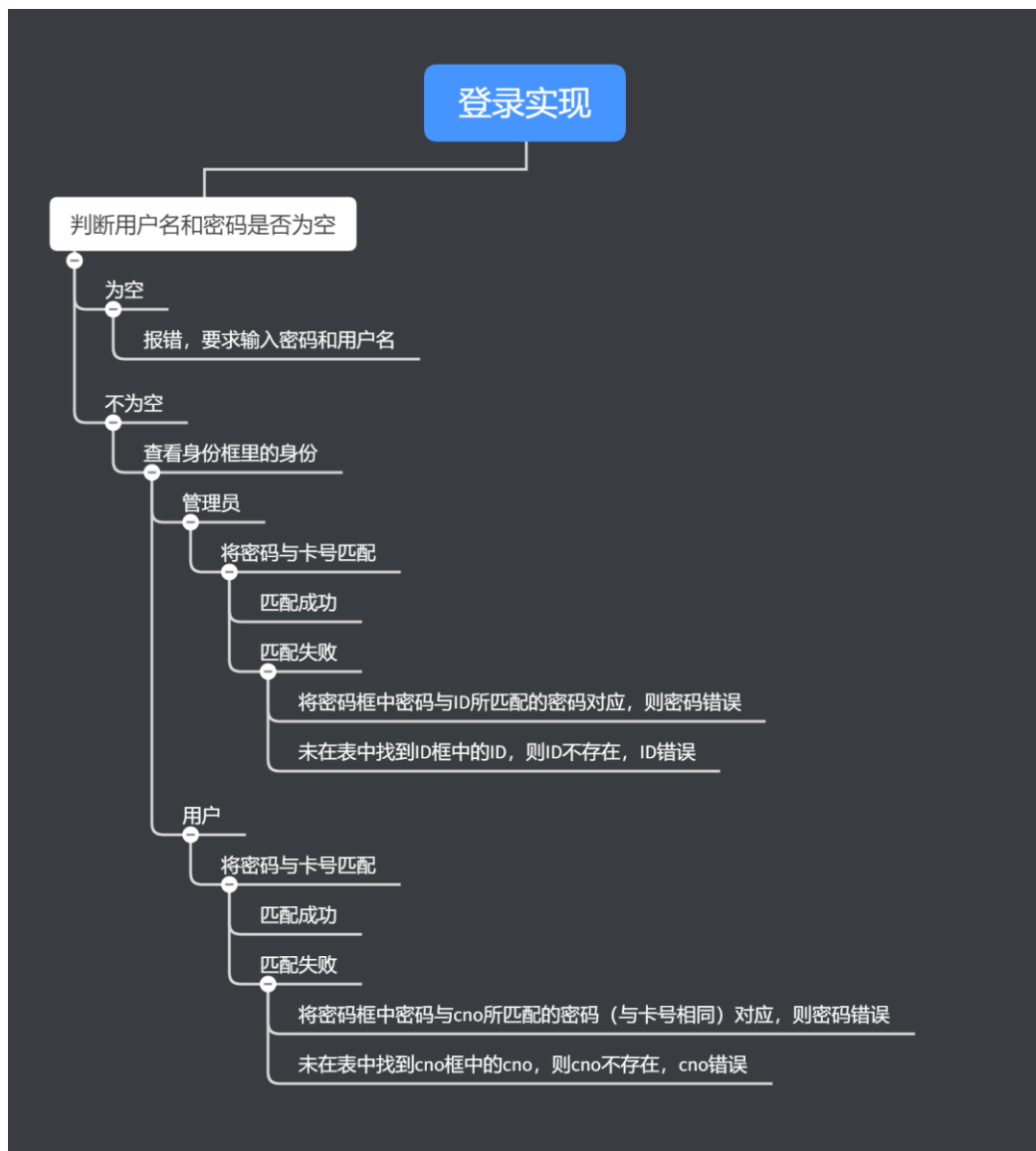
```
1. db = QSqlDatabase::addDatabase("QMYSQL");
2. db.setHostName("127.0.0.1");
3. db.setPort(3306);
4. db.setDatabaseName("booksystemmanager");
5. db.setUserName("root");
6. db.setPassword("111111");
7.
8. if(!db.open())//连接数据库失败
9. {
10.     QMessageBox::critical(nullptr,QObject::tr("连接数据库失败!!!"),db.lastError().text());
11.     return false;
12. }
13. else//连接数据库成功
14. {
15.     QMessageBox::information(nullptr,QObject::tr("Tips"),QObject::tr("连接数据库成功!!!"));
16.     cardQuery = QSqlQuery(db);
17.     borrowQuery = QSqlQuery(db);
18.     bookQuery = QSqlQuery(db);
19.     adminQuery = QSqlQuery(db);
20.     return true;
21. }
```

(2) 管理员和用户的登录

在登录界面，功能流程图如下：



具体实现方法流程图：



具体实现方法：

首先判断用户名和密码是否为空，如果为空则需要报错，不为空则查看身份

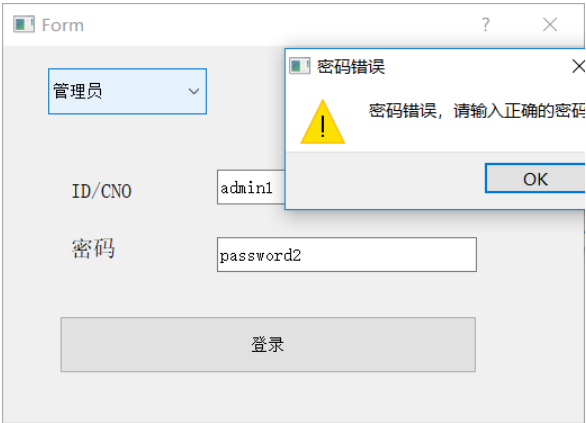
框里的身份，如果是管理员则将密码与 ID 匹配，如果匹配成功则登录成功，将密码框中密码与 ID 所匹配的密码对应，如果匹配成功则登录成功，如果匹配失败则密码错误，如果未在表中找到 ID 框中的 ID，则 ID 不存在，ID 错误；如果是普通用户则将密码与 cno 匹配，如果匹配成功则登录成功，将密码框中密码与 cno 所匹配的密码对应，如果匹配成功则登录成功，如果匹配失败则密码错误，如果未在表中找到 cno 框中的 cno，则 cno 不存在，cno 错误。

登录界面如下：



The screenshot shows a login window titled "Form". At the top left is a dropdown menu currently set to "普通用户" (Ordinary User). Below it are two input fields: the first is labeled "ID/CNO" and the second is labeled "密码" (Password). At the bottom of the form is a large button labeled "登录" (Login).

管理员登录密码错误：



This screenshot shows the same login window, but the dropdown menu is now set to "管理员" (Administrator). The "ID/CNO" field contains the text "admin1" and the "密码" field contains "password2". A modal dialog box titled "密码错误" (Password Error) is overlaid on the form. The dialog contains a yellow warning triangle icon and the text "密码错误，请输入正确的密码" (Password error, please enter the correct password). There is an "OK" button at the bottom right of the dialog.

管理员登录密码正确：

MainWindow

退出

当前管理员:admin1

图书入库 书目借阅 借阅信息/还书 管理用户

书号

类别

书名

出版社

出版年份

作者

价格

数量

导入

批量导入

普通用户登录密码错误:

Form

普通用户

ID/CNO

密码

登录

密码错误

密码错误, 请输入正确密码

OK

普通用户登录密码正确:

普通用户

退出

当前用户:cno1

类别

书名

出版社

出版年份 ----

作者

价格 ----

搜索 查询所有书目 刷新

借书人 借阅

书号	类别	书名	出版社	出版年份
----	----	----	-----	------

具体代码实现如下：

```
1. if(ui->UserNameContent->text() == "" || ui->PasswordContent->text() == "")//  
   如果输入的用户名为空或者密码为空  
2.     QMessageBox::warning(nullptr, tr("您所输入的用户名/密码为空"), tr("您所输入的  
   用户名/密码为空，请重新输入"));  
3. else{//用户输入了用户名和密码  
4.     if(ui->IdentityComboBox->currentIndex() == 1){//登录者为管理员  
5.         adminQuery.exec("select password from admin where ID = '"  
6.                             + ui->UserNameContent->text() + "'");  
7.         if(!adminQuery.isActive()) return;  
8.         else{  
9.  
10.            if(adminQuery.next()){  
11.                QString password = adminQuery.value(0).toString();  
12.                if(QString::compare(password, ui->PasswordContent->text()) !=  
   = 0)//密码错误  
13.                {  
14.                    QMessageBox::warning(nullptr, tr("密码错误"), tr("密码错  
   误，请输入正确的密码"));  
15.                    return;  
16.                }  
17.            }  
18.            else{//ID 错误  
19.                QMessageBox::warning(nullptr, tr("ID 错误"), tr("ID 错误，请输入  
   正确的 ID"));  
20.                return;  
21.            }  
22.            AdminInterface *ai = new AdminInterface(nullptr, ui->UserNameCon  
   tent->text());//要传入个人信息  
23.            ai->show();  
24.            close();  
25.        }  
26.    }  
27. else{//登录者为学生  
28.     cardQuery.exec("select password from card where cno = '"  
29.                         + ui->UserNameContent->text() + "'");  
30.     if(!cardQuery.isActive()) return;  
31.     else{  
32.         if(cardQuery.next()){  
33.             QString password = cardQuery.value(0).toString();  
34.             if(QString::compare(password, ui->PasswordContent->text()) !=  
   = 0)//密码错误  
35.             {
```

```

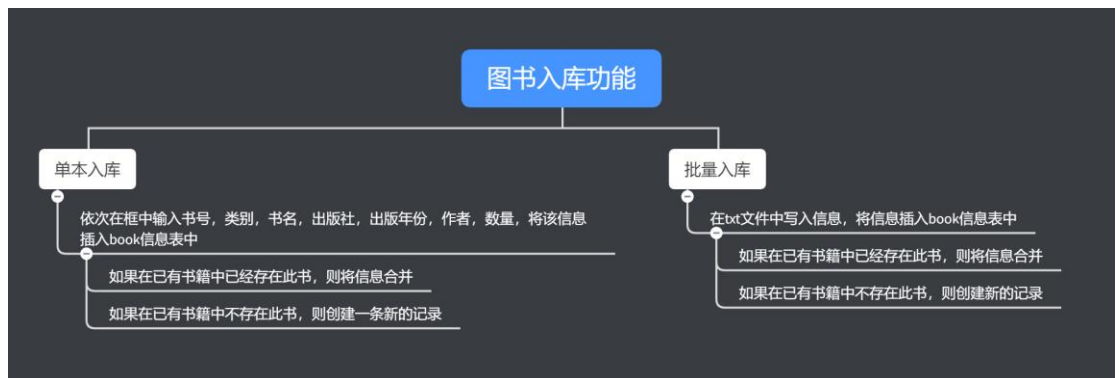
36.             QMessageBox::warning(nullptr, tr("密码错误"), tr("密码错
           误, 请输入正确密码"));
37.             return;
38.         }
39.     }
40.     else{//卡号错误
41.         QMessageBox::warning(nullptr, tr("卡号错误"), tr("卡号错误, 请
           输入正确卡号"));
42.         return;
43.     }
44.
45.     UserInterface *u = new UserInterface(0, ui->UserNameContent->tex
           t());//要传入个人信息
46.     u->show();
47.     close();
48. }
49. }

```

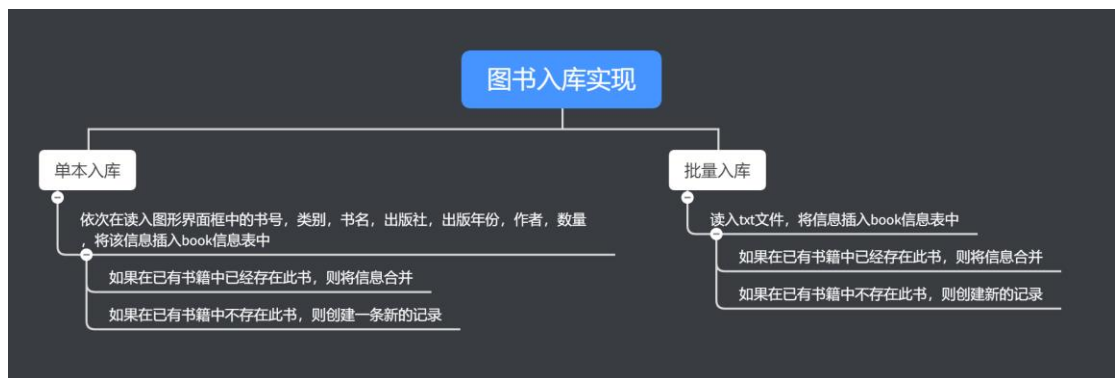
(3) 管理员登录

1) 图书入库

图书入库的功能流程图如下：



图书入库的具体实现流程图如下：



图书入库的具体实现过程：

①单本入库

单本入库时，依次在读入各个框中的书号，类别，书名，出版社，出版年份，作者，数量，将该信息插入 **book** 信息表中，如果在已有书籍中已经存在此书，则将信息合并，如果在已有书籍中不存在此书，则创建一条新的记录。

②批量入库

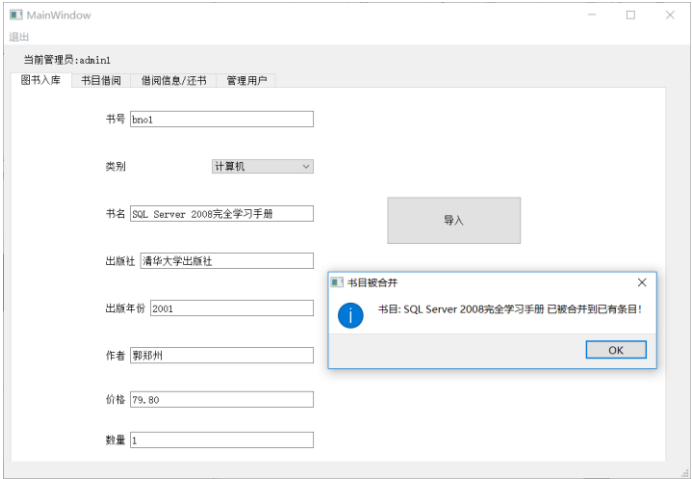
批量入库时，读入 **txt** 文件，将信息插入 **book** 信息表中，如果在已有书籍中已经存在此书，则将信息合并，如果在已有书籍中不存在此书，则创建新的记录。

图书入库的界面：

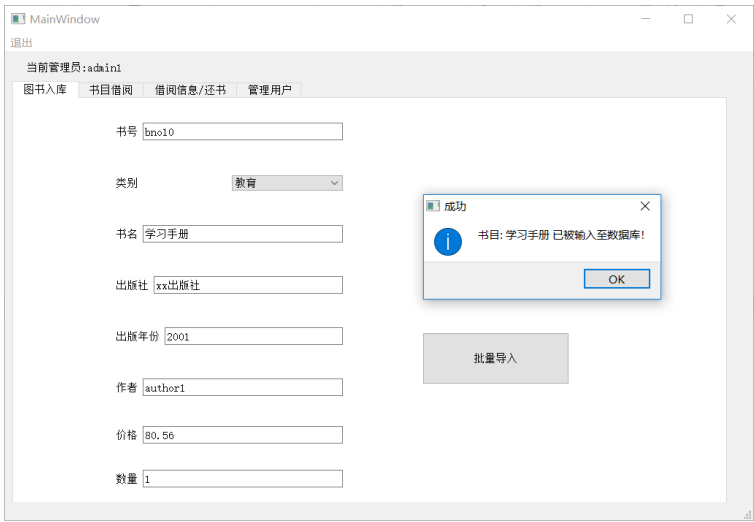


①单本入库：

导入已有书目：



导入已有书籍中没有的书目

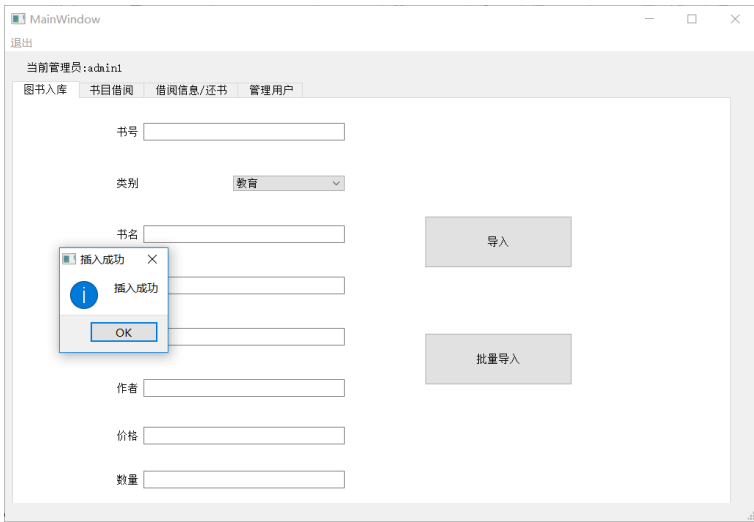


②批量入库：

导入已有书目：



导入已有书籍中没有的书目



初始数据库中书籍：

bno	category	title	press	year	author	price	total	stock
bno1	计算机	SQL Server 2008完全学习手册	清华大学出版社	2001	郭郑州	79.80	2	2
bno2	计算机	程序员的自我修养	电子工业出版社	2013	俞甲子	65.00	5	1
bno3	教育	做新教育的行者	福建教育出版社	2002	高云鹏	25.00	1	1
NULL	NULL		NULL	NULL	NULL	NULL	NULL	NULL

插入完成后数据库中书籍：

bno	category	title	press	year	author	price	total	stock
bno1	计算机	SQL Server 2008完全学习手册	清华大学出版社	2001	郭郑州	79.80	6	4
bno10	教育	学习手册	xx出版社	2001	author1	80.56	1	1
bno2	计算机	程序员的自我修养	电子工业出版社	2013	俞甲子	65.00	5	1
bno3	教育	做新教育的行者	福建教育出版社	2002	高云鹏	25.00	3	2
bno4	教育	做孩子眼中有本领的父母	电子工业出版社	2013	高云鹏	23.00	5	5
bno5	英语	实用英文写作	高等教育出版社	2008	庞继贤	33.00	3	1
bno6	计算机	数据库系统原理教程	清华大学出版社	1998	王珊	35.00	12	11
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

实现的源代码：

①单本入库：

```
1. //插入的书籍信息
2. QString bookbno = ui->bookbno->text();
3. QString bookcategory = ui->bookcategory->currentText();
4. QString booktitle = ui->booktitle->text();
5. QString bookpress = ui->bookpress->text();
6. QString bookyear = ui->bookyear->text();
7. QString bookauthor = ui->bookauthor->text();
8. QString bookprice = ui->bookprice->text();
9. QString booktotal = ui->booktotal->text();
10.
11.
12. if(QString::compare(bookcategory, "") == 0 ||
13.     QString::compare(bookbno, "") == 0 ||
14.     QString::compare(booktitle, "") == 0 ||
15.     QString::compare(bookpress, "") == 0 ||
16.     QString::compare(bookyear, "") == 0 ||
17.     QString::compare(bookauthor, "") == 0 ||
18.     QString::compare(bookprice, "") == 0 ||
19.     QString::compare(booktotal, "") == 0){
20.
21.     QMessageBox::warning(nullptr, "输入信息不完整", "请输入完整且恰当的信息！");
22.     return;
23. }
24.
25. bookQuery.exec("select * from book where title = '" + booktitle + "' "
```

```

26.         + "and press = '" + bookpress + "' "
27.         + "and category = '" + bookcategory + "' "
28.         + "and bno = '" + bookbno + "' "
29.         + "and year = '" + bookyear + "' "
30.         + "AND author = '" + bookauthor + "' "
31.         + "AND price = '" + bookprice + "' ");
32.
33. if(!bookQuery.isActive())
34. {
35.     QMessageBox::warning(nullptr, "错误", "查询错误");
36.     return;
37. }
38.
39. if(bookQuery.next())
40. {
41.     bookQuery.exec("update book set total = total + '" + booktotal +
42.                    "' where bno = '" + bookbno + "'");
43.     bookQuery.exec("update book set stock = stock + '" + booktotal +
44.                    "' where bno = '" + bookbno + "'");
45.     QMessageBox::information(nullptr, "书目被合并", "书
        目: " + booktitle + " 已被合并到已有条目!");
46. }
47. else
48. {
49.     bookQuery.exec("insert into book values("
50.                    "'" + bookbno + "',"
51.                    "'" + bookcategory + "',"
52.                    "'" + booktitle + "',"
53.                    "'" + bookpress + "',"
54.                    "'" + bookyear + "',"
55.                    "'" + bookauthor + "',"
56.                    "'" + bookprice + "',"
57.                    "'" + booktotal + "',"
58.                    "'" + booktotal + "'"");
59.     if(bookQuery.isActive())
60.         QMessageBox::information(nullptr, "成功", "书目: " + booktitle + " 已
        被输入至数据库!");
61. }

```

②批量入库:

```

1. QFile file("D:\\QTproject\\final\\data.txt");
2. if(!file.open(QIODevice::ReadOnly | QIODevice::Text)) {
3.     qDebug()<<"Can't open the file!"<<endl;

```

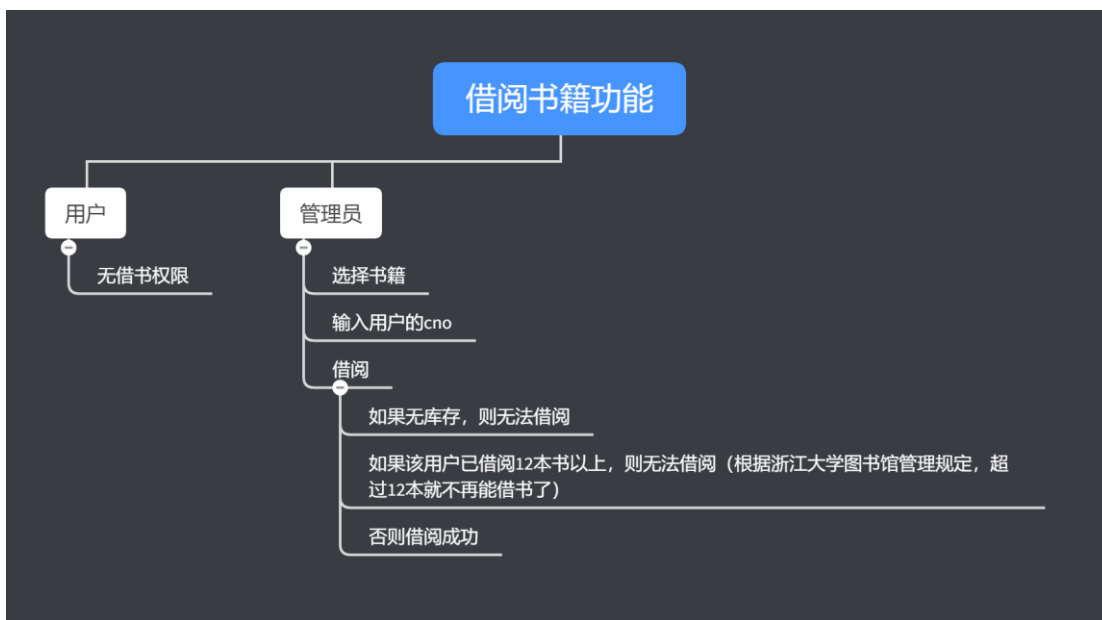
```

4.  }
5.  while(!file.atEnd()) {
6.      QByteArray line = file.readLine();
7.      QTextCodec *codec=QTextCodec::codecForName("GB2312");
8.
9.      QString l=codec->toUnicode(line);
10.     QString booktotal = l.section(',',7,7);
11.     QString bookbno = l.section(',',0,0);
12.     bookbno = bookbno.mid(1,4);
13.     QString bookstock = l.section(',',8,8);
14.     bookQuery.exec("select * from book where bno = '" + bookbno + '"
    ");
15.
16.     if(!bookQuery.isActive())
17.     {
18.         QMessageBox::warning(nullptr, "错误", "查询错误");
19.         return;
20.     }
21.
22.     if(bookQuery.next())
23.     {
24.         bookQuery.exec("update book set total = total + '" + booktotal +
    al +
25.             "' where bno = '" + bookbno + "'");
26.         bookQuery.exec("update book set stock = stock + '" + bookstock +
    ck +
27.             "' where bno = '" + bookbno + "'");
28.         QMessageBox::information(nullptr, "书目被合并", "书目已被合并
    到已有条目!");
29.     }
30.     else{
31.         bookQuery.exec("insert into book values(" +
    l + ")");
32.         if(!bookQuery.isActive()){
33.             QMessageBox::warning(nullptr, "错误", "
    错误");
34.             return;
35.         }
36.         else{
37.             QMessageBox::information(nullptr, "插入成
    功", "插入成功");
38.         }
39.     }
40.

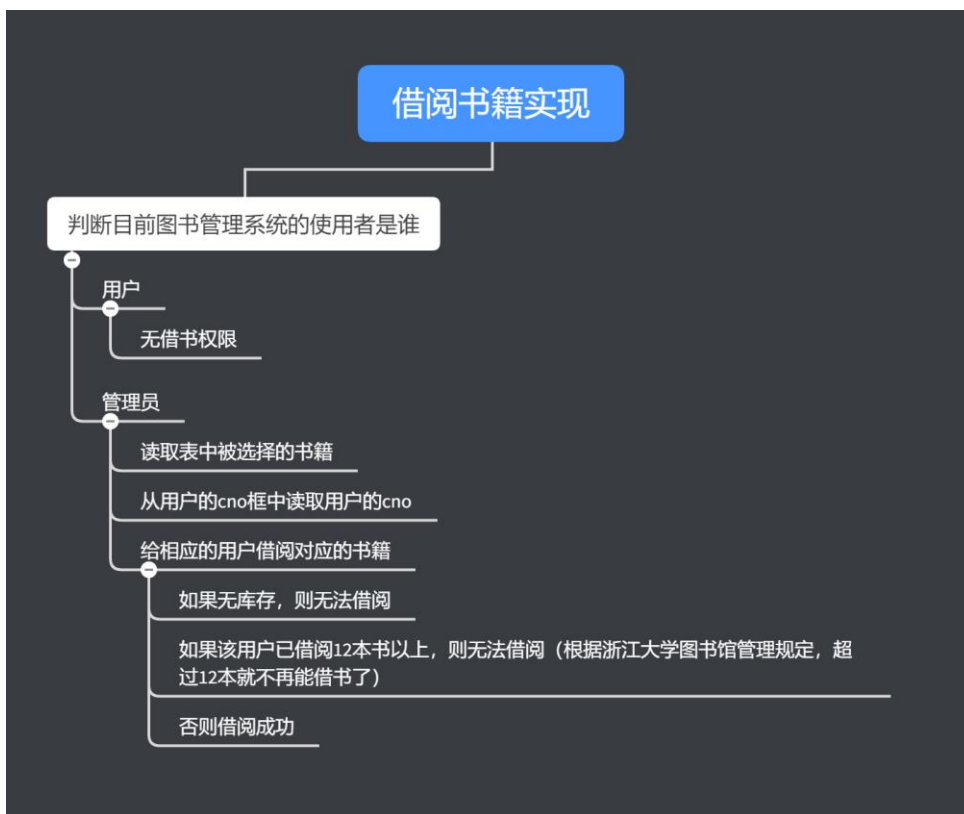
```

2) 操作用户的书目借阅（书籍搜索略去，在用户界面处详细介绍）

借阅书籍功能流程图如下：



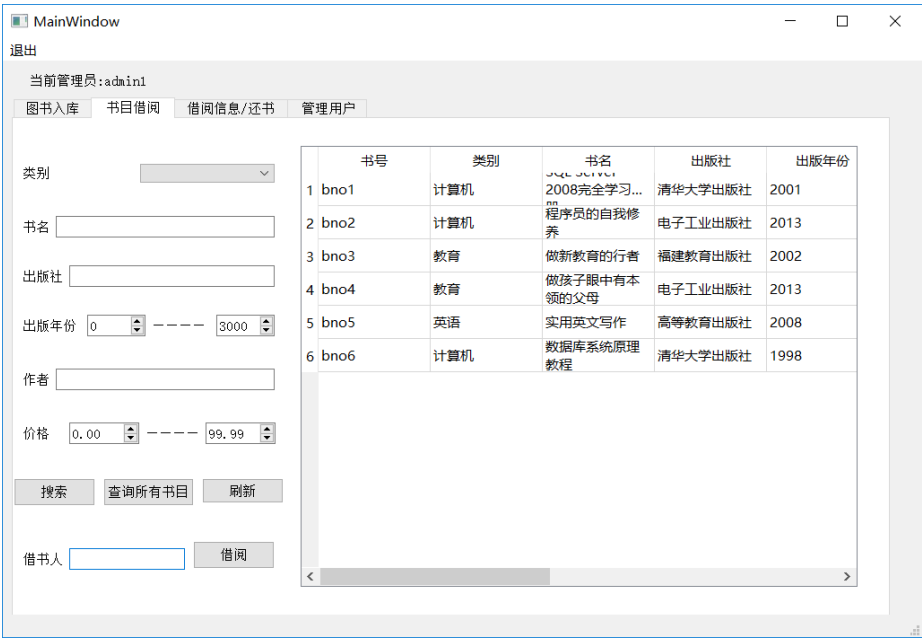
借阅书籍的具体实现流程图如下：



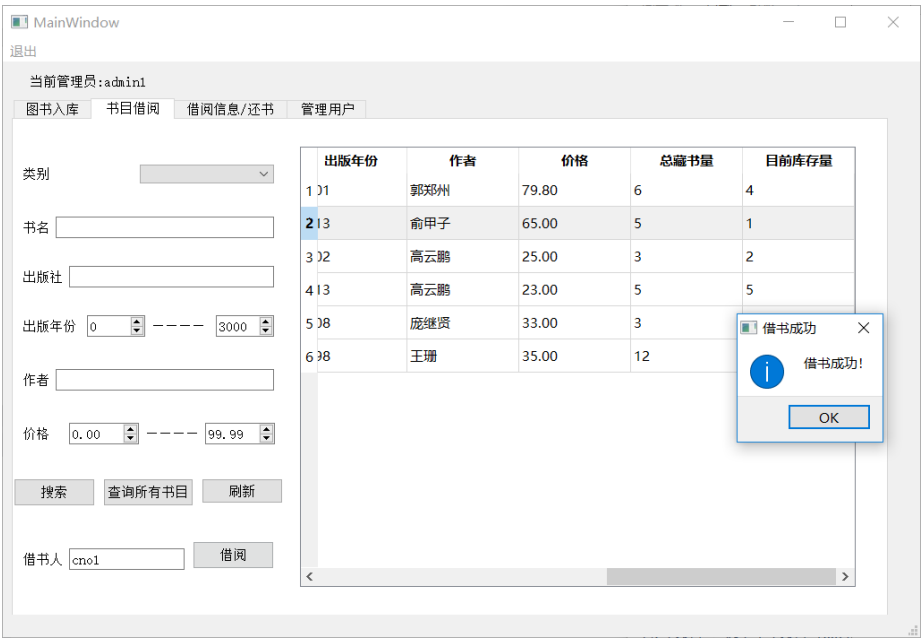
借阅书籍的具体实现：

首先需判断目前图书管理系统的使用者是谁。如果是用户操作借书则报错显示其无权限借书（根据题目要求，借书应由管理员操作）；如果是管理员操作则搜索相应的书籍，从用户的 cno 框中读取用户的 cno，给相应的用户借阅对应的书籍，其中需要注意的是，如果无库存则无法节约，如果该用户已借阅 12 本书以上则无法节约，否则则借阅成功。

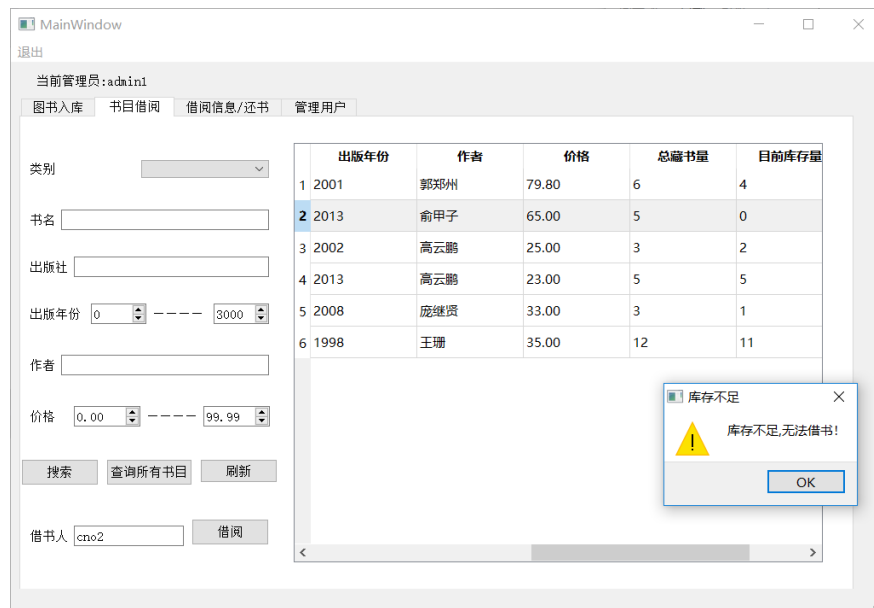
借阅书籍的界面：



用 cno1 借阅一本库存为 1 的书



用 cno2 继续借阅该库存为 0 的书



借阅功能具体代码:

```
1. adminQuery.exec("select ID from admin where ID = "
2.         + username + "'");
3.
4. if(!adminQuery.isActive())
5. {
6.     QMessageBox::warning(nullptr, "错误", "出现错误");
7.     return;
8. }
9.
10. if(!adminQuery.next())//只有管理员可以操作借书
11. {
12.     QMessageBox::warning(nullptr, "无借书权限", "您无借书权限!");
13.     return;
14. }
15. else //有权限
16. {
17.     QString borrower = ui->BorrowName->text();
18.     borrowQuery.exec("select bno from borrow where cno = "
19.         + username + "'");
20.     if(!borrowQuery.isActive())
21.     {
22.         QMessageBox::warning(nullptr, "内部错误", "出现内部错误!");
23.         return;
24.     }
```

```

25.     if(borrowQuery.size() >= 12) //根据浙江大学图书馆管理规定，超过 12 本就不能再
    借书了
26.     {
27.         QMessageBox::warning(nullptr, "借书书目超量", "您最多只能借 12 本书!
    ");
28.         return;
29.     }
30.     else //借书
31.     {
32.         int number;
33.         QList<QTableWidgetItem*> list = ui->BookTable->selectedItems();
34.         if(list.isEmpty())
35.         {
36.             QMessageBox::warning(nullptr, "未选择任何项目", "请选择您要借阅的
    书! ");
37.             return;
38.         }
39.         QString bookbno = list.at(0)->text();
40.         QDate d = QDate::currentDate();
41.         QString date = d.toString("yyyy-MM-dd");
42.
43.         bookQuery.exec("select stock from book where bno = '"
44.                         + bookbno + "'");
45.         if(!bookQuery.isActive())
46.         {
47.             QMessageBox::warning(nullptr, "内部错误", "出现内部错误! ");
48.             return;
49.         }
50.         if(bookQuery.next())
51.         {
52.             number = bookQuery.value(0).toInt();
53.             if(number == 0)
54.             {
55.                 QMessageBox::warning(nullptr, "库存不足", "库存不足,无法借书!
    ");
56.                 return;
57.             }
58.             else
59.             {
60.                 borrowQuery.exec("insert into borrow values ('"
61.                                   + borrower + "','",
62.                                   + bookbno + "','",
63.                                   + username + "','",
64.                                   + date + "','",

```

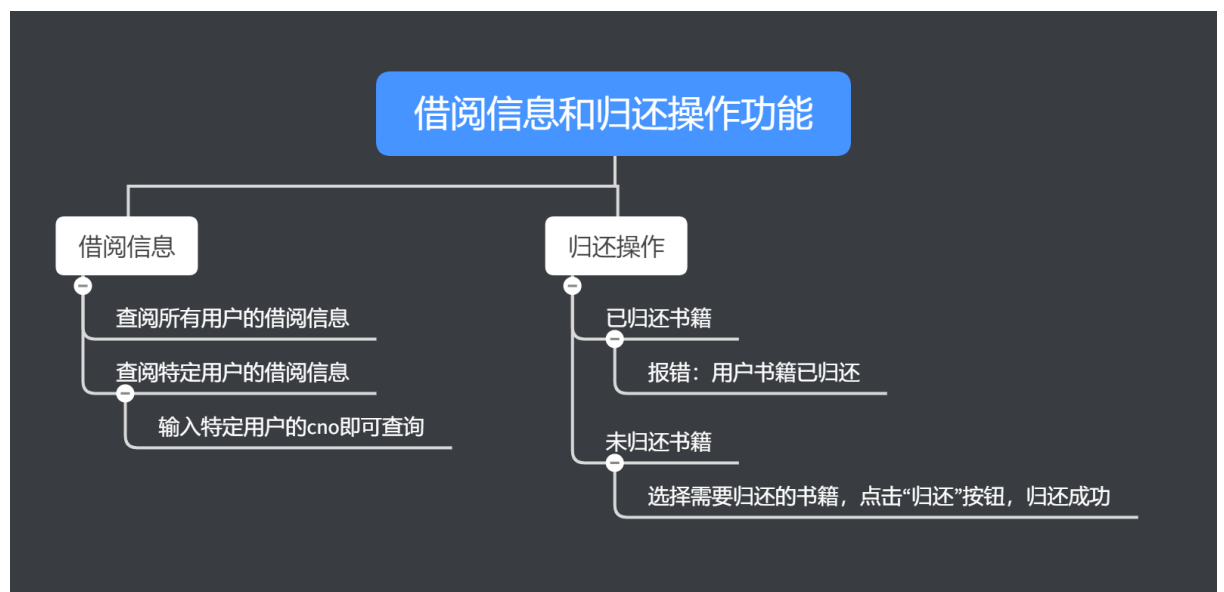
```

65.         + "null)");
66.
67.         if(borrowQuery.isActive())
68.         {
69.             bookQuery.exec("update book set "
70.                             "stock = stock - 1 where bno = '"
71.                             + bookbno + "'");
72.
73.             QMessageBox::information(nullptr, "借书成功", "借书成功!
74.             on_BookSearchPushButton_clicked());
75.         }
76.         else
77.         {
78.             QMessageBox::warning(0, "内部错误", "借书失败!");
79.             return;
80.         }
81.     }
82. }
83. }
84. }

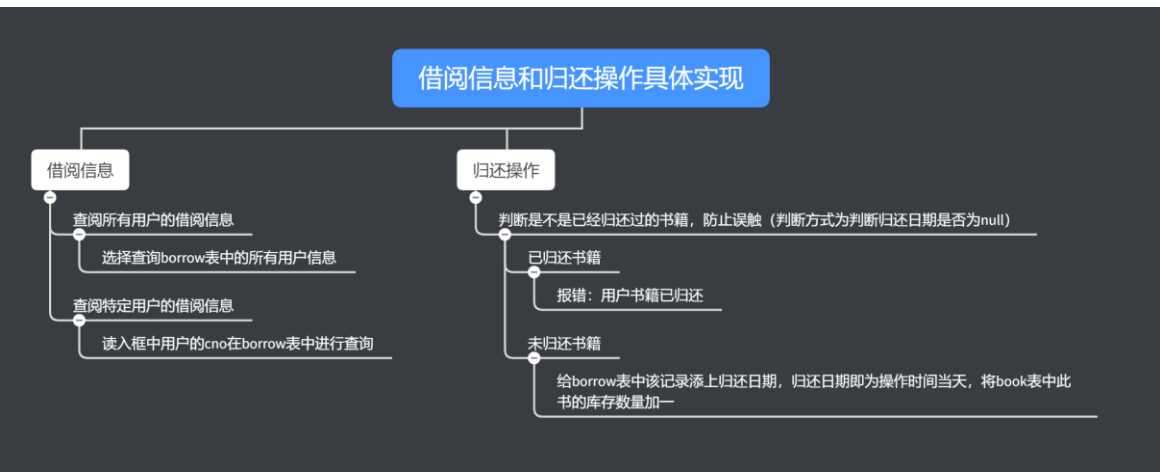
```

3) 借阅信息和归还操作

借阅信息和归还操作功能流程图



借阅信息和归还操作具体实现流程图



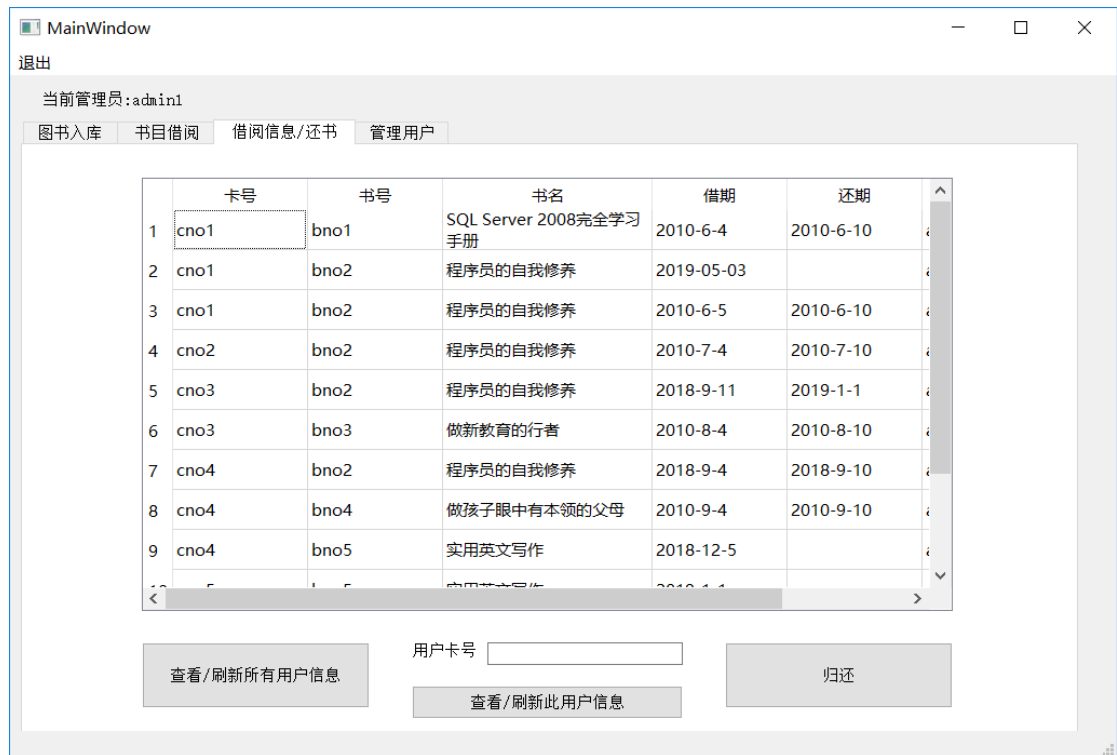
借阅信息和归还操作具体实现：

在查看用户的借阅信息时，如果查阅所有用户的借阅信息，则选择查询 borrow 表中的所有用户信息；如果想要查询特定用户的借阅信息，则读取框中的用户的 cno，以此作为条件在 borrow 表中查询该用户的借阅信息并显示。在归还操作中，首先判断这本书是不是已经归还过了，防止管理员因为误触再归还一遍书籍，判断方式为判断归还日期是否为 null，如果是已归还书籍，则报错显示用户书籍已归还，如果是未归还书籍，则给 borrow 表中该记录添上归还日期，归还日期即为操作时间当天，将 book 表中此书的库存数量加一。

借阅信息和归还操作的界面



查看所有用户的借阅信息：



查看部分用户的借阅信息（以 cno1 为例）：



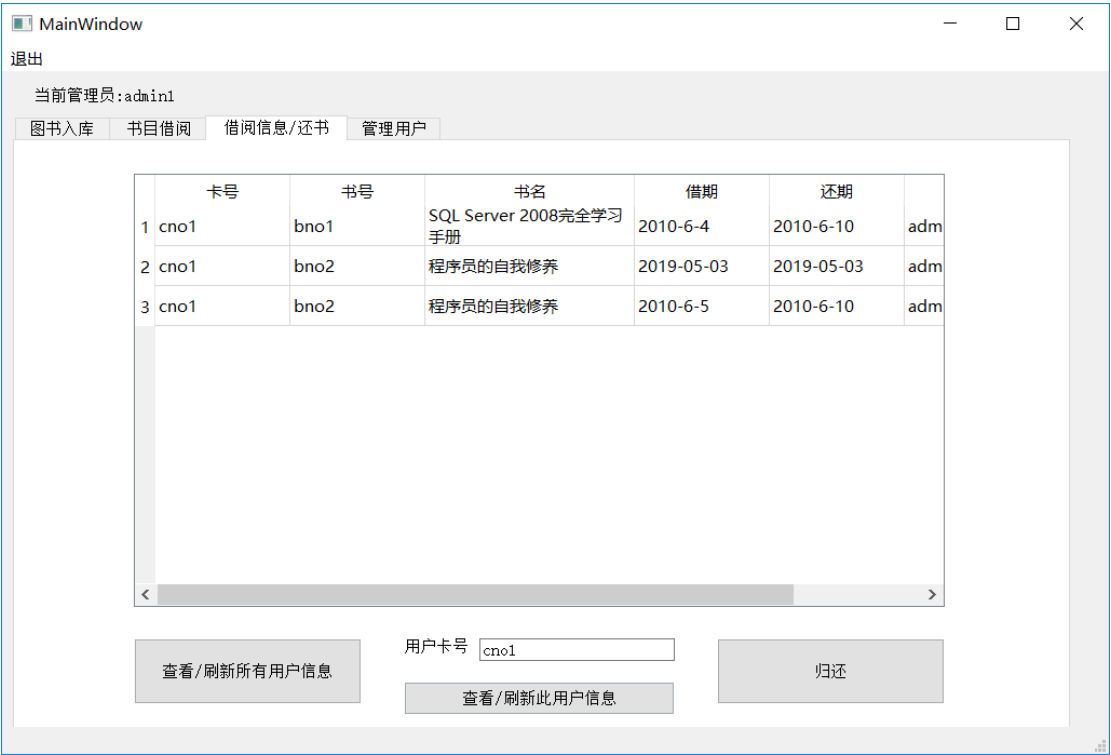
归还 cno1 借书证界面的第一本书



归还 cno1 借书界面的第二本书



归还之后刷新该用户的借阅信息界面：



归还之后的书籍情况（第二行为归还之后的库存，原库存为0）：

	出版年份	作者	价格	总藏书量	目前库存量
1	2001	郭郑州	79.80	6	4
2	2013	俞甲子	65.00	5	1

借阅信息与归还操作的相关代码：

①借阅信息相关代码：

查看所有用户信息：

```
1. borrowQuery.exec("select * from borrow");
2.
3. QString borrowerCno;
4. QString bookbno;
5. QString adminid;
6. QString loanDate;
7. QString returnDate;
8. QString bookTitle;
9.
10. if(!borrowQuery.isActive())
11. {
12.     QMessageBox::warning(nullptr, "内部错误", "查询 LOAN 表错误!");
```



```

13. }
14.
15. int row = 0;
16.
17. ui->BorrowTable->setRowCount(borrowQuery.size());
18.
19. while(borrowQuery.next())
20. {
21.     borrowercno = borrowQuery.value(0).toString();
22.     bookbno = borrowQuery.value(1).toString();
23.     adminid = borrowQuery.value(2).toString();
24.     loandate = borrowQuery.value(3).toString();
25.     returndate = borrowQuery.value(4).toString();
26.
27.     bookQuery.exec("select title from book where bno = '"
28.                   + bookbno + "'");
29.     if(bookQuery.next())
30.         booktitle = bookQuery.value(0).toString();
31.
32.     ui->BorrowTable->setItem(row, 0, new QTableWidgetItem(borrowercno));
33.     ui->BorrowTable->setItem(row, 1, new QTableWidgetItem(bookbno));
34.     ui->BorrowTable->setItem(row, 2, new QTableWidgetItem(booktitle));
35.     ui->BorrowTable->setItem(row, 3, new QTableWidgetItem(loandate));
36.     ui->BorrowTable->setItem(row, 4, new QTableWidgetItem(returndate));
37.     ui->BorrowTable->setItem(row, 5, new QTableWidgetItem(adminid));
38.     row++;
39. }

```

查看特定用户信息：

```

1. QString borrower = ui->BorrowerCno->text();
2. borrowQuery.exec("select cno, bno, ID, borrow_date, return_date from borrow
3.
4.                   "where cno = '" + borrower + "'");
5. QString borrowercno;
6. QString bookbno;
7. QString adminid;
8. QString loandate;
9. QString returndate;
10.
11. if(!borrowQuery.isActive())
12. {
13.     QMessageBox::warning(nullptr, "内部错误", "查询 LOAN 表错误！");

```

```

14. }
15.
16. int row = 0;
17.
18. ui->BorrowTable->setRowCount(borrowQuery.size());
19.
20. while(borrowQuery.next())
21. {
22.     borrowercno = borrowQuery.value(0).toString();
23.     bookbno = borrowQuery.value(1).toString();
24.     adminid = borrowQuery.value(2).toString();
25.     loandate = borrowQuery.value(3).toString();
26.     returndate = borrowQuery.value(4).toString();
27.
28.     bookQuery.exec("select title from book where bno = '"
29.                     + bookbno + "'");
30.     if(bookQuery.next())
31.         booktitle = bookQuery.value(0).toString();
32.
33.     ui->BorrowTable->setItem(row, 0, new QTableWidgetItem(borrowercno));
34.     ui->BorrowTable->setItem(row, 1, new QTableWidgetItem(bookbno));
35.     ui->BorrowTable->setItem(row, 2, new QTableWidgetItem(booktitle));
36.     ui->BorrowTable->setItem(row, 3, new QTableWidgetItem(loandate));
37.     ui->BorrowTable->setItem(row, 4, new QTableWidgetItem(returndate));
38.     ui->BorrowTable->setItem(row, 5, new QTableWidgetItem(adminid));
39.     row++;
40. }

```

②归还操作相关代码：

```

1. QList<QTableWidgetItem*> list = ui->BorrowTable->selectedItems();
2. QString returndate = list.at(4)->text();
3.
4. if(QString::compare(returndate, ""))
5. {
6.     QMessageBox::warning(nullptr, "错误", "用户书目已归还！");
7. }
8. else{
9.     QString borrower = list.at(0)->text();
10.    QString bookbno = list.at(1)->text();
11.    QString adminid = list.at(5)->text();
12.    QDate d = QDate::currentDate();
13.    QString date = d.toString("yyyy-MM-dd");
14.

```

```

15. borrowQuery.exec("update borrow set "
16.                 "return_date = '" + date + "'"
17.                 "where cno = '" + borrower + "' "
18.                 "and bno = '" + bookbno + "' "
19.                 "and ID = '" + adminid + "'");
20.
21. bookQuery.exec("update book set stock = stock + 1"
22.               " where bno = '" + bookbno + "'");
23. if(!borrowQuery.isActive()&&!bookQuery.isActive())
24. {
25.     QMessageBox::warning(nullptr, "内部错误", "归还失败!");
26. }
27. if(bookQuery.isActive()&&borrowQuery.isActive())
28.     QMessageBox::information(nullptr, "归还成功", " 归还成功!");
29. }

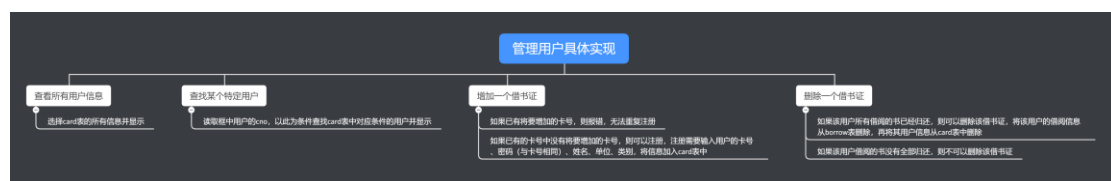
```

4) 管理用户

管理用户功能流程图:



管理用户具体实现流程图:



管理用户具体实现:

当需要查看所有用户信息时, 选择 card 表的所有信息并显示; 当需要查找某个特定用户时, 则读取框中用户的 cno, 以此为条件查找 card 表中对应条件的用户并显示; 如果需要增加一个借书证, 则需要判断该借书证是否存在, 如果已有将要增加的卡号, 则报错, 无法重复注册, 如果已有的卡号中没有将要增加的卡号, 则可以注册, 注册需要输入用户的卡号、密码(与

卡号相同)、姓名、单位、类别,将信息加入 card 表中;当需要删除一个借书证时,如果该用户所有借阅的书已经归还,则可以删除该借书证,将该用户的借阅信息从 borrow 表删除,再将其用户信息从 card 表中删除,如果该用户借阅的书没有全部归还,则不可以删除该借书证。

管理用户的界面:



查看所有用户的信息:



查找某个特定用户

MainWindow

退出

当前管理员:admin1

图书入库

书目借阅

借阅信息/还书

管理用户

	卡号	密码	姓名	单位	类别
1	cno1	cno1	张三	计算机学院	U

增加一个借书证

刷新/查看所有借书证

请输入您所删除的借书证的卡号：

查找此用户

删除该借书证

增加一个借书证

Form

?

×

注意：管理员只能在后台设定，无法注册

CNO

密码

姓名

单位

类别

注册

Form

注意：管理员只能在后台设定，无法注册

CNO cno

密码 cno

姓名 小粒

单位 农学院

类别 U

注册

注册成功

恭喜您，注册成功！

OK

如果是重复注册：

Form

注意：管理员只能在后台设定，无法注册

CNO cno1

密码 cno1

姓名 小粒

单位 计算机学院

类别 U

注册

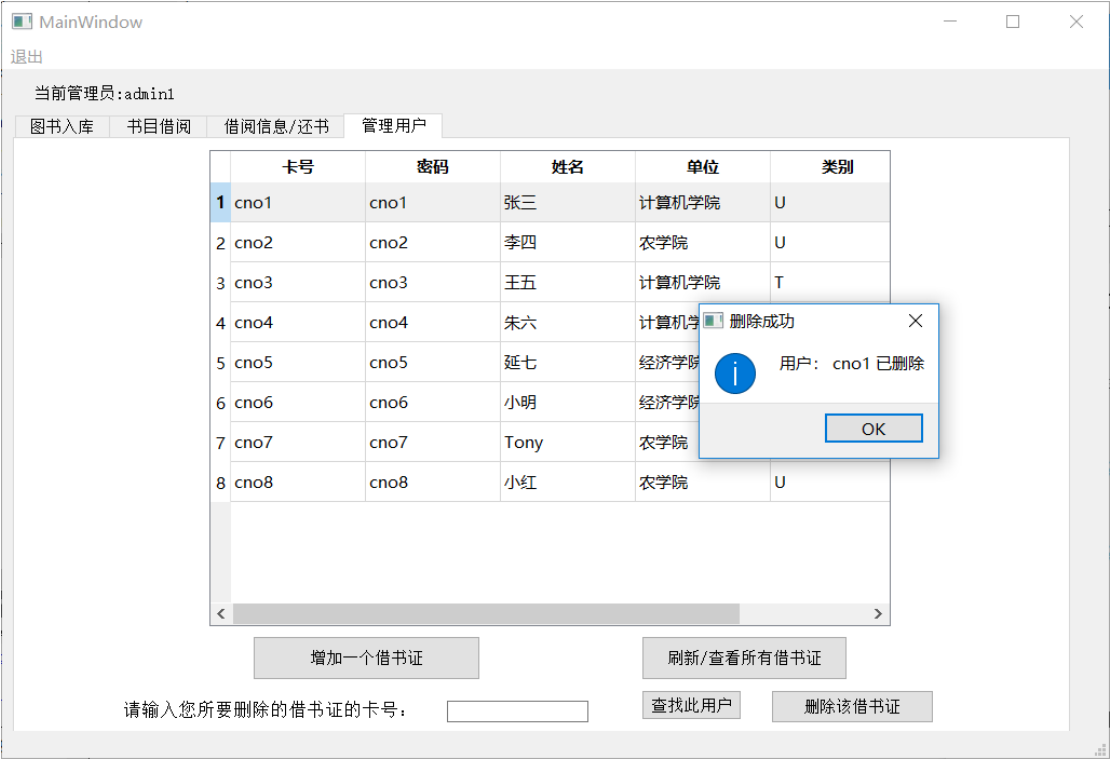
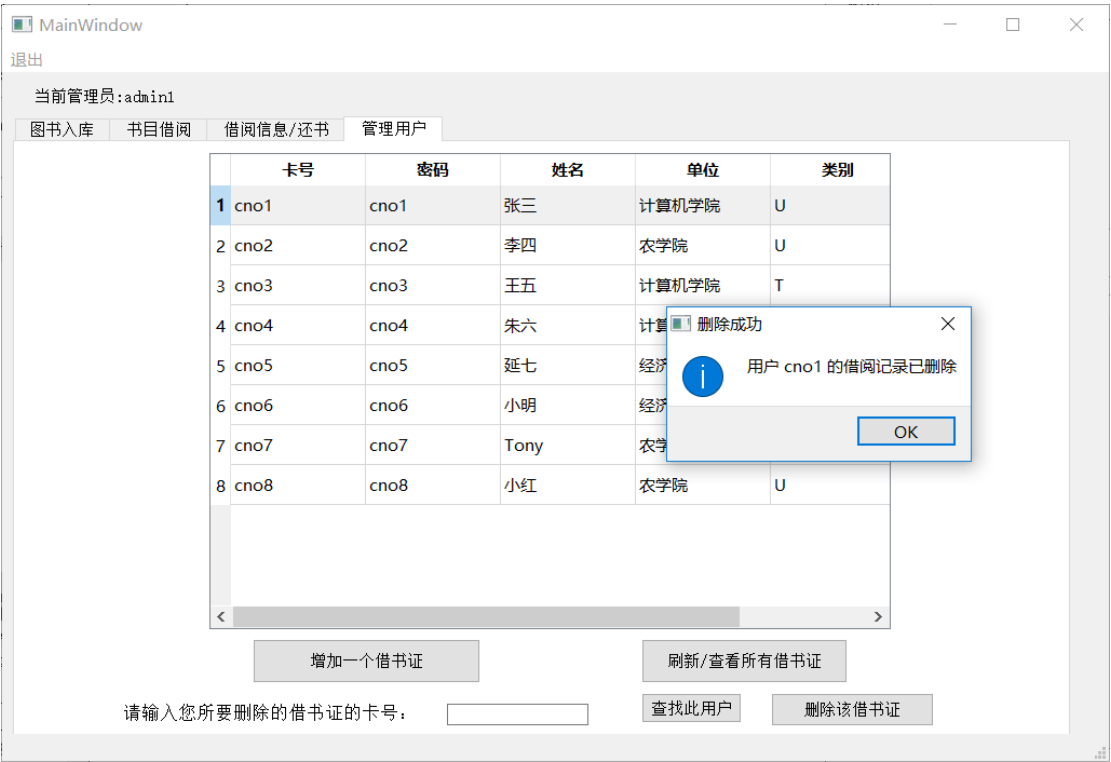
用户已注册

您已用该卡号注册过用户

OK

删除一个借书证：

如果该用户所有书已经归还，比如 cno1:



刷新用户管理界面之后的结果：

MainWindow

退出

当前管理员:admin1

图书入库 书目借阅 借阅信息/还书 管理用户

	卡号	密码	姓名	单位	类别
1	cno2	cno2	李四	农学院	U
2	cno3	cno3	王五	计算机学院	T
3	cno4	cno4	朱六	计算机学院	G
4	cno5	cno5	延七	经济学院	O
5	cno6	cno6	小明	经济学院	G
6	cno7	cno7	Tony	农学院	G
7	cno8	cno8	小红	农学院	U

< >

增加一个借书证 刷新/查看所有借书证

请输入您所删除的借书证的卡号: 查找此用户 删除该借书证

此时借阅信息中 cno1 的借阅信息已被删除：

MainWindow

退出

当前管理员:admin1

图书入库 书目借阅 借阅信息/还书 管理用户

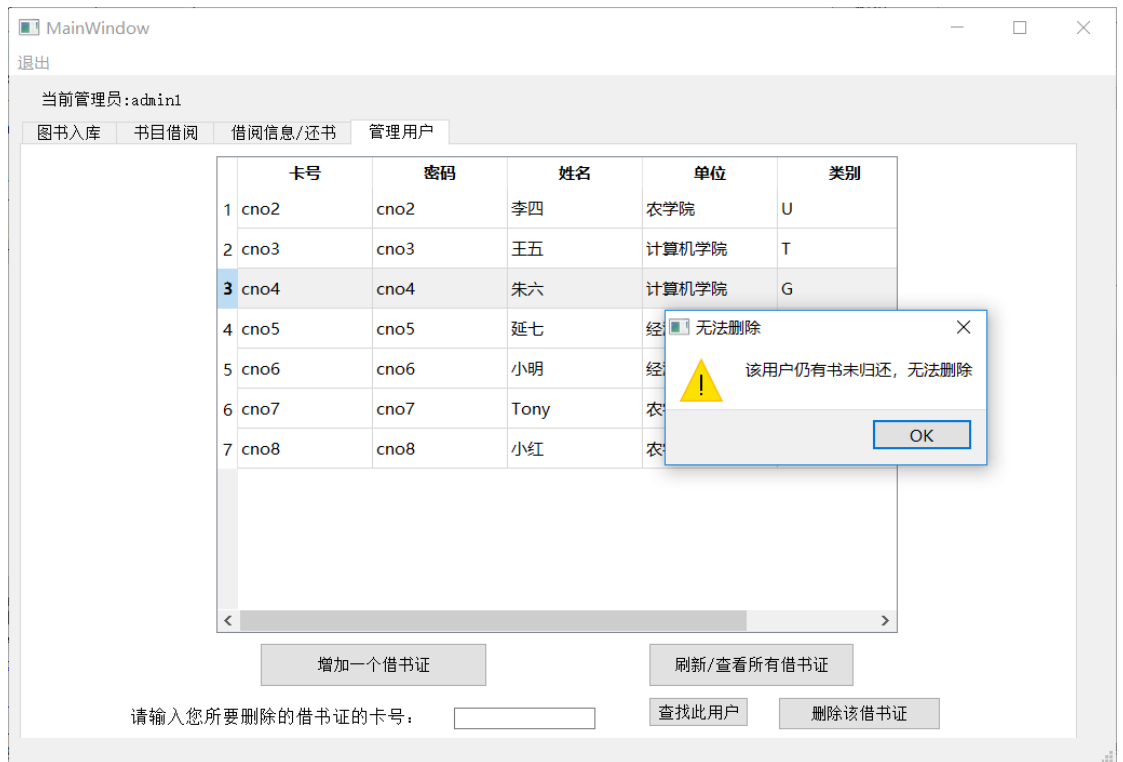
	卡号	书号	书名	借期	还期	经手人
1	cno2	bn02	程序员的自我修养	2010-7-4	2010-7-10	admin3
2	cno3	bn02	程序员的自我修养	2018-9-11	2019-1-1	admin2
3	cno3	bn03	做新教育的行者	2010-8-4	2010-8-10	admin1
4	cno4	bn02	程序员的自我修养	2018-9-4	2018-9-10	admin2
5	cno4	bn04	做孩子眼中有本领的父母	2010-9-4	2010-9-10	admin1
6	cno4	bn05	实用英文写作	2018-12-5		admin1
7	cno5	bn05	实用英文写作	2019-1-1		admin3
8	cno6	bn01	SQL Server 2008完全学习...	2019-3-3		admin3
9	cno6	bn06	数据库系统原理教程	2019-3-4		admin1

< >

查看/刷新所有用户信息 用户卡号 归还

查看/刷新此用户信息

如果该用户并未归还全部书籍（如 cno4）：



管理用户相关代码:

注册一个用户:

```

1. if(ui->RegisterCno->text() == "" || ui->RegisterPassword->text() == ""){
2.     QMessageBox::warning(nullptr, tr("卡号/密码不能为空"), tr("卡号/密码不能为
   空"));
3. }
4. else{
5.     if(ui->RegisterPassword->text().length() > 16 || ui->RegisterPassword->tex
      t().length() < 2){
6.         QMessageBox::warning(nullptr, tr("密码长度问题"), tr("密码长度必须在
      2-16 位之间"));
7.     }
8.     else{
9.         cardQuery.exec("select cno from card where cno = '"
10.             + ui->RegisterCno->text() + "'");
11.         if(cardQuery.next()){
12.             QMessageBox::warning(nullptr, tr("用户已注册"), tr("您已用该卡号注
              册过用户"));
13.         }
14.         else{
15.             cardQuery.exec("insert into card values('" +
16.                 ui->RegisterCno->text() + "','")

```

```

17.         + ui->RegisterPassword->text() + "',''"
18.         + ui->RegisterName->text() + "',''"
19.         + ui->RegisterDepartment->text() + "',''"
20.         + ui->RegisterType->text()+ "','");
21.         if(cardQuery.isActive()){
22.             QMessageBox::about(nullptr,tr("注册成功"),tr("恭喜您，注册成功！
    "));
23.             close();
24.         }
25.         else{
26.             QMessageBox::about(nullptr,tr("注册失败"),tr("注册失败"));
27.         }
28.     }
29. }
30. }

```

删除一个用户：

```

1. QList<QTableWidgetItem*> list = ui->UserTable->selectedItems();
2. QString usercno = list.at(0)->text();
3. borrowQuery.exec("select cno from borrow "
4.                 "where cno = '" + usercno + "' "
5.                 "and return_date is null");
6. if(!borrowQuery.isActive())
7. {
8.     QMessageBox::warning(nullptr, "内部错误", "查询错误！");
9. }
10. if(borrowQuery.next())
11. {
12.     QMessageBox::warning(nullptr, "无法删除", "该用户仍有书未归还，无法删除");
13. }
14. else{
15.     borrowQuery.exec("delete from borrow where cno = '" + usercno + "'");
16.     if(borrowQuery.isActive()){
17.         QMessageBox::information(nullptr, "删除成功", "用
        户 " + usercno + " 的借阅记录已删除");
18.     }
19.     else{
20.         QMessageBox::warning(nullptr, "删除失败", "用户 " + usercno + " 的借阅
        记录删除失败");
21.     }
22.     cardQuery.exec("delete from card where cno = '" + usercno + "'");
23.     if(cardQuery.isActive()){

```

```

24.         QMessageBox::information(nullptr, "删除成功", "用
    户: " + usercno + " 已删除");
25.     }
26.     else{
27.         QMessageBox::warning(nullptr, "删除失败", "用户: " + usercno + " 删除
    失败");
28.     }
29. }

```

(4) 普通用户登录

普通用户只能进行图书的查找，其中可以进行复合查找

图书查找的功能流程图：



图书查找的具体实现流程图：



图书查找的具体实现：

当要查询所有书籍时，则选择 book 表中所有的记录并显示，当要查询特定书籍时，读入框中各个不同的要求，并将这些要求作为条件在 book 表中进行复合查询（其中书名、出版社、作者可以进行字段查询）。

图书查找的界面

普通用户

退出

当前用户: cno2

类别

书名

出版社

出版年份

作者

价格

搜索

查询所有书目

刷新

借书人

借阅

书号	类别	书名	出版社	出版年份
----	----	----	-----	------

查询所有书目

普通用户

退出

当前用户: cno2

类别

书名

出版社

出版年份

作者

价格

搜索

查询所有书目

刷新

借书人

借阅

	书号	类别	书名	出版社
1	bn01	计算机	SQL Server 2008完全学习手册	清华大学出版社
2	bn06	计算机	数据库系统原理教程	清华大学出版社
3	bn05	英语	实用英文写作	高等教育出版社
4	bn03	教育	做新教育的行者	福建教育出版社
5	bn02	计算机	程序员的自我修养	电子工业出版社
6	bn04	教育	做孩子眼中有本领的父母	电子工业出版社

查询特定书目

单独查询类别：

普通用户

退出

当前用户: cno2

类别

计算机

书名

出版社

出版年份

0

3000

作者

价格

0.00

99.99

搜索

查询所有书目

刷新

借书人

借阅

	书号	类别	书名	出版社
1	bno1	计算机	SQL Server 2008完全学习手册	清华大学出版社
2	bno2	计算机	程序员的自我修养	电子工业出版社
3	bno6	计算机	数据库系统原理教程	清华大学出版社

单独查询书名：

普通用户

退出

当前用户: cno2

类别

书名孩子

出版社

出版年份

0

3000

作者

价格

0.00

99.99

搜索

查询所有书目

刷新

借书人

借阅

	书号	类别	书名	出版社
1	bno4	教育	做孩子眼中有本领的父母	电子工业出版社

单独查询出版社

普通用户

退出

当前用户: cno2

类别

书名

出版社

出版年份

0

3000

作者

价格

0.00

99.99

搜索 查询所有书目 刷新

借书人 借阅

	书号	类别	书名	出版社
1	bn01	计算机	SQL Server 2008完全学习手册	清华大学出版社
2	bn06	计算机	数据库系统原理教程	清华大学出版社

单独查询出版年份

普通用户

退出

当前用户: cno2

类别

书名

出版社

出版年份

1990

2003

作者

价格

0.00

99.99

搜索 查询所有书目 刷新

借书人 借阅

	书号	类别	书名	出版社
1	bn01	计算机	SQL Server 2008完全学习手册	清华大学出版社
2	bn03	教育	做新教育的行者	福建教育出版社
3	bn06	计算机	数据库系统原理教程	清华大学出版社

单独查询作者

普通用户

退出

当前用户: cno2

类别

书名

出版社

出版年份

0

3000

作者

云

价格

0.00

99.99

搜索

查询所有书目

刷新

借书人

借阅

	书号	类别	书名	出版社
1	bn03	教育	做新教育的行者	福建教育出版社
2	bn04	教育	做孩子眼中有本领的父母	电子工业出版社

单独查询价格

普通用户

退出

当前用户: cno2

类别

书名

出版社

出版年份

0

3000

作者

价格

20.00

25.00

搜索

查询所有书目

刷新

借书人

借阅

	书号	类别	书名	出版社
1	bn03	教育	做新教育的行者	福建教育出版社
2	bn04	教育	做孩子眼中有本领的父母	电子工业出版社

复合查询

普通用户

退出

当前用户: cno2

类别

书名

出版社

出版年份 ----

作者

价格 ----

借书人

	出版社	出版年份	作者	价格	总藏书量
1	清华大学出版社	2001	郭郑州	79.80	6

图书查找代码:

```
1. int condition = 0; //有无已知条件
2.
3. //复合查询条件
4. QString bookcategory = ui->BookCategory->currentText();
5.
6. QString booktitle = ui->BookTitle->text();
7.
8. QString bookpress = ui->BookPress->text();
9.
10. QString upperyear = ui->BookUpperYear->text();
11. QString loweryear = ui->BookLowerYear->text();
12. int lower_year = ui->BookLowerYear->value();
13. int upper_year = ui->BookUpperYear->value();
14.
15. QString bookauthor = ui->BookAuthor->text();
16.
17. QString upperprice = ui->BookUpperPrice->text();
18. QString lowerprice = ui->BookLowerPrice->text();
19. QString query = "select * from book ";
20.
```



```
21. //开始查询
22. if(lower_year>upper_year && lower_year>0 && upper_year>0){
23.     QMessageBox::warning(nullptr, tr("年份错误"), tr("年份输入有误"));
24.     return;
25. }
26. if(lowerprice.toDouble()>upperprice.toDouble() && lowerprice.toDouble()>0 &
    & upperprice.toDouble()>0){
27.     QMessageBox::warning(nullptr, tr("价格错误"), tr("价格输入有误"));
28.     return;
29. }
30.
31.
32. if(QString::compare(bookcategory, ""))
33. {
34.     query.append("where category = '" + bookcategory + "' ");
35.     condition = 1;
36. }
37.
38. if(QString::compare(booktitle, ""))
39. {
40.     if(condition == 0){
41.         query.append("where title like '%" + booktitle + "%' ");//涵盖其中字
            眼即可，不需要书名和书一一对应
42.         condition = 1;
43.     }
44.     else
45.         query.append("and title like '%" + booktitle + "%' ");
46. }
47.
48. if(QString::compare(bookpress, ""))
49. {
50.     if(condition == 0)
51.     {
52.         query.append("where press like '%" + bookpress + "%' ");
53.         condition = 1;
54.     }
55.     else
56.         query.append("and press like '%" + bookpress + "%' ");
57. }
58.
59. if(QString::compare(bookauthor, ""))
60. {
61.     if(condition == 0)
62.     {
```

```

63.         query.append("where author like '%" + bookauthor + "%' ");
64.         condition = 1;
65.     }
66.     else
67.         query.append("and author like '%" + bookauthor + "%' ");
68. }
69.
70.
71. if(lower_year >= 0 && upper_year >= 0 && lower_year <= upper_year)
72. {
73.     if(condition == 0)
74.     {
75.         query.append("where year <= '" + upperyear +
76.             "' and year >= '" + loweryear + "'");
77.         condition = 1;
78.     }
79.     else
80.         query.append("and year <= '" + upperyear +
81.             "' and year >= '" + loweryear + "'");
82. }
83.
84. if(lowerprice.toDouble() >= 0 && upperprice.toDouble() >= 0 && lowerprice.toD
    ouble() <= lowerprice.toDouble())
85. {
86.     if(condition == 0)
87.     {
88.         query.append("where price between '" + upperprice +
89.             "' and '" + lowerprice + "'");
90.         condition = 1;
91.     }
92.     else
93.         query.append("and price <= '" + upperprice +
94.             "' and price >= '" + lowerprice + "'");
95. }
96.
97. QSqlQuery temp = QSqlQuery(db);
98. ui->BookTable->clearContents();
99. bookQuery.exec(query);
100.
101. int row = 0;
102. ui->BookTable->setRowCount(bookQuery.size());
103. while(bookQuery.next())
104. {
105.     temp.exec("select * from book where bno = '"

```

```

106.     + bookQuery.value(0).toString() + "'");
107.     temp.next();
108.     for(int i = 0; i < 9; i++)
109.     {
110.         if(i == 6){
111.             float b=temp.value(i).toFloat();
112.             ui->BookTable->setItem(row, i, new QTableWidgetItem(QString::nu
mber(b,'f',2)));
113.         }
114.         else{
115.             ui->BookTable->setItem(row, i, new QTableWidgetItem(temp.value(
i).toString()));
116.         }
117.     }
118.
119.     row++;
120. }
121. ui->BookTable->show();

```

四、 实验感想

通过本次实验，我学会了通过 QT 连接 mysql 数据库的一些操作，做出了一个简易的图书管理系统，在这期间我遇到了很多困难，比如刚开始的如何连接数据库，后面如何使价格的两位小数得到显示，批量导入时如何读入 txt 文件以及如何从 txt 文件中获取 bno 及库存信息，如何在管理员界面帮助用户借书以及一些没有考虑到的情况等问题，最终均通过自己的查询搜索和询问同学得到了解决。通过本次实验我对数据库系统有了更深刻的理解，自己解决问题的能力也得到了一定的提高，能做出这个简易的图书管理系统是很令人有成就感的事情，感觉很有收获也很有意思，以后会再接再厉，不断寻找问题并解决问题。