

实验 5 数据库程序设计

一、实验目的

- 1、设计并实现一个精简的图书管理系统，具有入库、查询、借书、还书、借书证管理等基本功能。
- 2、通过本次设计来加深对数据库的了解和使用，同时提高自身的系统编程能力。

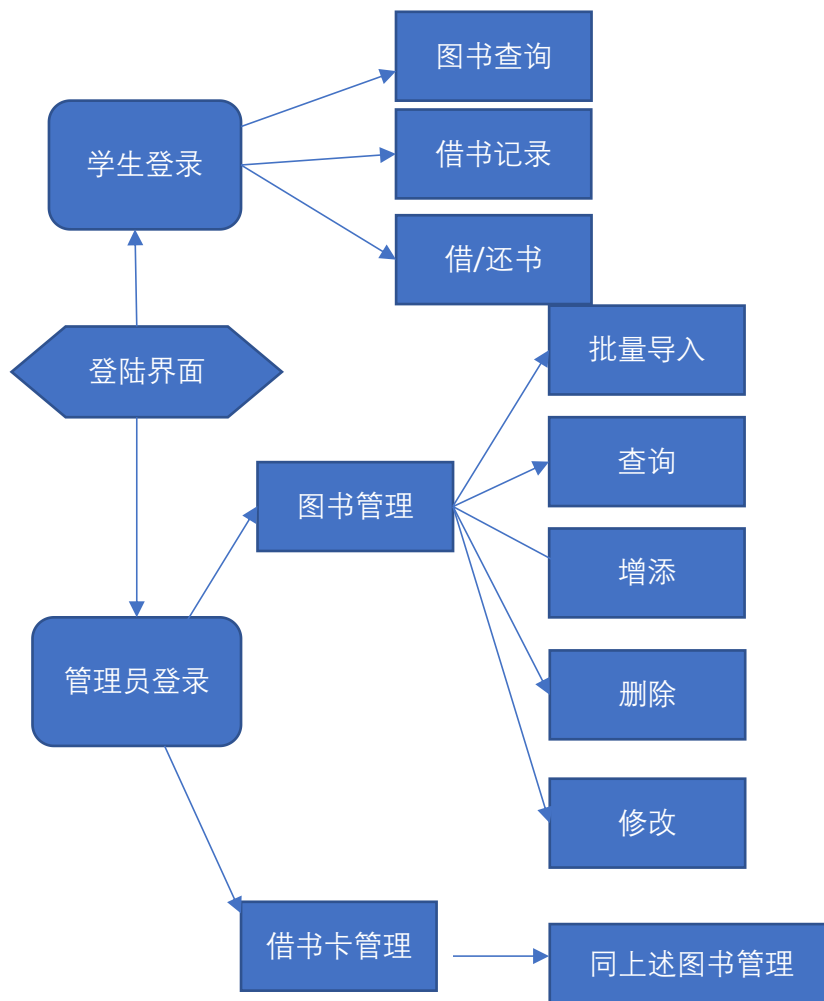
二、实验平台

开发工具：Qt Creator；数据库平台：Mysql 和 ODBC

三、总体设计

1、系统架构描述

本系统主要包括管理员登录、图书入库、图书查询、借书管理、还书管理、借书证管理六大功能模块。系统处理基本流程如下：



2、数据库表设计
图书信息表(book)

模块名称	功能描述
管理员登陆	根据管理员ID 和密码登录系统
图书管理	1. 包含增删查改以及批量导入的功能 2. 在界面上可以看到目前书库的所有信息
借书卡管理	1. 包含增删查改以及批量导入的功能 2. 在界面上可以看到目前所有借书卡的信息
学生登录	根据学生ID 和密码登录系统
借/还书	1. 输入书号，点击借书，如果该书还有库存以及该同学还有借书余量则借书成功 否则失败 2. 输入书号，点击还书，如果有借书的记录则还书成功，否则报错
借书记录	点击“显示记录”按钮即可查询自己的所有的借书记录
图书查询	同学也可以根据书名对书的信息进行模糊查询

字段名	数据类型	是否为主键	说明
bno	Char(8)	是	书序列号
title	varchar(40)		书名
author	varchar(20)		作者
price	decimal(7,2)		价格
total	int		总存书量
stock	Int		剩余存量
category	varchar(10)		种类
press	varchar(20)		出版社

借书证(card)

字段名	数据类型	是否为主键	说明
Cno	char(10)	是	卡号
Name	varchar(20)		用户姓名
Department	varchar(40)		所属部门
Type	Char(1)		用户身份(U,G,T,O)
Available_book	Int		可用借书量

登录信息(Identification)

字段名	数据类型	是否为主键	说明
Cno	Char(10)	是	用户卡号
Password	Varchar(40)		用户密码

借书记录(borrow)

字段名	数据类型	是否为主键	说明
Cno	Char(10)	否（为外键）	借书卡号
Bno	Char(10)	否（为外键）	借书书号

Borrowed_date	Date		借书日期
Return_date	Date		还书日期

3、所用开发技术

本次开发所用技术可以从3个方面来说明：MySQL，QT，ODBC

QT

Qt 是一个跨平台的C++应用程序开发框架。广泛用于开发GUI程序，这种情况下又被称为部件工具箱。也可用于开发非GUI程序，例如控制台工具和服务器。

- 它是基于 C++ 的一种语言扩展(Extention)。Qt 的类都是用 C++ 写出来的。这也就是说, Qt 本身已经继承了 C++的快速、简易、面向对象等许多的优点。
- 有基于XML的界面设计器Qt Designer，一款便捷的解码设计，可以说是C++领域使用最方便的界面设计器，在设计界面进行可视化设计后会自行在后台进行界面的代码编写，通过ui和.h 以及moc文件进行前后端连接。
- 还有很多完整的界面库函数以及与sql的接口函数，都有对应签名和使用说明在qt助手里，在开发过程中可以很自如的针对开发需求来使用。

MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的

RDBMS(Relational Database Management System: 关系数据库管理系统)应用软件之一。MySQL 在过去由于性能高、成本低、可靠性好，已经成为最流行的开源数据库，因此被广泛地应用在 Internet 上的中小型网站中。随着MySQL的不断成熟，它也逐渐用于更多大规模网站和应用，比如维基百科、Google和Facebook等网站。因为本次实验的涉及数据量较小，以及为了方便熟悉 sql语句等因素，MySQL称为了最佳选择

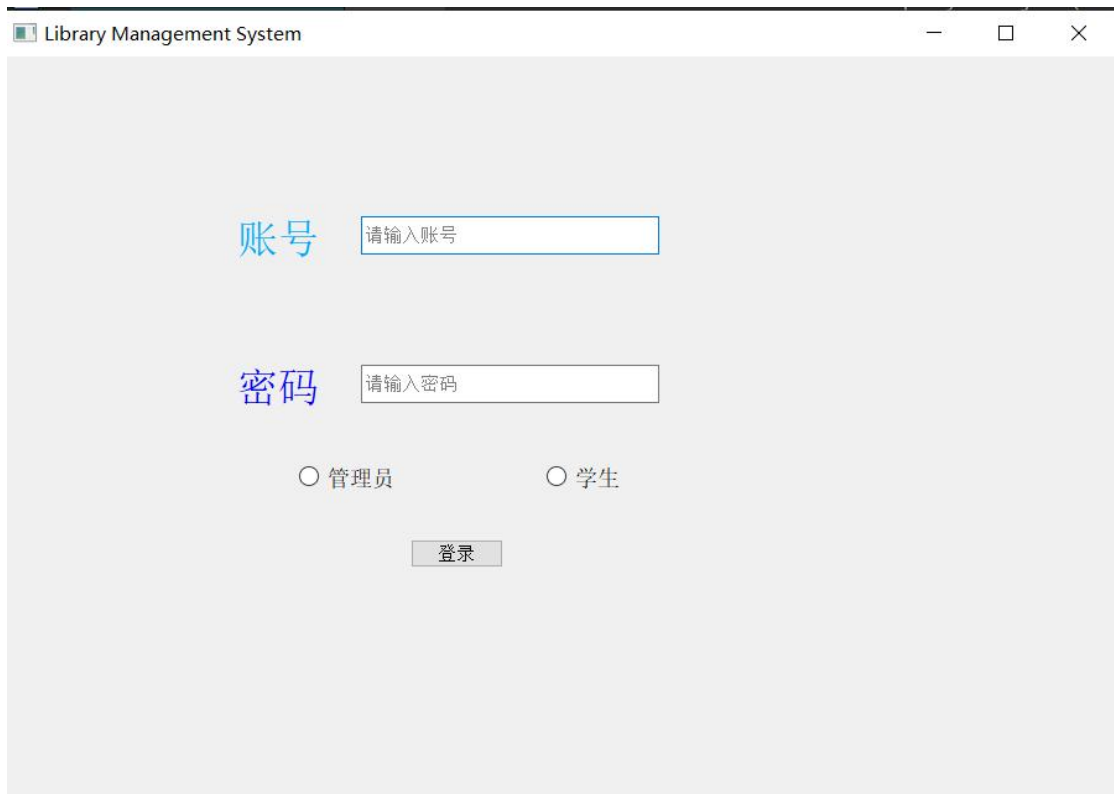
ODBC

ODBC (Open Database Connectivity, 开放数据库互连) 提供了一种标准的API (应用程序编程接口) 方法来访问数据库管理系统 (DBMS)。这些API利用SQL来完成其大部分任务。ODBC本身也提供了对SQL语言的支持，用户可以直接将SQL语句送给ODBC。ODBC的设计者们努力使它具有最大的独立性和开放性。

在windows中自带了ODBC(64bit)，通过ODBC实现与mysql连接，并作为中间件与我们编写的工程文件进行交互。

四、详细设计

1. 登录界面设计



图表 1 登陆界面

登录界面如图所示，有2个选项，对应2种不同界面，而管理员和学生的登录信息（账号（cno）和密码（password））（存储在 identification table中）库也对应不同：学生的账号密码只能登录进入学生的界面，而管理员的账号密码二者都能登入。

在选定某一个登录选项后，系统将根据选择，找到对应账号的密码，查看是否与输入的匹配，如果匹配成功，即可成功登录。

如下为学生登录对应的代码（管理员的类似）。

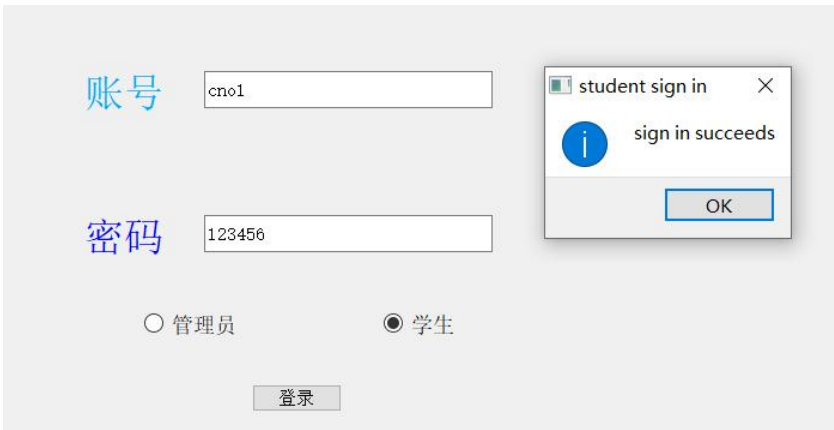
```
QString userInputNumber = ui->InputNumber->text();           // 从账号输入框组件中，
// 获取输入的信息，以字符串的形式返回给变量 userInputNumber
QString userInputPassWord = ui->InputPassWord->text();
if(ui->radioButton_2->isChecked()){
    // 获取数据库中的信息

    QSqlQuery sql;
    sql.prepare("select * from identification where cno = :userInputNumber");
    sql.bindValue(":userInputNumber",userInputNumber);
    sql.exec();

    QString userPassWord;
    while(sql.next())
    {
        userNumber = sql.value("cno").toString();
        userPassWord = sql.value("password").toString();
    }
}
```

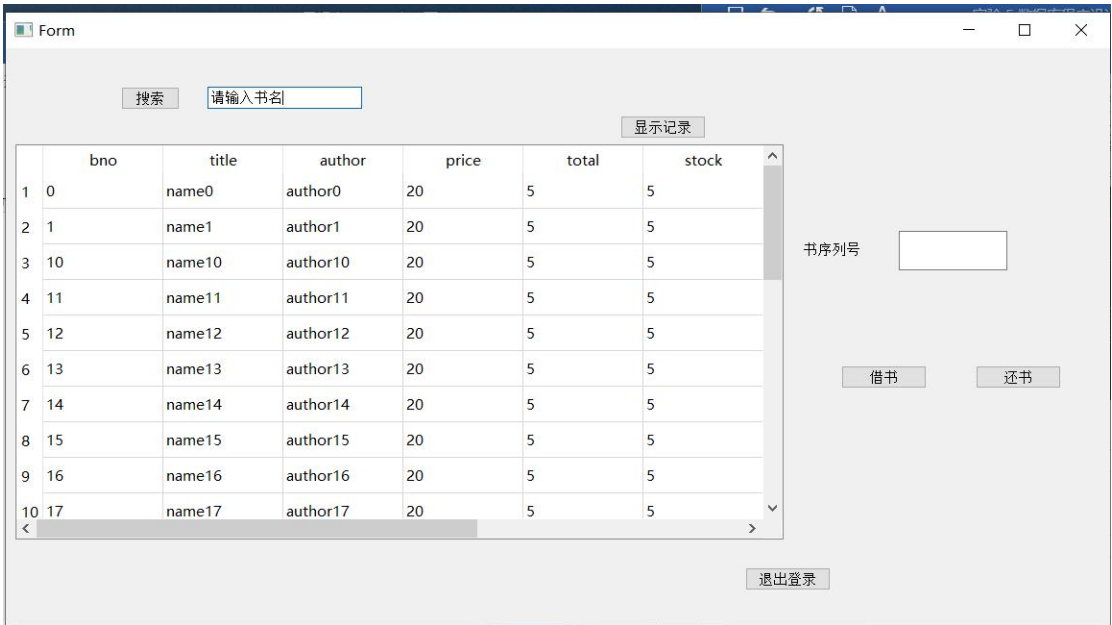
```
qDebug()<<"数据库找到的账号:"+userNumber;
QDebug()<<"数据库找到的密码:"+userPassWord;
sql.clear();

if(userInputPassWord == userPassWord&&userInputNumber==userNumber)
{
    QMessageBox::information(this,"student sign in","sign in succeeds");
    this->~SignIn_Page();
    stupage* w = new stupage();
    w->show();
}
```



图表 2 学生登录成功界面

2. 学生界面



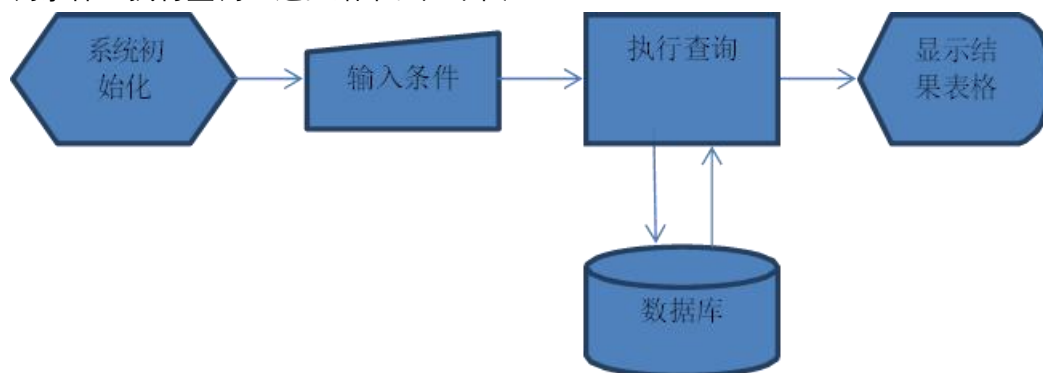
图表 3 学生界面

如图所示，学生界面包括了三大功能：查书，显示借书记录，借/还书。具体功能解析如下：

1. 查询书籍

图书查询在系统初始化完成后可直接开始使用，其基本工作流程是：输入查询条件→组合查

询条件→执行查询→返回结果。如下图：



查询的条件有书名。查询的方式为模糊查找，即可带有“%”符号进行前缀后缀或者前后缀的模糊查找。

实例：搜索以“SQL”开头的书籍

	bno	title	author	price	total	stock	
1	bn01	SQL Server 完...	郭郑州	79.8	5	4	计算

结果如上图所示

具体实现代码如下

```
void stupage::on_btmsch_clicked() //search
{
    if(ui->boxSch->text()==""){
        stupage::setTableInfo();
    }
    else{
        QSqlQuery sql;
        sql.prepare("select * from book where title like :title");
        sql.bindValue(":title",ui->boxSch->text());
        sql.exec();
        qDebug() << ui->boxSch->text();
        this->dataTableModel->clear();
        int row=0;

        this->dataTableModel->setHorizontalHeaderItem(0,new QTableWidgetItem("bno"));
        this->dataTableModel->setHorizontalHeaderItem(1,new QTableWidgetItem("title"));
        this->dataTableModel->setHorizontalHeaderItem(2,new QTableWidgetItem("author"));
        this->dataTableModel->setHorizontalHeaderItem(3,new QTableWidgetItem("price"));
        this->dataTableModel->setHorizontalHeaderItem(4,new QTableWidgetItem("total"));
        this->dataTableModel->setHorizontalHeaderItem(5,new QTableWidgetItem("stock"));
```

```

        this->dataTableModel->setHorizontalHeaderItem(6,new
QStandardItem("category"));
        this->dataTableModel->setHorizontalHeaderItem(7,new QStandardItem("press"));
        //设置显示的列头信息
        while(sql.next())
        {
            this->dataTableModel->setItem(row,0,new QStandardItem (sql.value("bno").
toString()));
            this->dataTableModel->setItem(row,1,new QStandardItem (sql.value("title").
toString()));
            this->dataTableModel->setItem(row,2,new QStandardItem (sql.value("author").
toString()));
            this->dataTableModel->setItem(row,3,new QStandardItem (sql.value("price").
toString()));
            this->dataTableModel->setItem(row,4,new QStandardItem (sql.value("total").
toString()));
            this->dataTableModel->setItem(row,5,new QStandardItem (sql.value("stock").
toString()));
            this->dataTableModel->setItem(row,6,new QStandardItem (sql.value ("category").
toString()));
            this->dataTableModel->setItem(row,7,new QStandardItem (sql.value("press").
toString()));
            row++;
        }
        //打印搜索信息
        sql.clear();
    }
}

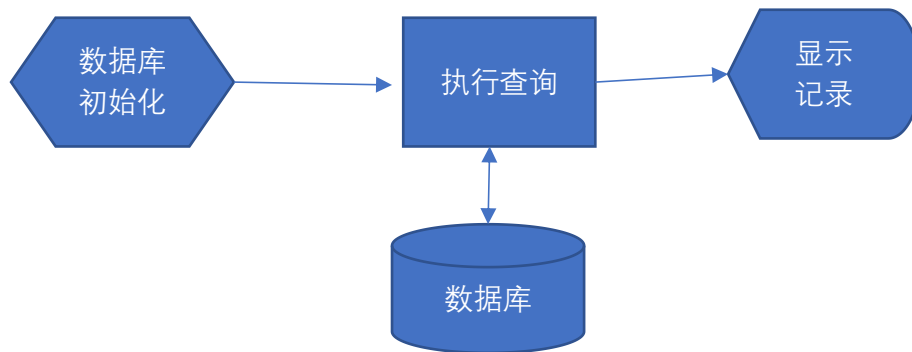
```

2. 显示借书记录

点击“显示记录”按钮后，在用户界面上，则会显示出对应用户账号（cno）的借/还书记录，包括cno，bno，书名和借还书日期，在顶端还显示着该借书卡对应的可用借余额

搜索		SQL%					显示记录
	cno	bno	title	borrowed date	return date	available book	
1	cno1	bno1	SQL Server 完...	2010-06-05	2022-05-06	5	
2	cno1	bno2	程序员的自我修...	2010-06-04	2022-05-06		
3	cno1	bno2	程序员的自我修...	2022-05-06	2022-05-06		

对应的后台流程是：数据库初始化->执行查询->返回记录



实现代码如下

```
void stupage::showhistory(){//show borrow info
    //设置表头
    QString userno ="cno1";
    //QString userno = userNumber;

    this->dataTableModel->setHorizontalHeaderItem(0,new
    QStandardItem(QObject::tr("cno")));
    this->dataTableModel->setHorizontalHeaderItem(1,new
    QStandardItem(QObject::tr("bno")));
    this->dataTableModel->setHorizontalHeaderItem(2,new
    QStandardItem(QObject::tr("title")));
    this->dataTableModel->setHorizontalHeaderItem(3,new
    QStandardItem(QObject::tr("borrowed_date")));
    this->dataTableModel->setHorizontalHeaderItem(4,new
    QStandardItem(QObject::tr("return _date")));
    this->dataTableModel->setHorizontalHeaderItem(5,new
    QStandardItem(QObject::tr("available_book")));
    QSqlQuery sql;
    sql.prepare("select cno, bno, title, borrowed_date, return_date ,available_book
    from book natural join borrow natural join card where cno = :userno ");
    sql.bindValue(":userno",userno);
    sql.exec();

    int row = 0;    // 定义变量 row 表示表格行数
    while(sql.next())
    {
        this->dataTableModel->setItem(row,0,new QStandardItem
        (sql.value("cno").toString()));
```



```

        this->dataTableModel->setItem(row,1,new QStandardItem
(sql.value("bno").toString()));
        this->dataTableModel->setItem(row,2,new QStandardItem
(sql.value("title").toString()));
        this->dataTableModel->setItem(row,3,new QStandardItem
(sql.value("borrowed_date").toString()));
        this->dataTableModel->setItem(row,4,new QStandardItem
(sql.value("return_date").toString()));
        if(!row)this->dataTableModel->setItem(row,5,new QStandardItem
(sql.value("available_book").toString()));

        row++;
    }
    sql.clear();

    ui->bookTable->setEditTriggers(QAbstractItemView::NoEditTriggers);
    ui->bookTable->setModel(this->dataTableModel);
}

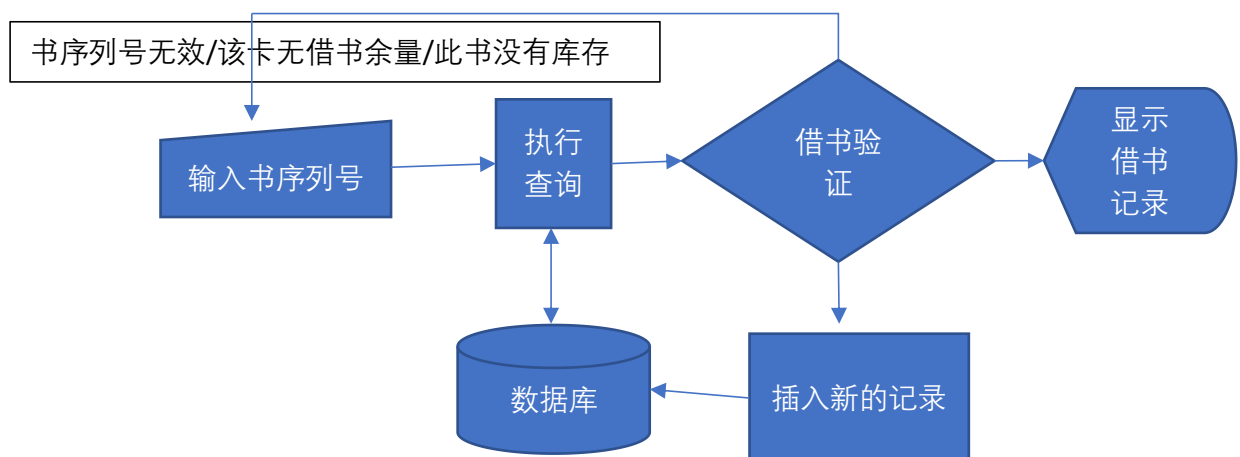
```

3. 借还书

借还书为学生界面的核心功能板块，以下分为借书，还书2个板块进行详细阐述。

● 借书

在“书序列号”框内输入对应bno，点击借书按钮后，后台会根据①书序列号的有效与否②书的可用余量③借书卡的可用借书余量 来进行判别是否借书成功。如若借书成功则会在 borrow（借书记录表）中插入一条新的信息，记录着bno cno 书名 借书日期
具体工作流程如下：



显示记录						
	cno	bno	title	borrowed date	return date	available book
1	cno1	bno1	SQL Server 完...	2010-06-05	2010-07-10	5
2	cno1	bno2	程序员的自我修...	2010-06-04	2010-07-10	
3	cno1	bno2	程序员的自我修...	2022-05-06	2010-07-10	

如图是未进行任何处理之前的界面

显示记录						
	cno	bno	title	borrowed date	return date	available book
1	cno1	bno1	SQL Server 完...	2010-06-05	2010-07-10	4
2	cno1	bno2	程序员的自我修...	2010-06-04	2010-07-10	
3	cno1	bno2	程序员的自我修...	2022-05-06	2010-07-10	
4	cno1	bno1	SQL Server 完...	2022-05-28		

如图是在输入“bno2”并点击借书后的画面，可看到available_book 减少了1，且记录中增添了一个新的借书记录

具体代码如下：

```
void stupage::on_borrowbtn_clicked()//借书
{
    if(ui->bnoBox->text().size()){
        QDate curDate = QDate::currentDate();
        //QString userno = userNumber;
        QString userno = "cno1";
        QSqlQuery sql;
        //check if user have available books to borrow;

        sql.prepare("select available_book from card where cno = :userno");
        sql.bindValue(":userno",userno);
        sql.exec();
        int curAvBook = sql.value("available_book").toInt();
        qDebug() << userno;
        qDebug() << curAvBook;
        sql.clear();

        //check if the book still in library;
        sql.prepare("select stock from book where bno = :bno");
        sql.bindValue(":bno",ui->bnoBox->text());
        sql.exec();
```

```

int curBookStock;
curBookStock = sql.value("stock").toInt();
qDebug() << ui->bnoBox->text();
qDebug()<< sql.value("stock");
sql.clear();

//judge
if(curAvBook == 0 || curBookStock <= 0 || curBookStock == NULL){
    QMessageBox::warning(this,"borrow","borrow failed");
}
else{
    //insert into borrow table
    sql.prepare("insert into borrow(cno,bno,borrowed_date)
values(:userid,:bno,:curDate)");
    sql.bindValue(":userid",userid);
    sql.bindValue(":bno",ui->bnoBox->text());
    sql.bindValue(":curDate",curDate);
    bool upbr = sql.exec();
    qDebug() << curDate;
    qDebug() << userid;
    qDebug() <<ui->bnoBox->text();
    //update card and book;
    sql.prepare("update book set stock = stock - 1 where bno = :bno");
    sql.bindValue(":bno",ui->bnoBox->text());
    bool upbk = sql.exec();

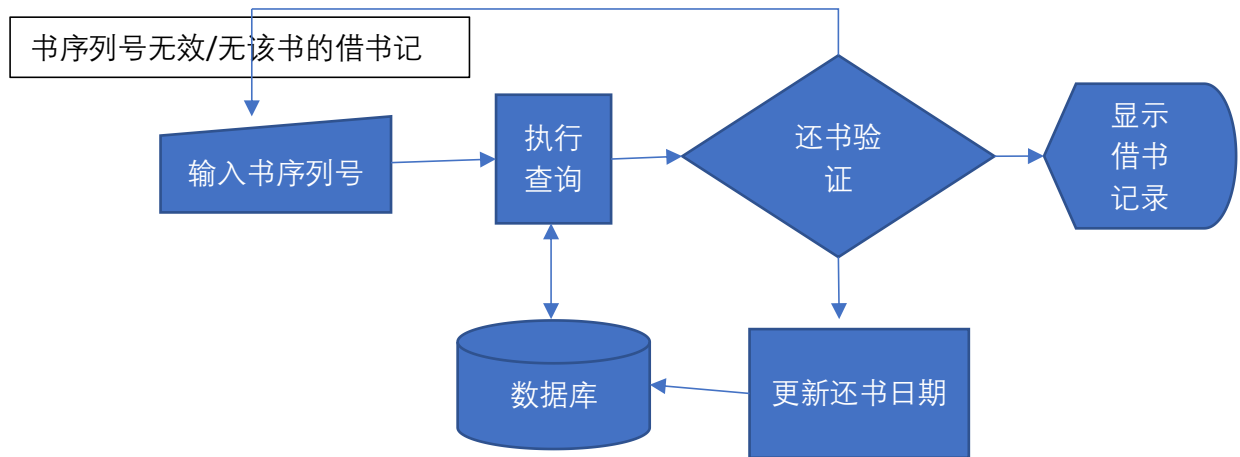
    sql.prepare("update card set available_book = available_book - 1 where cno
= :userid");
    sql.bindValue(":userid",userid);
    bool upcd = sql.exec();
    //judge
    if(upbr && upbk && upcd){
        QMessageBox::information(this,"title","borrow success!");
    }
    else{
        QMessageBox::warning(this,"title","borrowed but failed!");
    }
    sql.clear();
    //refresh the table info
    this->dataTableModel->clear();
    this->showhistory();
}
}
}

```

- 还书

在“书序列号”框内输入对应bno，点击还书按钮后，后台会根据①是否有对应借书记录②书序列号是否有效来进行判别是否借书成功。如若借书成功则会在borrow（借书记录表）中更新还书日期，并刷新显示借书记录

具体工作流程如下：



如图是输入“bno1”后点击还书的界面：

	cno	bno	title	borrowed date	return date	available book
1	cno1	bno1	SQL Server 完...	2010-06-05	2010-07-10	5
2	cno1	bno2	程序员的自我修...	2010-06-04	2010-07-10	
3	cno1	bno2	程序员的自我修...	2022-05-06	2010-07-10	
4	cno1	bno1	SQL Server 完...	2022-05-28	2022-05-28	

如上图显示，available_book增加了1，且第4条借书记录也更新了对应的还书时间。

具体代码实现如下：

```

void stupage::on_returnbtn_clicked()//还书
{
    if(ui->bnoBox->text().size()){
        QDate curDate = QDate::currentDate();
        //QString userno = userNumber;
        QString userno = QObject::tr("cno1");
        QSqlQuery sql;
    }
}
  
```

```

        //insert into borrow table
        sql.prepare("update borrow set return_date = :curDate where cno = :userno and bno
= :bno and return_date is NULL");
        sql.bindValue(":userno",userno);
        sql.bindValue(":bno",ui->bnoBox->text());
        sql.bindValue(":curDate",curDate);
        bool upbr = sql.exec();
        qDebug() << curDate;
        qDebug() << userno;
        qDebug() <<ui->bnoBox->text();
        //update card and book;
        sql.prepare("update book set stock = stock + 1 where bno = :bno");
        sql.bindValue(":bno",ui->bnoBox->text());
        bool upbk = sql.exec();

        sql.prepare("update card set available_book = available_book + 1 where cno
= :userno ");
        sql.bindValue(":userno",userno);
        bool upcd = sql.exec();
        //judge
        if(upbr && upbk && upcd){
            QLabel label( QLabel::tr("return success!") );
            label.show();
        }
        else{
            QMessageBox::warning(NULL,"return book","return processed but failed!");
        }
        sql.clear();
        //refresh the table info
        this->dataTableModel->clear();
        this->showhistory();
    }
}

```

3. 管理员界面

管理员界面主要分为借书卡管理和图书库管理两大部分。

1) 借书卡管理

借书卡信息主要由卡号，姓名，部门以及类别和可借书籍余量组成。

借书卡管理界面主要由借书卡查询，建卡，修改，删卡，批量导入，以及一个显示框组成。

具体如下图所示：

Form

	cno	name	department	type	available_book
29	admin	administrator		O	5
30	cno1	Zhang San	Computer Sci	U	5
31	cno2	Li Si	ISEE	U	5
32	cno3	Wang Wu	Ecnomics	U	5
33	cno4	Zhu Liu	Architechture	G	5
34	cno5	Yan Qi	management	T	5
35	cno6	Saul Goodman	Law	T	5
36	cno7	Trump	Comedy	O	5
37	cno8	Wu Di	KongFu	T	5
38	cno9	Bob	Language	T	5

建卡

cno (必填)

姓名

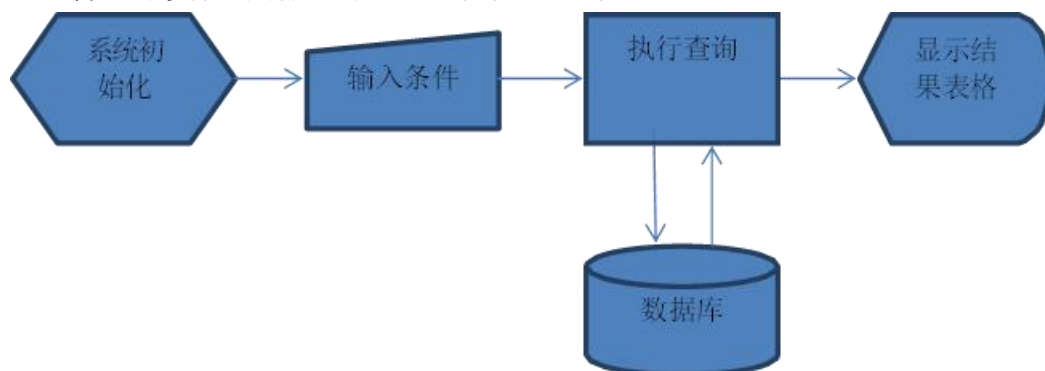
部门

身份

可用借书量

a) 查询:

借书卡查询在系统初始化完成后可直接开始使用，其基本工作流程是：输入查询条件→组合查询条件→执行查询→返回结果。如下图：



查询的条件有姓名。查询的方式为模糊查找，即可带有“%”符号进行前缀后缀或者前后缀的模糊查找。

实例：搜索以“Zhang”开头的借书卡

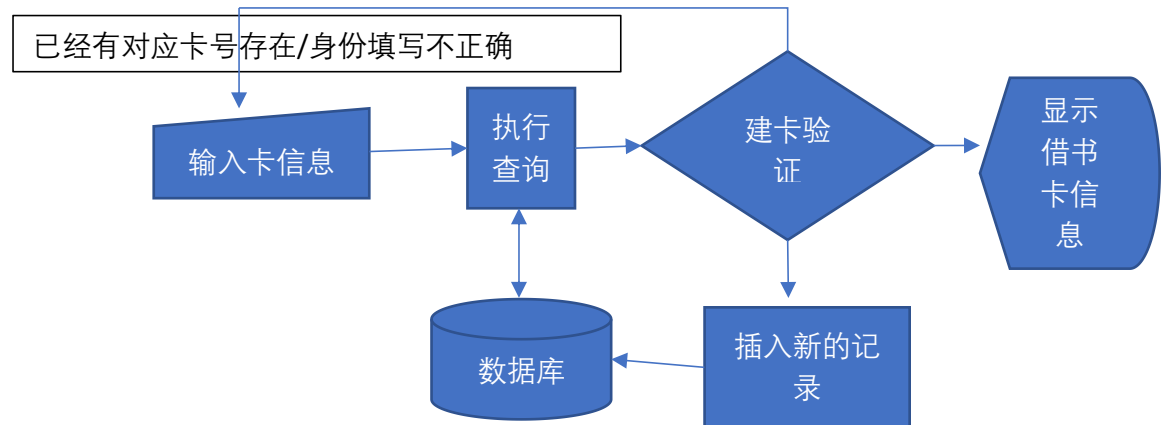
	cno	name	department	type	available_book
1	cno1	Zhang San	Computer Sci	U	5

具体代码实现与书籍查询类似

b) 建卡

点击界面上的“建卡”按钮后，会跳转到对应的输入界面，在输入界面中输入对饮卡号，姓名，部门，身份后，点击确定，在后台会查询是否有相同卡号的借书卡已经被建立、以及是否填入正确的身份，如果都满足即可完成建卡。

具体实现流程如下流程图所示：



查询		请输入姓名		建卡		修改信息		删卡	
		批量导入							
	cno	name	department	type	available_book				
29	admin	administrator		O	5				
30	cno0	Wei Yiyang	ISEE	U	5				
31	cno1	Zhang San	Computer Sci	U	5				
32	cno2	Li Si	ISEE	U	5				
33	cno3	Wang Wu	Ecnomics	U	5				
34	cno4	Zhu Liu	Architechture	G	5				
35	cno5	Yan Qi	management	T	5				
36	cno6	Saul Goodman	Law	T	5				
37	cno7	Trump	Comedy	O	5				
38	cno8	Wu Di	KongFu	T	5				

建卡

cno (必填)

姓名

部门

身份

可用借书量

如图所示为填入“cno0” “Wei Yiyang” “ISEE” “U” “5” 信息后建立的卡号信息，对应的新卡信息由红线标出。

具体实现代码如下：

```

void cardpage::on_okbottom_clicked()//新建卡
{
    //load in info of new book
    QString cno = ui->cnoBox->text();
    QString name = ui->nameBox->text();
    QString department = ui->departmentBox->text();
    QString type = ui->idBox->text();
    QString av_bks = ui->avbookBox->text();
  
```

```

qDebug()<<cno;
QDebug()<<name;
QDebug()<<department;
QDebug()<<type;
QDebug()<<av_bks;

QSqlQuery sql;
sql.prepare("insert into card values(:cno,:name,:department,:type,:av_bks)");
sql.bindValue(":cno",cno);
sql.bindValue(":name",name);
sql.bindValue(":department",department);
sql.bindValue(":type",type);
sql.bindValue(":av_bks",av_bks);

bool addIF = sql.exec();
if(addIF)
{
    QMessageBox::information(this,"title","create succeeds");
    this->getDatabaseInfo();
}
else
{
    QMessageBox::warning(this,"title","create failed");
}

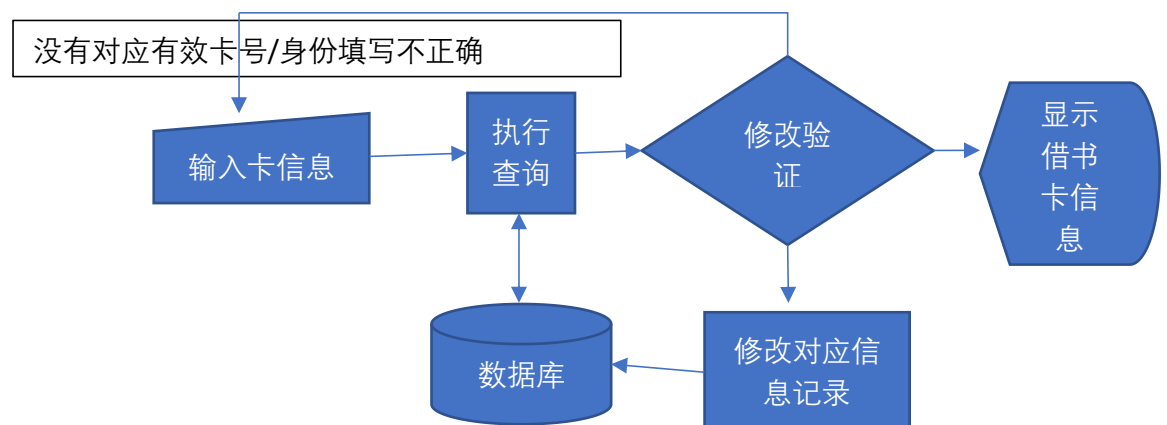
sql.clear();
}

```

c) 修改信息

点击界面上的“修改信息”按钮后，会跳转到对应的输入界面，在输入界面中输入对饮需要修改的信息：卡号，姓名，部门，身份后（卡号必填且不可修改，其余的信息按需要填入修改）点击确定，在后台会查询是否有有效的卡号、以及是否填入正确的身份，如果都满足即可完成信息修改。

具体实现流程如下流程图所示：



查询

请输入姓名

建卡

修改信息

删卡

批量导入

	cno	name	department	type	available_book
29	admin	administrator		O	5
30	cno0	wyy	Computer Sci	G	6
31	cno1	Zhang San	Computer Sci	U	5
32	cno2	Li Si	ISEE	U	5
33	cno3	Wang Wu	Ecnomics	U	5
34	cno4	Zhu Liu	Architechture	G	5
35	cno5	Yan Qi	managemant	T	5
36	cno6	Saul Goodman	Law	T	5
37	cno7	Trump	Comedy	O	5
38	cno8	Wu Di	KongFu	T	5

修改

cno (必填)

cno0

姓名

wyy

部门

Computer Sci

身份

G

可用借书量

6

确认

退出登录

如图所示为填入“cno0” “wyy” Computer Sci” “G” “6” 信息后修改成功的卡号信息，对应的卡信息由红线标出。

具体实现代码如下：

```
void cardpage::on_okbottom_2_clicked()//修改
{
    QString cno = ui->cnoBox_2->text();
    QString name =ui->nameBox_2->text();
    QString department =ui->departmentBox_2->text();
    QString type =ui->idBox_2->text();
    QString av_bks =ui->avbookBox_2->text();

    qDebug()<<cno;
    qDebug()<<name;
    qDebug()<<department;
    qDebug()<<type;
    qDebug()<<av_bks;
    QSqlQuery sql;

    if(name.size()){
        sql.prepare("update card set name = :name where cno = :cno");
        sql.bindValue(":cno",cno);
        sql.bindValue(":name",name);
        sql.exec();
    }
    if(department.size()){
        sql.prepare("update card set department = :department where cno = :cno");
        sql.bindValue(":cno",cno);
```

```

        sql.bindValue(":department",department);
        sql.exec();
    }
    if(type.size()){
        sql.prepare("update card set type = :type where cno = :cno");
        sql.bindValue(":cno",cno);
        sql.bindValue(":type",type);
        sql.exec();
    }
    if(av_bks.size()){
        sql.prepare("update card set available_book = :av_bks where cno = :cno");
        sql.bindValue(":cno",cno);
        sql.bindValue(":av_bks",av_bks);
        sql.exec();
    }

    bool modificationIF = sql.exec();

    if(modificationIF)
    {
        QMessageBox::information(this,"title","modification succeeds");
        this->getDatabaseInfo();
    }
    else
    {
        QMessageBox::warning(this,"title","modification failed");
    }

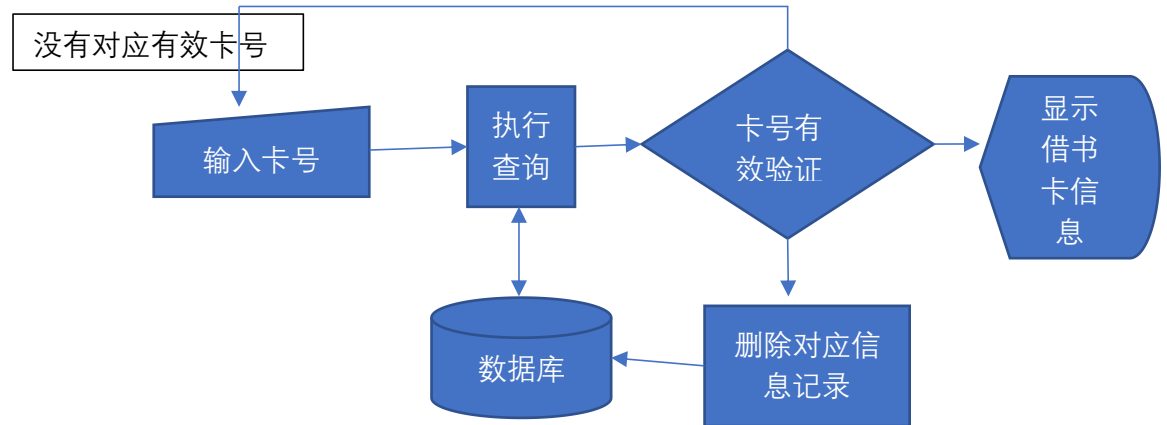
    sql.clear();
}

```

d) 删卡

点击界面上的“删卡”按钮后，会跳转到对应的输入界面，在输入界面中输入对应需要删除的卡的卡号，点击确定，在后台会查询是否有有效的卡号如果都满足即可完成删除。

具体实现流程如下流程图所示：



如图所示是输入了“cno0”，并确定删除后的管理界面

	cno	name	department	type	available_book
29	admin	administrator	.	O	5
30	cno1	Zhang San	Computer Sci	U	5
31	cno2	Li Si	ISEE	U	5
32	cno3	Wang Wu	Ecnomics	U	5
33	cno4	Zhu Liu	Architechture	G	5
34	cno5	Yan Qi	management	T	5
35	cno6	Saul Goodman	Law	T	5
36	cno7	Trump	Comedy	O	5
37	cno8	Wu Di	KongFu	T	5
38	cno9	Bob	Language	T	5

原先在红线区域附近的cno0卡号的信息已经被删除完毕
具体代码实现如下

```

void cardpage::on_Btndelete_clicked()//delete
{
    QString cno = ui->Boxdelete->text();//用户姓名

    qDebug()<<cno;

    QSqlQuery sql;
    sql.prepare("delete from card where cno = :cno");
    sql.bindValue(":cno",cno);

    sql.exec();
    bool deleteIF = sql.exec();
    if(deleteIF)
    {
        QMessageBox::information(this,"title","deletion succeeds");
    }
}
  
```

```

    }
    else
    {
        QMessageBox::warning(this,"title","deletion failed");
    }

    this->dataTableModel->clear();
    this->setTableInfo();
    sql.clear();
}

```

e) 批量导入

批量导入利用到了QT中的AxObjec 类以及文件操作，完成了与excel文件的连接和处理。在用户点击“批量导入”按钮后，会转到文件选择界面，选择一个xlsx或xls文件，会自动完成对文件内信息进入数据库的导入。

(注：在表格文件中的信息从第二行开始读入，第一行为表头信息如“卡号”，“姓名“等)

	A	B	C	D	E
1	cno	name	departme	type	num_book
2	0	name0	none	U	5
3	1	name1	none	U	5
4	2	name2	none	U	5
5	3	name3	none	U	5
6	4	name4	none	U	5
7	5	name5	none	U	5
8	6	name6	none	U	5
9	7	name7	none	U	5
10	8	name8	none	U	5
11	9	name9	none	U	5
12	10	name10	none	U	5
13	11	name11	none	U	5
14	12	name12	none	U	5
15	13	name13	none	U	5
16	14	name14	none	U	5
17	15	name15	none	U	5
18	16	name16	none	U	5
19	17	name17	none	U	5
20	18	name18	none	U	5
21	19	name19	none	U	5
22	20	name20	none	U	5
23	21	name21	none	U	5
24	22	name22	none	U	5
25	23	name23	none	U	5
26	24	name24	none	U	5
27	25	name25	none	U	5
28	26	name26	none	U	5
29	27	name27	none	U	5
30					
31					

如图是需要导入的excel内容

查询 建卡

	cno	name	department	type	available_book
1	0	name0	none	U	5
2	1	name1	none	U	5
3	10	name10	none	U	5
4	11	name11	none	U	5
5	12	name12	none	U	5
6	13	name13	none	U	5
7	14	name14	none	U	5
8	15	name15	none	U	5
9	16	name16	none	U	5
10	17	name17	none	U	5

查询 建卡

	cno	name	department	type	available_book
10	17	name17	none	U	5
11	18	name18	none	U	5
12	19	name19	none	U	5
13	2	name2	none	U	5
14	20	name20	none	U	5
15	21	name21	none	U	5
16	22	name22	none	U	5
17	23	name23	none	U	5
18	24	name24	none	U	5
19	25	name25	none	U	5

查询 建卡

	cno	name	department	type	available_book
23	4	name4	none	U	5
24	5	name5	none	U	5
25	6	name6	none	U	5
26	7	name7	none	U	5
27	8	name8	none	U	5
28	9	name9	none	U	5
29	admin	administrator		O	5
30	cno1	Zhang San	Computer Sci	U	5
31	cno2	Li Si	ISEE	U	5
32	cno3	Wang Wu	Ecnomics	U	5

导入后的信息。

代码如下：

```
void cardpage::on_excelbtn_clicked()
{
    QString fileName = QFileDialog::getOpenFileName(
```

```

        this,
        tr("open a file."),
        "C:/Users/22470/Documents/lab5_1/excel",
        tr("excel(*.xls *.xlsx );;All files(*.*)"));
//寻找文件
if (fileName.isEmpty() || fileName.count(".xls") == 0) {
    QMessageBox::warning(this, "Warning!", "Please select an excel");
}

else{//找到了文件
    QMessageBox::information(this,"patch import",fileName + " found
successfully");
    QAxObject *excel = new QAxObject("Excel.Application",this);
    excel->dynamicCall("SetVisible(bool)", false);
    excel->setProperty("DisplayAlerts", false);
    QAxObject* workbooks = excel->querySubObject("WorkBooks");//获取工作簿集合
    QAxObject* workbook = workbooks->querySubObject("Open(Qstring
&)",fileName);
    QAxObject* worksheet = workbook->querySubObject("WorkSheets(int)",1);
    //获取第一个集合表

    QAxObject* usedrange = worksheet->querySubObject("UsedRange"); // sheet 范
围

    int intRowStart = usedrange->property("Row").toInt(); // 起始行数 为1
    QAxObject *rows;
    rows = usedrange->querySubObject("Rows"); // 行
    int intRow = rows->property("Count").toInt(); // 行数
    //int intCol = columns->property("Count").toInt(); // 列数
    QAxObject* cell; // 单元格
    for(int i = intRowStart+1; i < intRow + intRowStart; i++){
        //get values
        cell = worksheet -> querySubObject("Cells(int,int)",i,1);
        QString cno = cell -> dynamicCall("text").toString();
        cell = worksheet -> querySubObject("Cells(int,int)",i,2);
        QString name = cell -> dynamicCall("text").toString();
        cell = worksheet -> querySubObject("Cells(int,int)",i,3);
        QString department = cell -> dynamicCall("text").toString();
        cell = worksheet -> querySubObject("Cells(int,int)",i,4);
        QString type = cell -> dynamicCall("text").toString();
        cell = worksheet -> querySubObject("Cells(int,int)",i,5);
        int av_bks = cell -> dynamicCall("Value2()").toInt();
        QSqlQuery sql;
        //qDebug() <<

```

```

        sql.prepare("insert into card
values(:cno,:name,:department,:type,:av_bks)");
        sql.bindValue(":cno",cno);
        sql.bindValue(":name",name);
        sql.bindValue(":department",department);
        sql.bindValue(":type",type);
        sql.bindValue(":av_bks",av_bks);
        sql.exec();
        sql.clear();
    }
    this->dataTableModel->clear();
    this->setTableInfo();
    workbook->dynamicCall("Close()");
    excel->dynamicCall("Quit()");
    if (excel)
    {
        delete excel;
        excel = NULL;
    }
}
}

```

2) 图书库管理

借书卡信息主要由书号，书名，作者，价格，总库存，剩余库存，种类，出版社组成。（因为组成和功能结构与借书卡管理类似，且由于篇幅原因，具体代码就不放了，主要解析结构上的功能实现过程 and 用户界面的表现。）

书库管理界面主要由图书查询，新书入库，修改信息，删除书籍，批量导入，以及一个显示框组成。具体如下图所示：

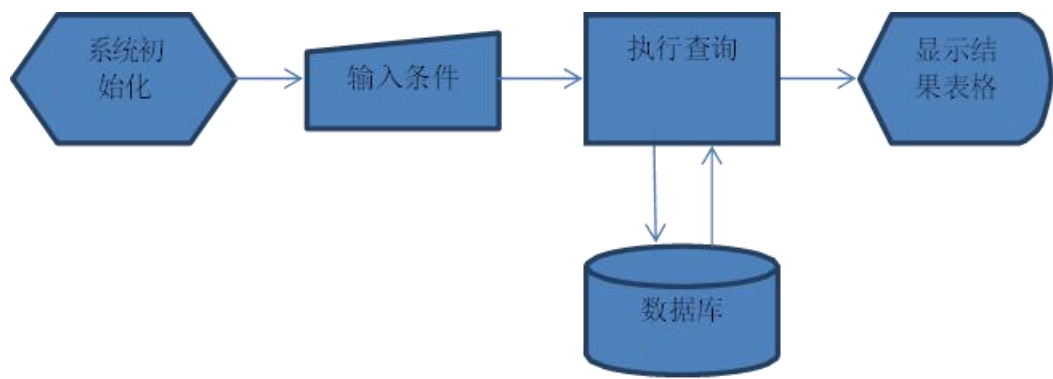
The interface includes a search bar at the top left with a '查询' (Search) button. To the right are buttons for '新书入库' (New Book In), '修改信息' (Modify Info), '删除书籍' (Delete Book), and '批量导入' (Batch Import). The main area contains a table with the following data:

	bno	title	author	price	total	stock	category	press
1	bno1	SQL Server 完全学习手册	郭郑州	79.8	5	4	计算机	清华大学出版社
2	bno2	程序员的自我修养	俞甲子	65	5	4	计算机	电子工业出版社
3	bno3	做新教育的行者	高云鹏	18.5	3	2	教育	福建教育出版社
4	bno4	阴阳太极命理	王庆	36.5	3	0	文化	广西师范大学出版社
5	bno5	实用英文写作	庞集贤	50.5	3	2	外语	高等教育出版社
6	bno6	数字系统设计	童教授	49	4	2	电子	清华大学出版社
7	bno7	模拟电路设计	童教授	50	4	2	电子	清华大学出版社

On the right, there is a form titled '入库新书' (New Book In) with the following fields: bno (必填), 书名, 作者, 价格, 总藏书, 本馆剩余, 种类, and 出版社. A '确认' (Confirm) button is at the bottom right of the form. At the bottom of the interface is a '退出登录' (Logout) button.

d) 查询：

借书卡查询在系统初始化完成后可直接开始使用，其基本工作流程是：输入查询条件→组合查询条件→执行查询→返回结果。如下图：



查询的条件有书名。查询的方式为模糊查找，即可带有“%”符号进行前缀后缀或者前后缀的模糊查找。

实例：搜索以“SQL”开头的借书卡

Form

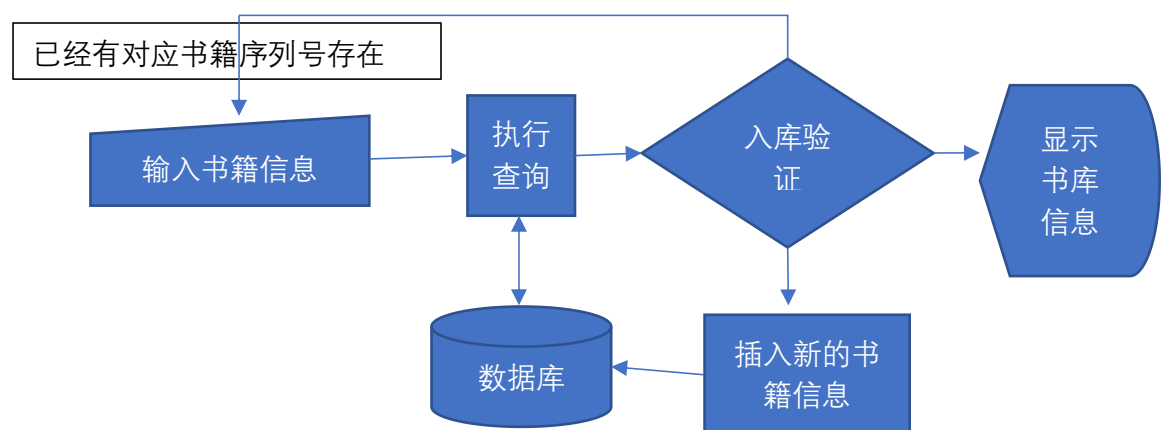
查询		SQL%		新书入库		修	
						批	
	bno	title	author	price	total	stock	category
1	bno1	SQL Server 完...	郭郑州	79.8	5	4	计算机

具体代码实现与书籍查询类似

e) 书籍入库

点击界面上的“书籍入库”按钮后，会跳转到对应的输入界面，在输入界面中输入对应 bno，书名，作者，价格，总藏书，本馆剩余，种类，出版社后，点击确定，在后台会查询是否有相同序列号的书籍已经被入库，如果都满足即可完成建卡。

具体实现流程如下流程图所示：



Form

查询 新书入库 修改信息 删除书籍 批量导入

	bno	title	author	price	total	stock	category	press
1	bno0	name	me	999	10	10	class	ZJU press
2	bno1	SQL Server 完全学习手册	郭郑州	79.8	5	4	计算机	清华大学出版社
3	bno2	程序员的自我修养	俞甲子	65	5	4	计算机	电子工业出版社
4	bno3	做新教育的行者	高云鹏	18.5	3	2	教育	福建教育出版社
5	bno4	阴阳太极命理	王庆	36.5	3	0	文化	广西师范大学出版社
6	bno5	实用英文写作	庞集贤	50.5	3	2	外语	高等教育出版社
7	bno6	数字系统设计	童教授	49	4	2	电子	清华大学出版社
8	bno7	模拟电路设计	童教授	50	4	2	电子	清华大学出版社

入库新书

bno (必填)

书名

作者

价格

总藏书

本馆剩余

种类

出版社

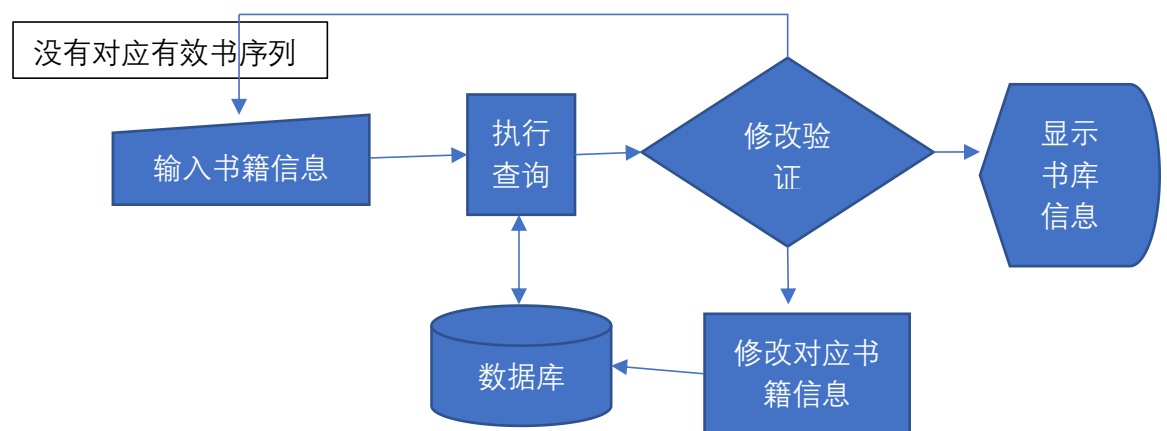
确认

如图所示为填入对应图中右侧框内信息后建立的卡号信息，对应的新卡信息由红线标出。

f) 修改信息

点击界面中的“修改信息”按钮后，会跳转到对应的输入界面，在输入界面中输入对饮需要修改的信息：书名，作者，价格，总藏书，本馆剩余，种类，出版社后（书序列号必填且不可修改，其余的信息按需要填入修改）点击确定，在后台会查询是否有有效的卡号、以及是否填入正确的身份，如果都满足即可完成信息修改。

具体实现流程如下流程图所示：



Form

查询 新书入库 修改信息 删除书籍 批量导入

	bno	title	author	price	total	stock	category	press
1	bno0	Database exp	you	233	99	99	计算机	ZJU press
2	bno1	SQL Server 完全学习手册	郭郑州	79.8	5	4	计算机	清华大学出版社
3	bno2	程序员的自我修养	俞甲子	65	5	4	计算机	电子工业出版社
4	bno3	做新教育的行者	高云鹏	18.5	3	2	教育	福建教育出版社
5	bno4	阴阳太极命理	王庆	36.5	3	0	文化	广西师范大学出版社
6	bno5	实用英文写作	庞集贤	50.5	3	2	外语	高等教育出版社
7	bno6	数字系统设计	童教授	49	4	2	电子	清华大学出版社
8	bno7	模拟电路设计	童教授	50	4	2	电子	清华大学出版社

修改

bno (必填)

书名

作者

价格

总藏书

本馆剩余

种类

出版社

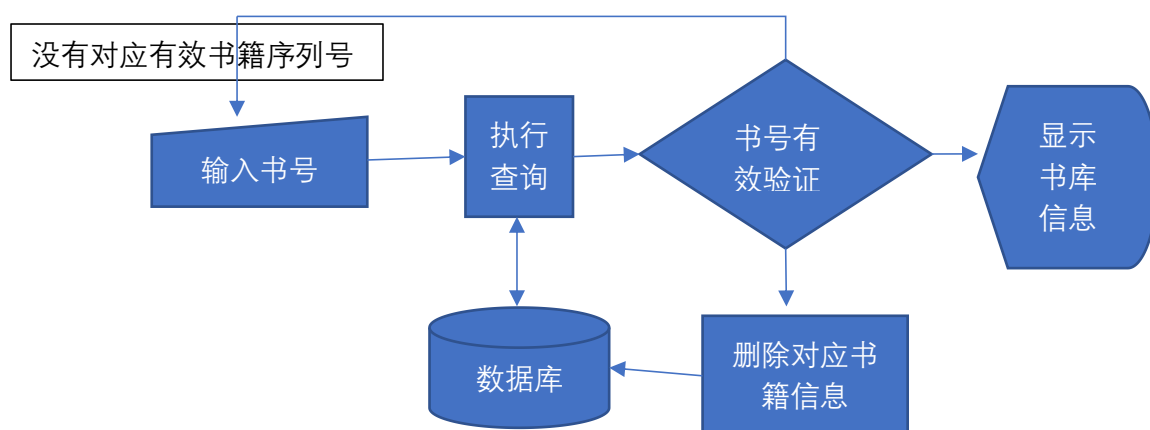
确定

如图所示为填入图中右侧框内信息后修改成功的卡号信息，对应的新卡信息由红线标出。

f) 删除书籍

点击界面上的“删除书籍”按钮后，会跳转到对应的输入界面，在输入界面中输入对应需要删除的书的书籍序列号，点击确定，在后台会查询是否有有效的书籍序列号，如果都满足即可完成删除。

具体实现流程如下流程图所示：



	bno	title	author	price	total	stock	category	press
1	bno1	SQL Server 完全学习手册	郭郑州	79.8	5	4	计算机	清华大学出版社
2	bno2	程序员的自我修养	俞甲子	65	5	4	计算机	电子工业出版社
3	bno3	做新教育的行者	高云鹏	18.5	3	2	教育	福建教育出版社
4	bno4	阴阳太极命理	王庆	36.5	3	0	文化	广西师范大学出版社
5	bno5	实用英文写作	庞集贤	50.5	3	2	外语	高等教育出版社
6	bno6	数字系统设计	童教授	49	4	2	电子	清华大学出版社
7	bno7	模拟电路设计	童教授	50	4	2	电子	清华大学出版社

如图所示是输入了“bno0”，并确定删除后的管理界面

原先在第一行的的cno0卡号的信息已经被删除完毕

g) 批量导入

批量导入利用到了QT中的AxObjec 类以及文件操作，完成了与excel文件的连接和处理。在用户点击“批量导入”按钮后，会转到文件选择界面，选择一个xlsx或xls文件，会自动完成对文件内信息进入数据库的导入。

(注：在表格文件中的信息从第二行开始读入，第一行为表头信息如“书名”，“作者”等)

	A	B	C	D	E	F	G	H	I	J
1	bno	title	author	price	total	stock	category	press		
2	0	name0	author0	20	5	5	cat0	press0		
3	1	name1	author1	20	5	5	cat1	press1		
4	2	name2	author2	20	5	5	cat2	press2		
5	3	name3	author3	20	5	5	cat3	press3		
6	4	name4	author4	20	5	5	cat4	press4		
7	5	name5	author5	20	5	5	cat5	press5		
8	6	name6	author6	20	5	5	cat6	press6		
9	7	name7	author7	20	5	5	cat7	press7		
10	8	name8	author8	20	5	5	cat8	press8		
11	9	name9	author9	20	5	5	cat9	press9		
12	10	name10	author10	20	5	5	cat10	press10		
13	11	name11	author11	20	5	5	cat11	press11		
14	12	name12	author12	20	5	5	cat12	press12		
15	13	name13	author13	20	5	5	cat13	press13		
16	14	name14	author14	20	5	5	cat14	press14		
17	15	name15	author15	20	5	5	cat15	press15		
18	16	name16	author16	20	5	5	cat16	press16		
19	17	name17	author17	20	5	5	cat17	press17		
20	18	name18	author18	20	5	5	cat18	press18		
21										

图中所示为需要导入的excel书籍信息

点击批量导入并选择books.xlsx后即可完成导入

查询

新书入库

修改

批量

	bno	title	author	price	total	stock	category	press	
1	0	name0	author0	20	5	5	0	20	^
2	1	name1	author1	20	5	5	0	20	
3	10	name10	author10	20	5	5	0	20	
4	11	name11	author11	20	5	5	0	20	
5	12	name12	author12	20	5	5	0	20	
6	13	name13	author13	20	5	5	0	20	
7	14	name14	author14	20	5	5	0	20	
8	15	name15	author15	20	5	5	0	20	
9	16	name16	author16	20	5	5	0	20	
10	17	name17	author17	20	5	5	0	20	

查询

新书入库

修改信息

批量导入

	bno	title	author	price	total	stock	category	press	
11	18	name18	author18	20	5	5	0	20	<div>^</div>
12	2	name2	author2	20	5	5	0	20	
13	3	name3	author3	20	5	5	0	20	
14	4	name4	author4	20	5	5	0	20	
15	5	name5	author5	20	5	5	0	20	
16	6	name6	author6	20	5	5	0	20	
17	7	name7	author7	20	5	5	0	20	
18	8	name8	author8	20	5	5	0	20	
19	9	name9	author9	20	5	5	0	20	
20	bno1	SQL Server 完全学习手册	郭郑州	79.8	5	4	计算机	清华出版社	

查询

新书入库

修改信息

批量导入

	bno	title	author	price	total	stock	category	press
17	7	name7	author7	20	5	5	0	20
18	8	name8	author8	20	5	5	0	20
19	9	name9	author9	20	5	5	0	20
20	bno1	SQL Server 完全学习手册	郭郑州	79.8	5	4	计算机	清华出版社
21	bno2	程序员的自我修养	俞甲子	65	5	4	计算机	电子工业出版社
22	bno3	做新教育的行者	高云鹏	18.5	3	2	教育	福建教育出版社
23	bno4	阴阳太极命理	王庆	36.5	3	0	文化	广西师范大学出版社
24	bno5	实用英文写作	庞集贤	50.5	3	2	外语	高等教育出版社
25	bno6	数字系统设计	童教授	49	4	2	电子	清华出版社
26	bno7	模拟电路设计	童教授	50	4	2	电子	清华出版社

完成导入后的书库信息如上图所示