



浙江大学  
Zhejiang University

ZHEJIANG UNIVERSITY

DESIGN REPORT

---

## 简单图书馆系统设计报告

---

*Author:*

邓 墨琳

*Stu ID:*

No. 3150103457

2018 年 4 月 17 日

# 摘要

本次作业使用 Java 编写，采用 JDBC 接口执行 SQL 相关操作，采用 Swing 开发简易的图形界面，设计并实现了图书馆系统，具有图书入库、借书、还书、借书证管理、管理员注册等功能。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 transaction 事务处理，完整性约束等数据库原理的理解。

**关键词:** JDBC, SQL server, 事务处理, Java, 图书馆系统

# 目录

摘要	I
第 1 章 设计要求	1
1.1 数据对象	1
1.2 功能模块	1
1.2.1 管理员登陆	1
1.2.2 图书入库	1
1.2.3 图书查询	1
1.2.4 借书	1
1.2.5 还书	2
1.2.6 借书证管理	2
1.3 数据库平台	2
1.4 编译环境	2
第 2 章 整体设计	3
2.1 ER 关系图	3
2.2 数据库设计	4
2.3 程序架构流程图	5
第 3 章 详细设计	6
3.1 数据库	6
3.1.1 book 关系	6
3.1.2 card 关系	6
3.1.3 admin 关系	7
3.1.4 borrow 关系	7
3.1.5 trigger	8
3.2 应用层权限	9
3.2.1 图书管理员	9
3.2.2 普通用户	9
3.3 类设计	10
3.3.1 UML 关系图	10
3.3.2 LogIn.java	10
3.3.3 Register.java	13
3.3.4 Menu.java	16
3.3.5 StockIn.java	17
3.3.6 Card.java	20

3.3.7	Query.java . . . . .	21
3.3.8	Bor_Ret.java . . . . .	24
3.3.9	MaDate.java . . . . .	28
<b>第 4 章</b>	<b>运行结果截图</b>	<b>30</b>
4.1	管理员登陆 . . . . .	30
4.1.1	图书入库 . . . . .	31
4.1.2	管理员注册 . . . . .	35
4.1.3	查询 . . . . .	36
4.1.4	借还书 . . . . .	38
4.1.5	借书证管理 . . . . .	40
4.2	普通用户登陆 . . . . .	44
4.2.1	借还书 . . . . .	46
4.2.2	查询 . . . . .	46
<b>A</b>	<b>SQL Source Code</b>	<b>47</b>
<b>B</b>	<b>Java Source Code</b>	<b>49</b>

## 第 1 章 设计要求

### 1.1 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)

表 1.1: 数据对象

### 1.2 功能模块

#### 1.2.1 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.

#### 1.2.2 图书入库

- 单本入库
- 批量入库

#### 1.2.3 图书查询

- 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. 返回最多前五十条符合条件的图书信息, 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)
- 可以按用户指定属性对图书信息进行排序. (默认是书名)

#### 1.2.4 借书

- 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)
- 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。

### 1.2.5 还书

- 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)
- 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.

### 1.2.6 借书证管理

增加或删除一个借书证.

## 1.3 数据库平台

SQL Server 2017

## 1.4 编译环境

- Eclipse OXYGEN 3
- Java(JDK 8)
- JDBC Driver: Microsoft JDBC Driver 6.2 for SQL Server

## 第 2 章 整体设计

### 2.1 ER 关系图

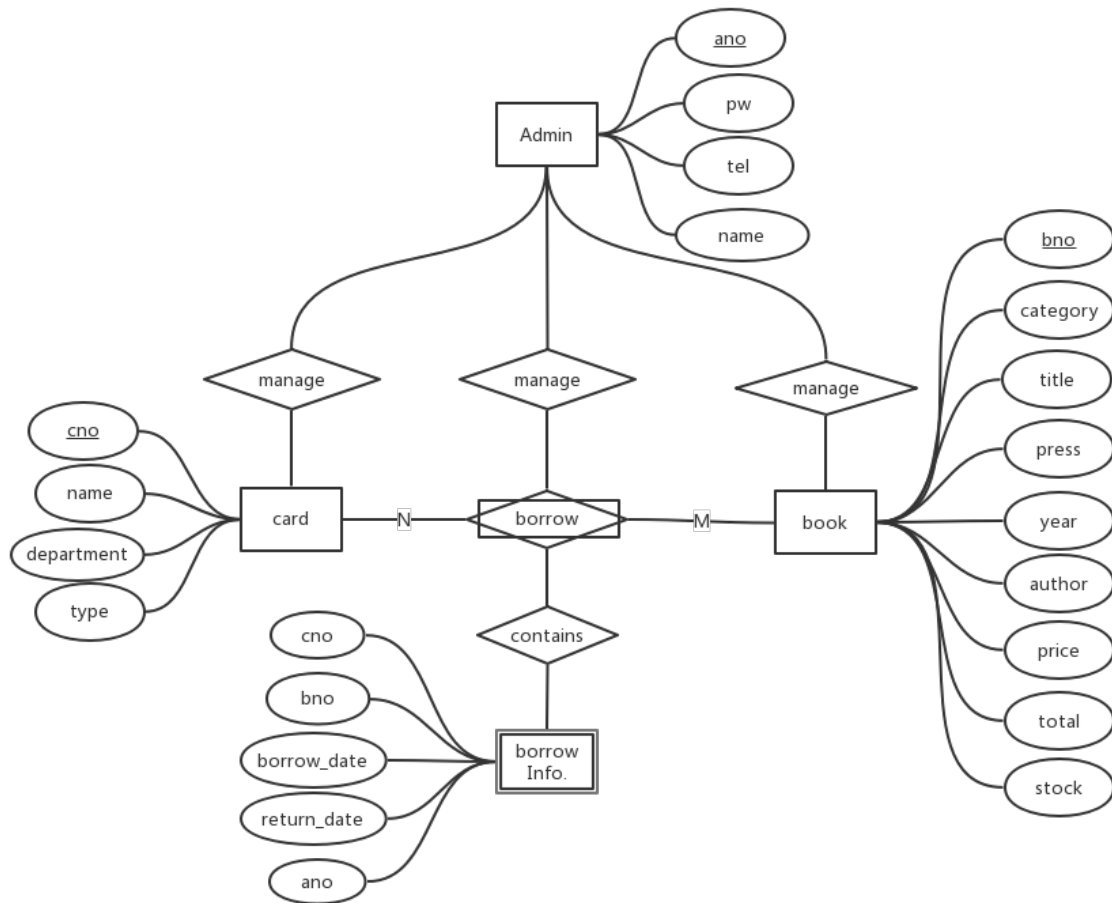


图 2.1: ER Model

## 2.2 数据库设计

数据库命名为 LMS(Library Managemnet System)

- book 表，记录图书馆书籍信息，主键为 bno(书号)
- card 表，记录借书证信息，主键为 cno(借书证号)
- admin 表，记录管理员信息，主键为 ano 管理员管理员账号
- borrow 表，记录借书信息，外键 cno 级联更新，外键 bno 级联删除与更新

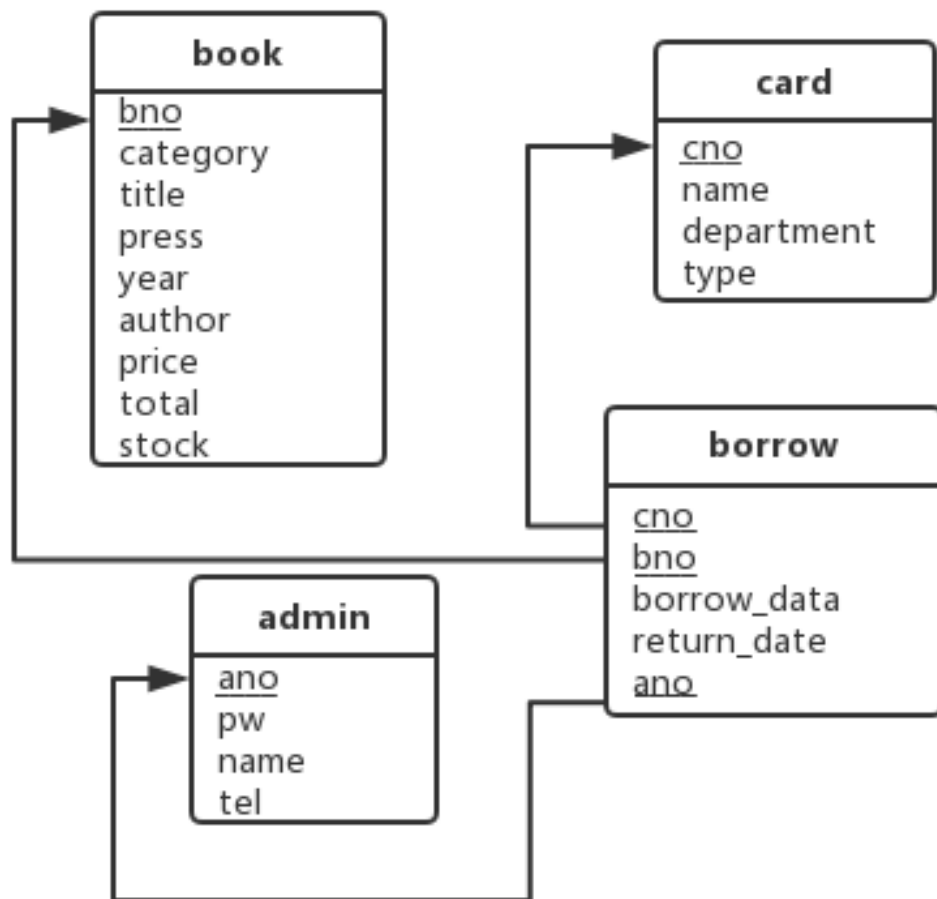


图 2.2: Schema diagram for the Library database



## 2.3 程序架构流程图

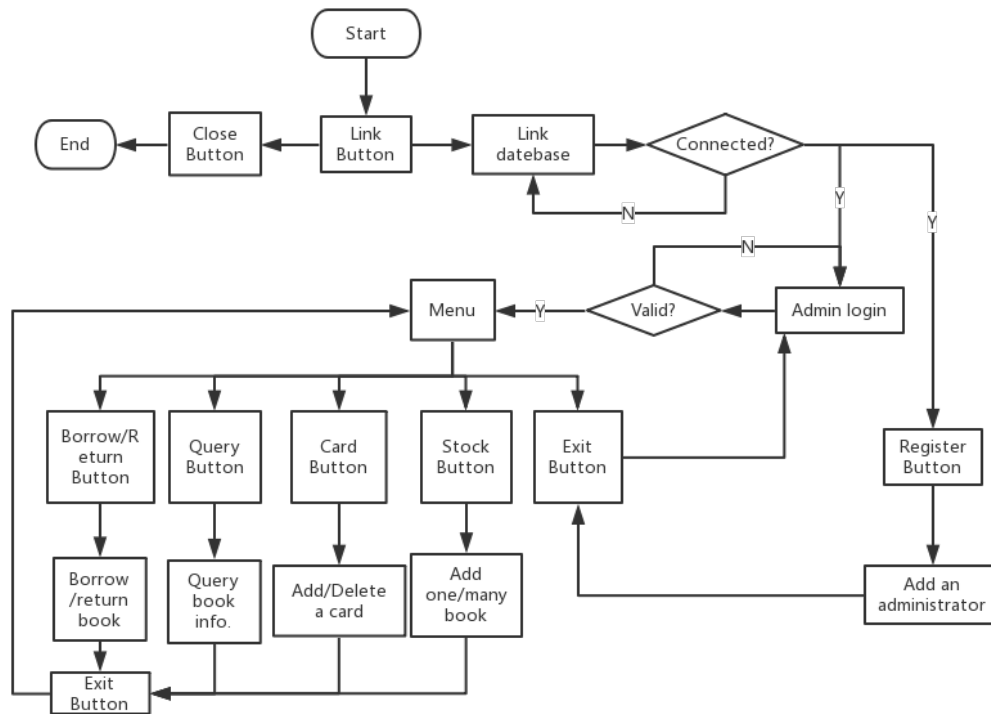


图 2.3: Flow chart for program

## 第 3 章 详细设计

### 3.1 数据库

#### 3.1.1 book 关系

book 关系中有 9 个字段，其中 bno 表示图书编号，为主键。值得注意的是，在数据库中除主键外的其余字段没有设置非空约束，而实际在图书馆中，书籍的信息均不为空，这一点在程序中检测，保证入库图书信息完整。

```
1 create table book
2 (
3 bno char(8) primary key,
4 category char(10),
5 title varchar(40),
6 press varchar(30),
7 year int,
8 author varchar(20),
9 price decimal(7,2),
10 total int,
11 stock int
12 );
```

#### 3.1.2 card 关系

card 关系记录借书证信息，cno 为借书证编号，为主键，同时约束个人信息不能为 null 值，而借书证类型只能属于集合  $[U, G, T, O]$ 。程序会检测 “” 空串的输入，保证信息的有效性。

```
1 create table card
2 (
3 cno char(7) not null primary key,
4 name varchar(10) not null,
5 department varchar(40) not null,
6 type char(1) check(type in ('T', 'G', 'U', 'O'))
7 );
```

### 3.1.3 admin 关系

admin 关系记录图书管理员信息，ano 为管理员登录账号，为主键，除个人电话字段 (tel) 外，其余字段不允许空值。注册管理员时，程序会检测空串的输入，保证数据完整性。

```
1 create table admin
2 (
3 ano varchar(10) not null ,
4 pw varchar(20) not null ,
5 name varchar(10) not null ,
6 tel char(11) null ,
7 primary key(ano)
8 );
```

### 3.1.4 borrow 关系

borrow 关系记录借书信息，外键为 bno, cno 和 ano，表示该借书证下所借书籍标号和处理这次借阅的管理员。借、还时间的数据结构为 date，只包含年月日信息。该关系中没有设置主键，因为同一张借书证可以在同一天，经由同一个管理员，借得两本相同得书籍。虽然如此，程序中将检测同一张借书证是否重复借书，保证在归还之前，同一本书籍只能借一次。

- 外键 cno 级联更新，保证当某张借书证仍有借书记录时，不能注销 (即在数据库中删除)
- 外键 bno 级联删除，考虑到当图书馆不在需要某本书时，关于这本书得借书记录也就不在重要
- 外键 bno 级联更新，考虑书号更新时，借书信息一同更新

```
1 create table borrow
2 (
3 cno char(7) ,
4 bno char(8) ,
5 borrow_date date ,
6 return_date date ,
7 ano varchar(10) ,
8 foreign key(cno) references card on update cascade ,
9 foreign key(bno) references book on update cascade
```

```

10 on delete cascade ,
11 foreign key(ano) references admin
12 );

```

### 3.1.5 trigger

由于管理员身份得特殊性，在数据库中建立先验触发器，检测管理员信息是否为空串。

```

1 CREATE TRIGGER chec_empty ON admin
2 INSTEAD of INSERT
3 AS
4 BEGIN
5     DECLARE @ano VARCHAR(10)
6     DECLARE @pw VARCHAR(20)
7     DECLARE @name VARCHAR(10)
8     SELECT @ano = ano FROM inserted
9     SELECT @pw = pw FROM inserted
10    SELECT @name = name FROM inserted
11    IF (@ano = '' OR @ano is null
12        OR @pw = '' OR @pw is null
13        OR @name = '' OR @name is null)
14        print ( 'Insert refused ' )
15    ELSE
16        INSERT into admin SELECT * FROM inserted
17 END

```

## 3.2 应用层权限

程序中存在两种角色：图书管理员和普通用户。其不同权限通过代码层面实现，在数据库中没有相关设置。

### 3.2.1 图书管理员

功能	权限
登陆	以管理员账号和密码登陆
管理员注册	✓
借书证注册、删除	✓
处理借书、还书	✓
查询	✓
图书入库	✓

表 3.1: 管理员权限

管理员能处理任何借书证的借还书。

### 3.2.2 普通用户

功能	权限
登陆	以借书证号 cno 登陆
管理员注册	×
借书证注册、删除	×
借书、还书	✓
查询	✓
图书入库	×

表 3.2: 普通用户权限

普通用户仅能处理自己账户的借还书。

## 3.3 类设计

### 3.3.1 UML 关系图

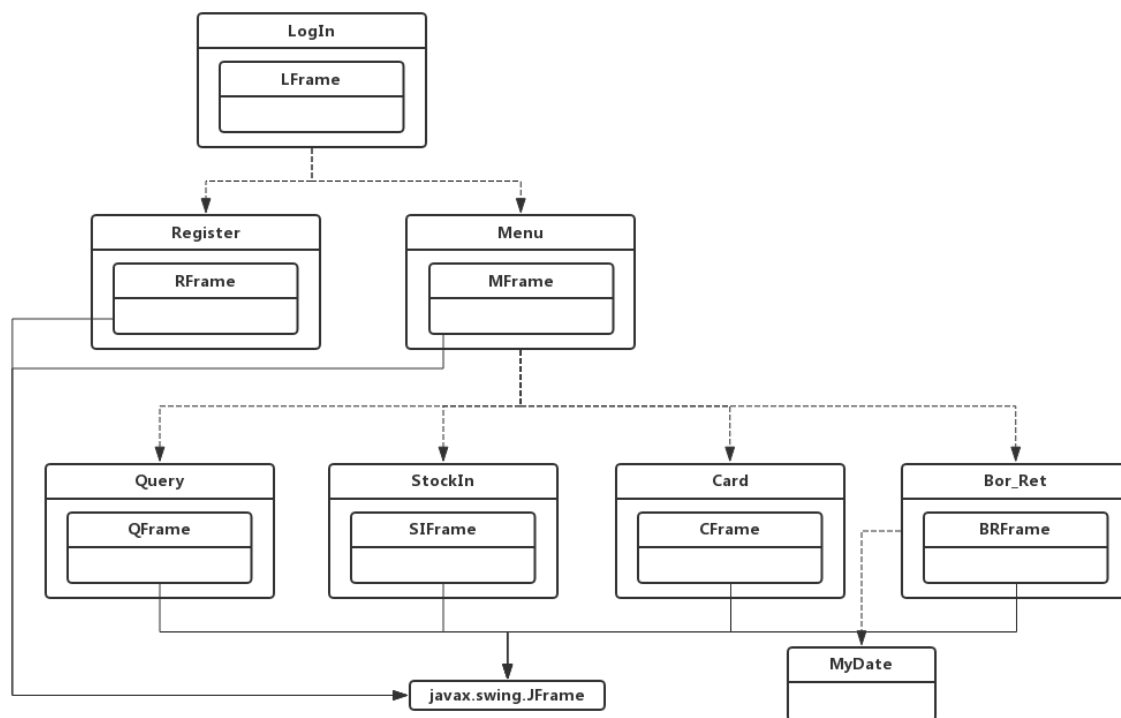


图 3.1: UML diagram

### 3.3.2 LogIn.java

LogIn 类中定义了内部类 LFrame, LFrame 继承了 JFrame, 用于绘制窗口, 其中也定义了数据库操作。LogIn 类依赖 Register 类和 Menu 类, 相应事件响应后创建注册界面和 Menu 界面。

#### 成员变量

- -driverName:String JDBC 驱动类型,连接 SQL server 时,默认值为"com.microsoft.sqlserver.jdbc"
- -dbURL:String 指定服务器地址 (本地地址)、监听端口和数据库名称, 在本作业中, 该值为"jdbc:sqlserver://localhost:1433;DatabaseName=LMS"
- -Name:String 数据库用户登陆名
- -Pwd:String 数据库用户登陆密码
- +conn:Connection Connection 接口变量, 设置为静态, 以便其他类引用。

- +isconn:boolean 该布尔值记录数据库是否已连接
- +curr\_admin:String 记录当前登陆得管理员 id, 以便处理接还记录时方便获取经手人。该变量设置为静态。默认值为'system', 当非管理员登陆, 进行借书操作时, 以系统管理员 'system' 登记当前经手人。
- +curr\_user:String 记录当前普通用户登陆后的 cno(借书证号), 普通用户登陆系统后, 只能查询、借还书。
- +Lmode:int Lmode = 0 登陆模式为普通用户, Lmode = 1 登陆模式为管理员。Lmode 默认值为 0, 设置此变量目的为方便执行不同的交互模式。

#### 成员函数 成员函数为内部类 LFrame 成员函数

- +LFrame() 构造函数, 调用 setWindows() 和 EventsListener()
- +EventListener():void 事件监听函数
- +setWindows():void 窗口绘制函数
- +SQL\_loginCheck():boolean SQL 操作函数, 查询管理员登陆账号是否在数据库中, 检查密码是否正确。

流程图 登陆模块流程图如下所示。

流程图仅为管理员登陆模式流程图，而普通用户登陆后仅拥有部分操作权限，流程大致相似，在此省略，下同。

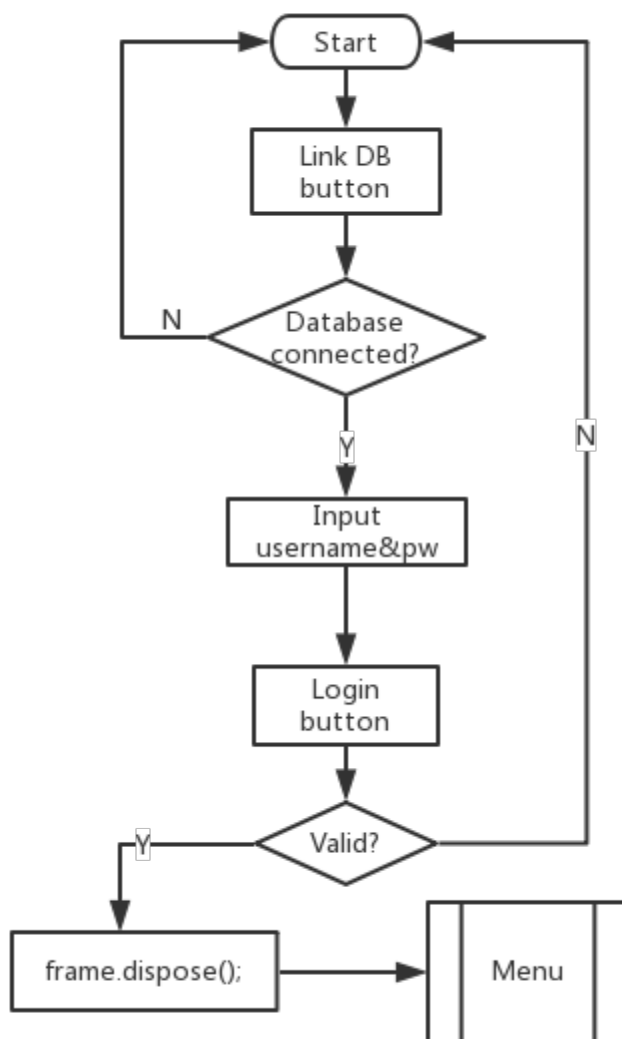


图 3.2

## 数据库处理

- 连接数据库：Link DB 按钮按下后，在事件监听函数中执行

```
1 Class.forName(driverName);  
2 conn = DriverManager.getConnection(dbURL,Name,Pwd);
```

连接成功或失败，通过 `JOptionPane.showMessageDialog()` 接口显示对应信息。



- 账号密码查询：调用 SQL\_loginCheck() 方法

```

1 public boolean SQL_loginCheck(String username) {
2     String sql = "select * from admin where ano = ?";
3     @SuppressWarnings("deprecation")
4     String pw = pwField.getText();
5     try {
6         PreparedStatement pStm = conn.prepareStatement(sql);
7         pStm.setString(1, username);
8         ResultSet rs = pStm.executeQuery();
9         if(rs.next())
10             return rs.getString("pw").equals(pw);
11         else return false;
12     } catch (SQLException e) {
13         return false;
14     }
15 }

```

该函数使用了 PreparedStatement 接口,可以向 sql 语句中传递参数,sql 语句"select \* from admin where ano = ?" 中的'?' 即参数的位置。调用 executeQuery() 后,将结果集储存在 rs 中,由于 admin 表中 ano 为主键,结果集中最多只有一条记录,调用 equals() 接口判断用户输入密码与储存密码是否一致。

**dispose() 方法** 当进入 Menu 或 Register 界面时,调用此方法释放由此 Window、其子组件及其拥有的所有子组件所使用的所有本机屏幕资源,但不会销毁类中的成员,在其他类中调用 setVisible(true) 方法,可恢复该窗口的显示。

### 3.3.3 Register.java

Register 类中定义内部类 RFrame,RFrame 类继承 JFrame,绘制管理员注册界面,执行数据库中表 admin 插入操作。

#### RFrame 成员函数

- +RFrame() 构造函数,调用 setWindows() 和 EventsListener()
- +EventsListener:void 事件监听函数
- +setWindows:void 窗口绘制函数

- +SQL\_keyCheck:boolean SQL 操作函数，检查 admin 表中是否已经存在用户输入的用户名，存在返回 true，反之返回 false。
- +SQL\_adminInsert:void SQL 操作函数，当注册用户名和密码有效时，将此条记录插入 admin 表中。

流程图 注册模块流程图如下所示。

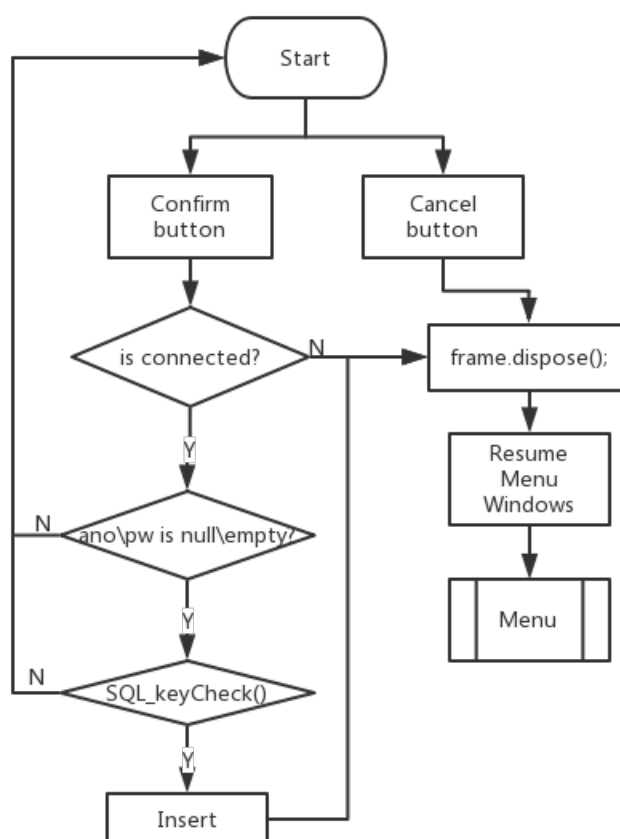


图 3.3

## 数据库处理

- SQL\_keyCheck() 方法：检查用户输入 ano 是否已存在于 admin 表中，若存在返回 true，反之 false。

```

1 public boolean SQL_keyCheck() {
2     String ano = anoField.getText().trim();
3     String sql = "select * from admin where ano = ?";
4     try {

```

```

5      PreparedStatement pStm = LogIn.conn.prepareStatement(sql);
6      pStm.setString(1, ano);
7      ResultSet rs = pStm.executeQuery();
8      if(rs.next())          return true;
9      else return false;
10 }catch (SQLException e) {
11     e.printStackTrace();
12     JOptionPane.showMessageDialog(frame,
13         "Checktime error",
14         "Error", JOptionPane.ERROR_MESSAGE);
15     return true;
16 }
17 }

```

- SQL\_adminInsert() 方法: 检查用户输入 ano 和 pw 是否为空或"", 为空调用 showMessageDialog 方法显示相关信息; 检查 ano 是否已存在, 若存在提示用户“用户已存在”, 当 ano 和 pw 均有效时, 使用 PreparedStatement 接口, 插入数据库。伪代码如下

```

1 String sql = "insert into admin values(?,?,?,?)";
2 String ano = getText();
3 String pw = getText();
4 if(check null or empty)
5     showMessageDialog("Username, Password and Name can't be empty!");
6 else if(SQL_keyCheck())
7     showMessageDialog("Username already exists");
8 else
9     try{
10         PreparedStatement pStm = LogIn.conn.prepareStatement(sql);
11         pStm.setString(1, ano);
12         pStm.setString(2, pw);
13         pStm.setString(3, name);
14         pStm.setString(4, tel);
15         pStm.executeUpdate();
16         showMessageDialog("Account created successfully");
17         frame.dispose();
18         new LogIn().new LFrame().frame.setVisible(true);

```

```

19  }catch (SQLException e){
20      showMessageDialog("Opraetion failed");
21  }
    
```

当数据成功插入后,调用 `frame.dispose()` 方法,释放窗口资源,调用 `setVisible(true)` 重新显示登陆界面。

### 3.3.4 Menu.java

Menu 类定义内部类 MFrame,MFrame 类继承 JFrame, 绘制选择界面, 可进入查询、入库、借书处理、借还书子流程。

#### MFrame 成员函数

- +MFrame(): 构造函数, 调用 `setWindows()` 和 `EventsListener()`
- +EventsListener():void 事件监听函数
- +setWindows():void 窗口绘制函数

流程图 目录模块流程图如下所示。

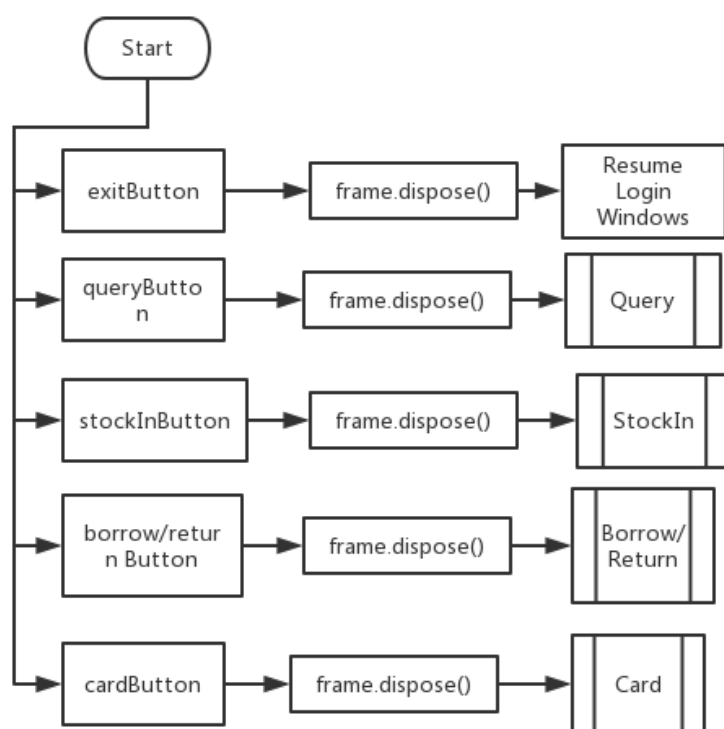


图 3.4

### 3.3.5 StockIn.java

StockIn 类中定义内部类 SIFrame, 继承 JFrame, 绘制图书入库窗口, 同时定义成员函数处理 book 表插入操作。

- 单本入库: 用户输入入库书籍各个字段信息, 均不能为空。对于 bno(bno) 已存在的书籍入库, 仅对总库存和当前库存加一。
- 批量入库: 读取指定位置文件, 对记录进行分词, 提取字段信息, 创建批量处理, 通过对数据库的一次调用提交他们, 提高性能; 将批量提交定义为一次事务, 通过 commit 方法提交数据库, 出错则 rollback。

#### SIFrame 成员函数

- +SIFrame(): 构造函数, 调用 setWindows() 和 EventsListener()
- +EventsListener():void 事件监听函数
- +setWindows():void 窗口绘制函数
- +checkEmpty():boolean 单本入库时, 检查用户输入各个字段是否为空, 全不为空, 返回 true, 反之返回 false。
- +SQL\_singInsert():void 调用数据库, 执行单本入库操作。
- +SQL\_check\_bno():boolean 调用数据库, 检查用户输入 bno(书号) 是否已存在, 存在返回 true, 不存在返回 false。
- +SQL\_addStock():void 对于已经存在的 bno(书号), 入库仅对总库存和库存加一, 默认其他信息无误。
- +SQL\_batInsert():void 读取指定位置文件, 调用数据库执行批量入库操作。

流程图 图书入库模块流程图如下所示。

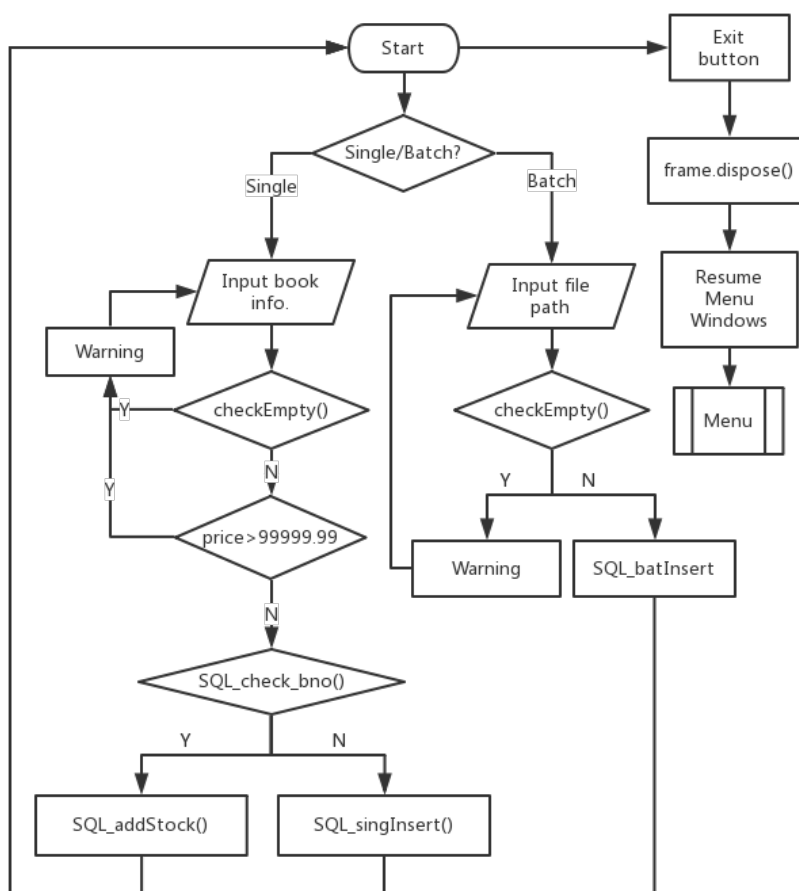


图 3.5

## 数据库处理

- SQL\_check\_bno() 根据用户输入 bno 进行简单查询，检测结果集是否为空即可。SQL 语句如下

```
1 select * from book where bno = @bno;
```

- SQL\_addStock() 调用数据库将某书总库存和当前库存加一。SQL 语句如下

```
1 update book set stock = stock + 1, total = total + 1
2 where bno = @bno;
```

- SQL\_singInsert() 调用数据库执行单本插入。使用 PreparedStatement 接口传值。SQL 语句为

```
1 String sql = "insert into book values(?,?,?,?,?,?,?,?,?)";
```

值得注意的是，数据库中 book 表的 price 字段类型为 decimal(7,2)，所以用户输入的字符串如下转换。

```
1 float price = (float)(Math.round(Float.parseFloat(pri)*100))/100;
```

最后调用 executeUpdate() 提交执行。

- SQL\_batInsert() 通过一次调用数据库，插入多条数据。先从指定路径打开文件，对文件进行读行操作，对于每一行，通过定义 java.util 下的 StringTokenizer 类对每一行进行分词处理，得到每一个 token 储存在 token 数组中

```
1 StringTokenizer st = new StringTokenizer(line, "(, )");
2 int i = 0;
3 while(st.hasMoreElements()) {
4     token.set(i++, st.nextToken());
5 }
```

对于每一行，通过 PreparedStatement 对象调用 addBatch()，将插入改行的数据库语句放入 pStm 的命令列表中，最后调用 executeBatch() 和 commit() 接口提交数据库执行，伪代码如下

```
1 while((line = reader.readLine()) != null){
2     getToken();
3     LogIn.conn.setAutoCommit(false);
4     pStm.addBatch();
5 }
6 pStm.executeBatch();
7 LogIn.conn.commit();
8 LogIn.conn.setAutoCommit(true);
```

若调用数据库发生错误，则通过 rollback() 方法滚回 transaction 执行前。

```
1 catch (SQLException e2) {
2     e2.printStackTrace();
3     try {
4         LogIn.conn.rollback();
5     } catch (SQLException e) {
6         e.printStackTrace();
7     }
8 }
```

### 3.3.6 Card.java

Card 类中定义内部类 CFrame 类，继承了 JFrame，绘制借书证管理界面，执行 card 表的插入删除操作。该窗口会动态显示当前所有借书证。

#### CFrame 成员函数

- +CFrame(): 构造函数，调用 setWindows() 和 EventsListener()
- +EventsListener():void 事件监听函数
- +setWindows():void 窗口绘制函数
- +SQL\_insert():void 向 card 表插入一条借书证记录。
- +SQL\_del():void 删除 card 表中的一条记录
- +SQL\_keyCheck():boolean 检查 card 表中是否已存在用户希望删除或创建的 cno(借书证号)
- +SQL\_cardInfo():void 查询 card 表全部信息，传递给二维数组，该二维数组作为参数传递给 table 类对象绘制表格。



流程图 借书证管理模块流程图如下所示。

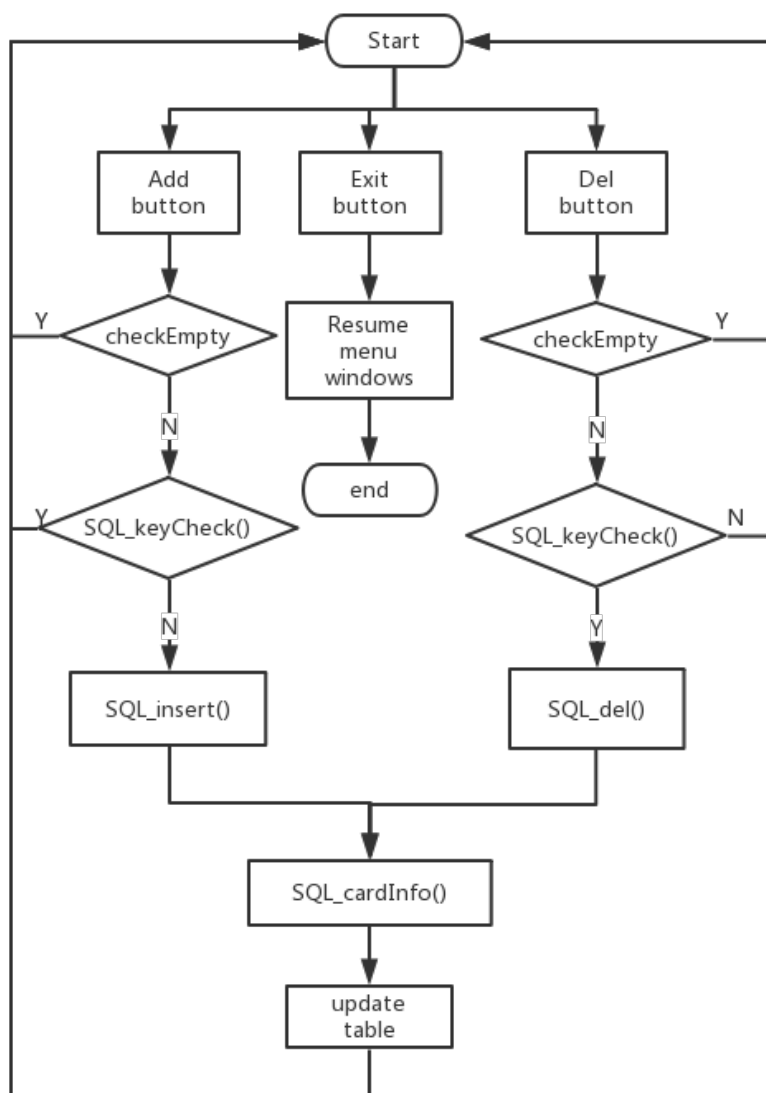


图 3.6

数据库处理 调用数据库执行删除和插入的语句如下

- 插入 card 表: insert into card values(?,?,?,?)
- 删除一条记录: delete from card where cno = ?

### 3.3.7 Query.java

Query 类中定义内部类 QFrame,QFrame 继承 JFrame, 绘制查询界面, 实现了

- 对指定字段信息查询
- 对字段区间查询
- 已指定字段排序显示
- 升序或降序显示

### **QFrame 成员函数**

- +QFrame(): 构造函数，调用 setWindows() 和 EventsListener()
- +EventListener():void 事件监听函数
- +setWindows():void 窗口绘制函数
- +SQL\_\_defaultInfo():void 初始化对象时，调用该方法获取 book 表中信息，构建表格，并以书名排序。
- +SQL\_\_queryInfo():void 根据用户输入字段或字段区间或排序方式进行查询，获取结果集在表格中更新。

流程图 查询模块流程图如下所示。

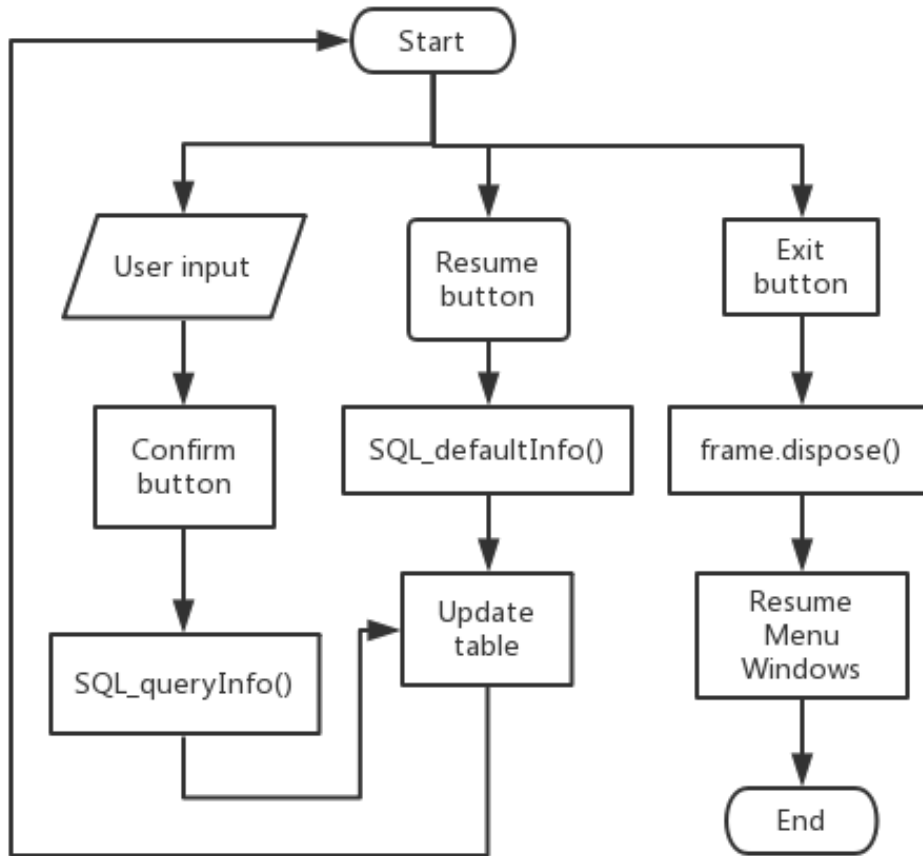


图 3.7

## 数据库处理

- SQL\_defaultInfo() 方法只需进行默认查询即可，SQL 语句如下

```
1 select * from book order by title
```

- SQL\_queryInfo() 方法调用数据库执行以下语句

```
1 String sql = "select * from book where "
2   + qItem + " between ? and ? order by "
3   + sBox.getSelectedItem().toString().trim() + " "
4   + oBox.getSelectedItem().toString().trim();
```

其中

- qItem 是查询字段。
- 问号处传递变量指定查询范围。若用户不进行范围查询，两个问号传递值相同
- sBox 为下拉选框，选择以哪个字段排序
- oBox 为下拉选框，选择升序或者降序

经过查询后遍历结果集，将其传递给构造 table 对象的二维数组，调用 updateUI() 方法更新 table。

### 3.3.8 Bor\_Ret.java

Bor\_Ret 类中定义了内部类 BRFrame，继承 JFrame，实现

- 还书
- 借书，同时自定义借阅时间
- 查询某借书证已借书籍，以列表显示
- 若无库存，显示最近的归还时间

#### BRFrame 成员函数

- +QFrame(): 构造函数，调用 setWindows() 和 EventsListener()
- +EventsListener():void 事件监听函数
- +setWindows():void 窗口绘制函数
- +SQL\_bookInfo():void 对象初始化时调用一次，显示图书馆内当前所有书籍
- +SQL\_borrowInfo():void 根据 cno(借书证号) 查询已借书籍，更新表格
- +SQL\_borrow():void 执行一系列检查，满足 cno 和 bno 非空；cno 已注册；书籍有库存；当前未借阅等条件才会调用 SQL\_bTrans() 方法，进行事务处理，同步更新 stock 和 borrow 表。
- +SQL\_return():void 执行一系列检查，满足满足 cno 和 bno 非空；cno 已注册；bno 存在；已借阅等条件才会调用 SQL\_rTrans() 方法，进行事务处理，同步更新 stock 和 borrow 表。
- +SQL\_cnoCheck():boolean 查询 cno 是否已存在

- +SQL\_bnoCheck():boolean 查询 bno 是否存在
- +SQL\_hasStock():boolean 查询某书是否有库存
- +SQL\_isBorrow():boolean 查询某借书证是否已借阅某书
- +SQL\_returnTime():String 当某书没有库存时，查询并显示最近的归还时间
- +SQL\_bTrans():void 调用数据库执行事务处理，保证 book 表与 borrow 表更新的同步性。
- +SQL\_rTrans():void 作用类似。

流程图 借书还书模块如下所示

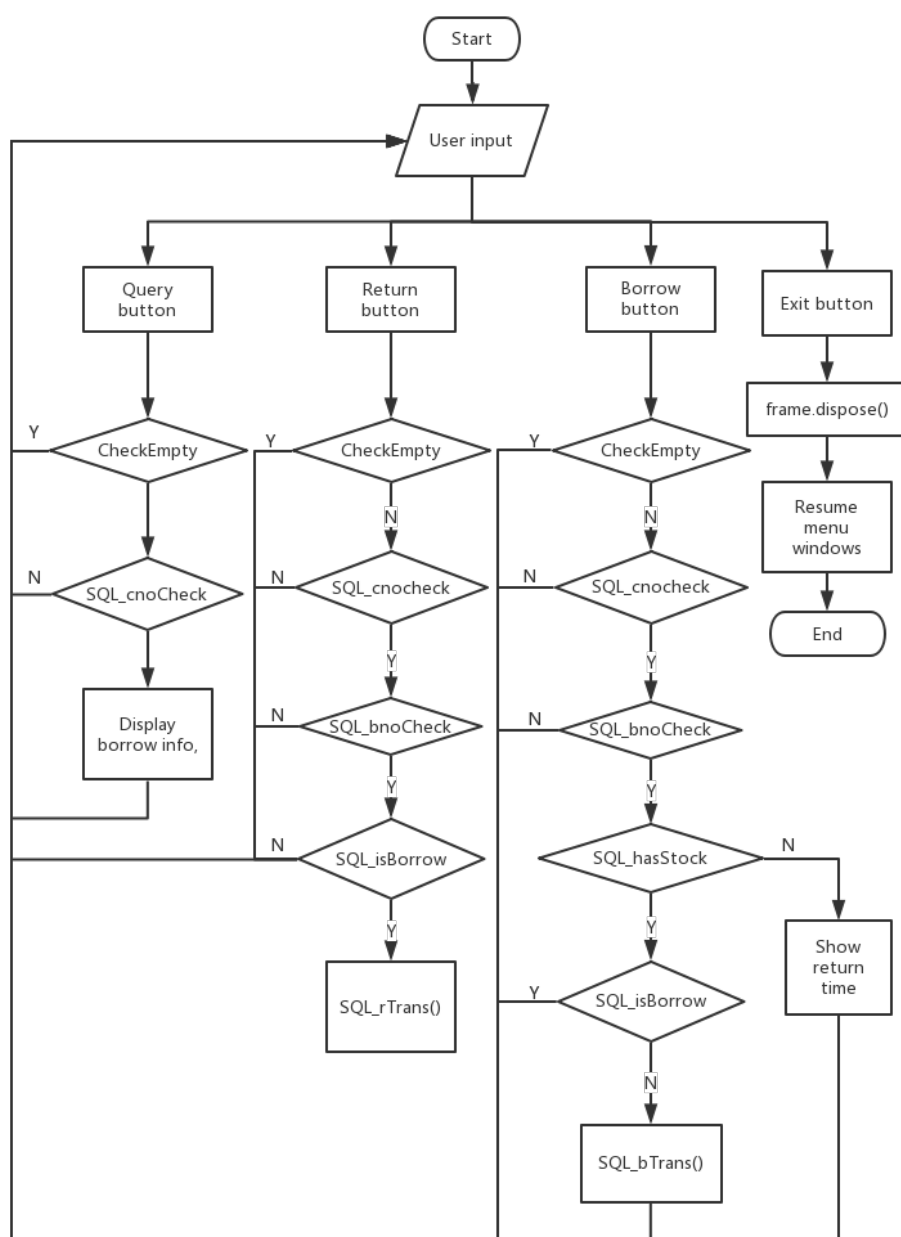


图 3.8

数据库操作 调用数据库的相关操作大部分与之前模块相似，需要提及的以下三点

- 借书还书事务处理，保证 borrow 表和 book 表同步更新。首先调用方法 `setSavepoint()` 方法创建储存点

```

1 LogIn.conn.setAutoCommit(false);
2 savepoint1 = LogIn.conn.setSavepoint("BEFORE");
    
```

然后依次执行 borrow 表和 book 表更新操作，此时自动提交已经关闭，需要调用 commit() 方法才会提交数据库

```
1 ...
2 pstmt1.executeUpdate();
3 ...
4 pstmt2.executeUpdate();
5 LogIn.conn.commit();
6 ...
```

在执行完两条语句后，调用 commit() 方法一同提交，至此事务结束。若该过程发生错误，调用 rollback() 方法，回到 savepoint。

```
1 catch (SQLException e2) {
2     e2.printStackTrace();
3     JOptionPane.showMessageDialog(frame,
4         "BorrowTransaction failed!",
5         "Error", JOptionPane.ERROR_MESSAGE);
6     try {
7         LogIn.conn.rollback(savepoint1);
8     } catch (SQLException e3) {
9         e3.printStackTrace();
10    }
11 }
```

- 自定义借阅时间，由用户输入数值和选择单位 (天、周、月)，在传递参数之前计算 timespan

```
1 if( timeField.getText().trim() != null
2 && !timeField.getText().trim().equals("")) {
3     int n = Integer.parseInt(timeField.getText().trim());
4     if(tBox.getSelectedItem().toString().equals("day(s)"))
5         timespan = n * day;
6     else if(tBox.getSelectedItem().toString().equals("week(s)"))
7         timespan = n * week;
8     else    timespan = n * month;
9 }
```

通过 System.currentTimeMillis() 方法获取当前系统时间毫秒数，调用 MyDate 类方法 dateTrans() 将长整型的毫秒数转化为日期格式。

```

1 java.sql.Date fDate = java.sql.Date.valueOf(
2   MyDate.dateTrans(System.currentTimeMillis()));
3 java.sql.Date tDate = java.sql.Date.valueOf(
4   MyDate.dateTrans(System.currentTimeMillis()+timespan));

```

之后将 fDate 与 tDate 传入 SQL 语句即可。

- 获取最近的归还时间，首先调用数据库查询对应书号的所有应还时间，并以升序向下排列；同时定义 String now 记录当前时间

```

1 String sql = "select distinct return_date from borrow
2             where bno = ? order by return_date ASC";
3 String now = MyDate.dateTrans(System.currentTimeMillis());

```

执行数据库查询后得到结果集，从前往后 (从小到大) 依次遍历，调用 String 类下的 compareTo 方法比较两字符串大小，由于日期都已格式化，比较很方便。若找到第一个大于当前日期的归还日期，该日期即为所求；反之，所有归还时间均已过期，返回最大的日期，同时在结尾加上“(expired)”。

```

1 String d = "None";
2 while(iter.hasNext()) {
3   d = iter.next();
4   if(d.compareTo(now) >= 0) {
5     break;
6   }
7   if(!iter.hasNext()) {
8     d += "(expired)";
9     break;
10  }
11 }
12 return d;

```

### 3.3.9 MaDate.java

MyDate 类中定义静态方法 dateTrans()，将调用 System.currentTimeMillis() 方法得到的时间转换为格式为 yyyy-MM-dd 的日期，在 BRFrame 中被调用。

```

1 public static String dateTrans(long t) {
2   SimpleDateFormat df = (SimpleDateFormat)DateFormat.getInstance();
3   df.applyPattern("yyyy-MM-dd");

```



```
4   String date = df.format(t);  
5   return date;  
6 }
```

## 第 4 章 运行结果截图

### 4.1 管理员登陆

点击 Link DB 连接数据库



图 4.1

以管理员登陆，输入账号密码；此时，cno 域无法编辑

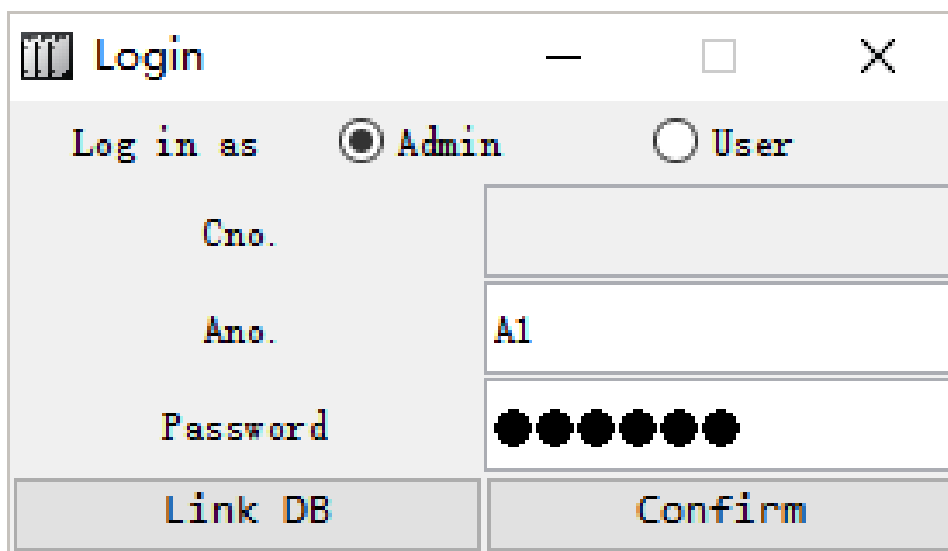


图 4.2

点击 Confirm，登陆成功，进入菜单

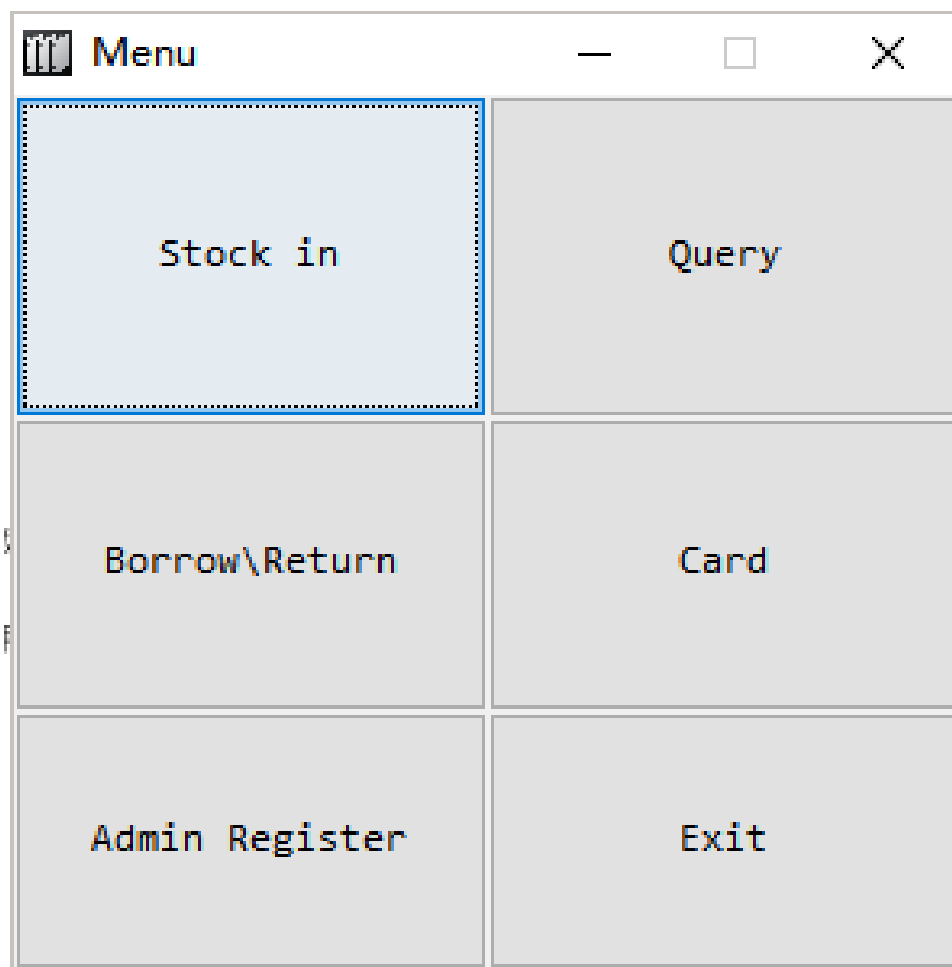


图 4.3

#### 4.1.1 图书入库

点击 Stock in 进入图书入库界面，选择 Single Mode 单本入库，同时输入信息。

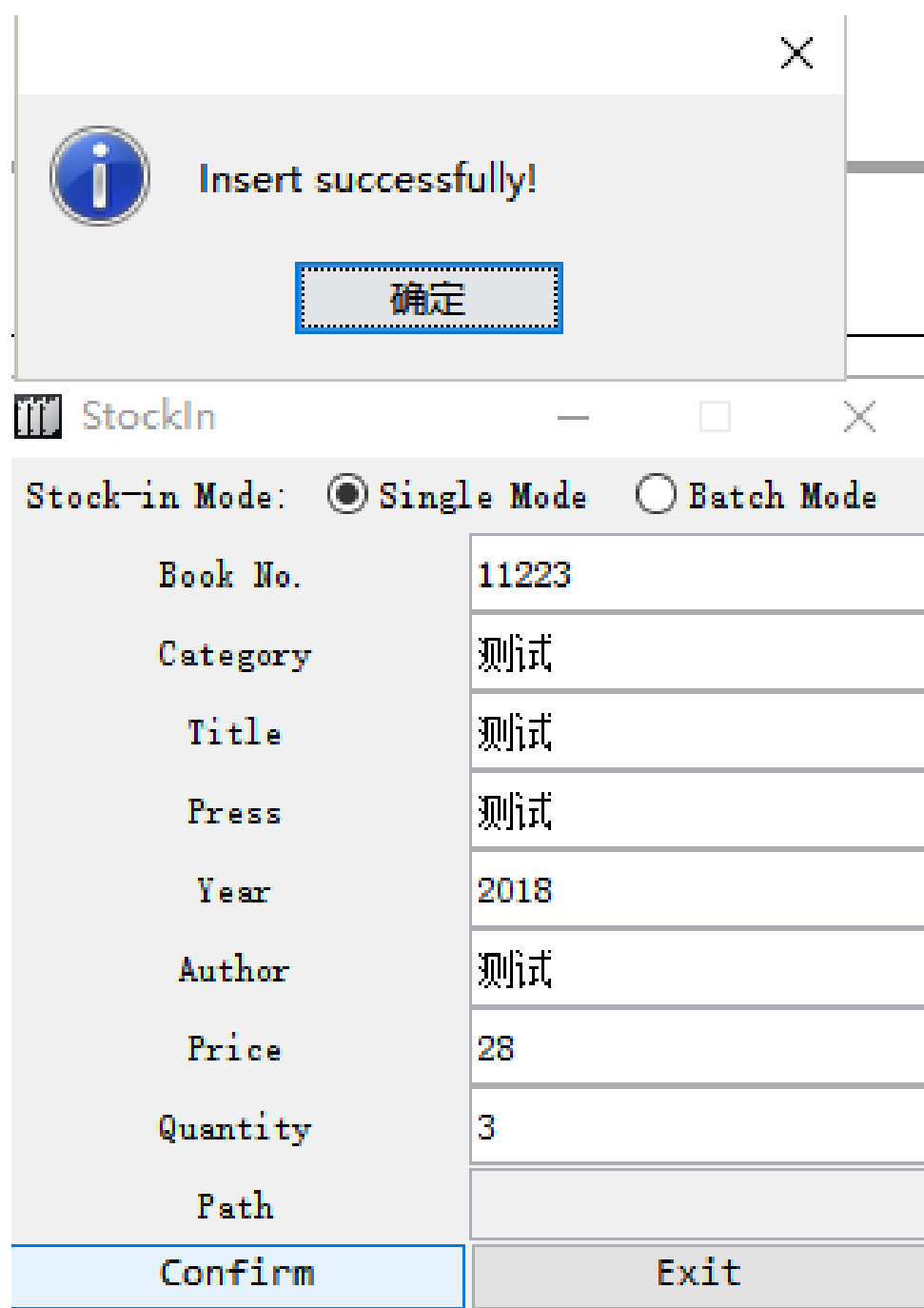
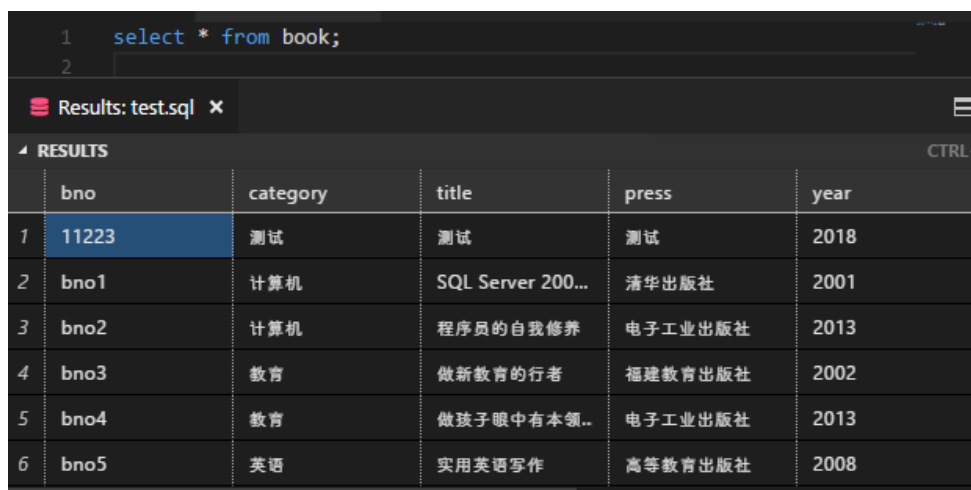


图 4.4



1 select \* from book;

2

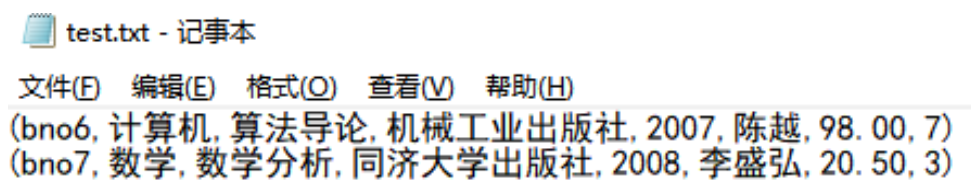
Results: test.sql x

RESULTS

	bnos	category	title	press	year
1	11223	测试	测试	测试	2018
2	bnos1	计算机	SQL Server 200...	清华大学出版社	2001
3	bnos2	计算机	程序员的自我修养	电子工业出版社	2013
4	bnos3	教育	做新教育的行者	福建教育出版社	2002
5	bnos4	教育	做孩子眼中有本领..	电子工业出版社	2013
6	bnos5	英语	实用英语写作	高等教育出版社	2008

图 4.5

选择 Batch Mode 批量入库，不做任何输入，从默认路径导入



test.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

(bnos6, 计算机, 算法导论, 机械工业出版社, 2007, 陈越, 98.00, 7)

(bnos7, 数学, 数学分析, 同济大学出版社, 2008, 李盛弘, 20.50, 3)

图 4.6

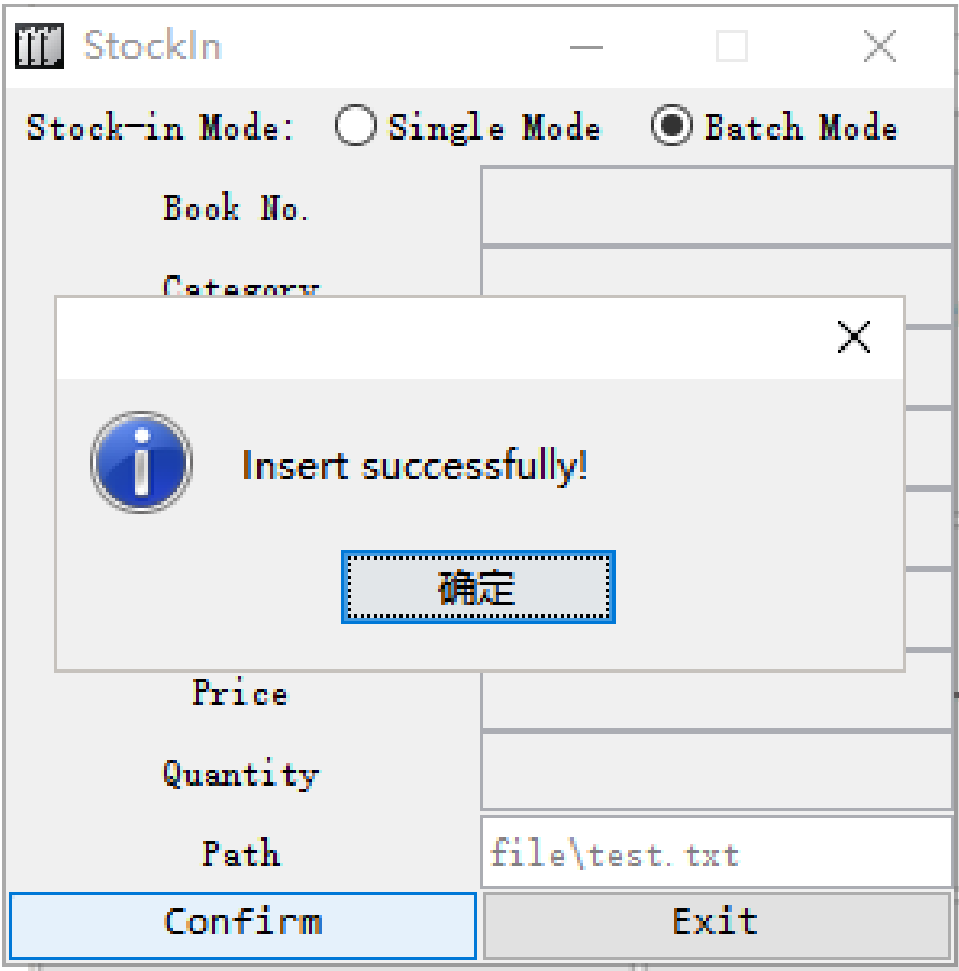


图 4.7

```
1 select * from book;
```

Results: test.sql					
RESULTS					
	bno	category	title	press	year
1	11223	测试	测试	测试	2018
2	bno1	计算机	SQL Server 200...	清华大学出版社	2001
3	bno2	计算机	程序员的自我修养	电子工业出版社	2013
4	bno3	教育	做新教育的行者	福建教育出版社	2002
5	bno4	教育	做孩子眼中有本领..	电子工业出版社	2013
6	bno5	英语	实用英语写作	高等教育出版社	2008
7	bno6	计算机	算法导论	机械工业出版社	2007
8	bno7	数学	数学分析	同济大学出版社	2008

图 4.8

4.1.2 管理员注册

点击 admin register，输入相关信息

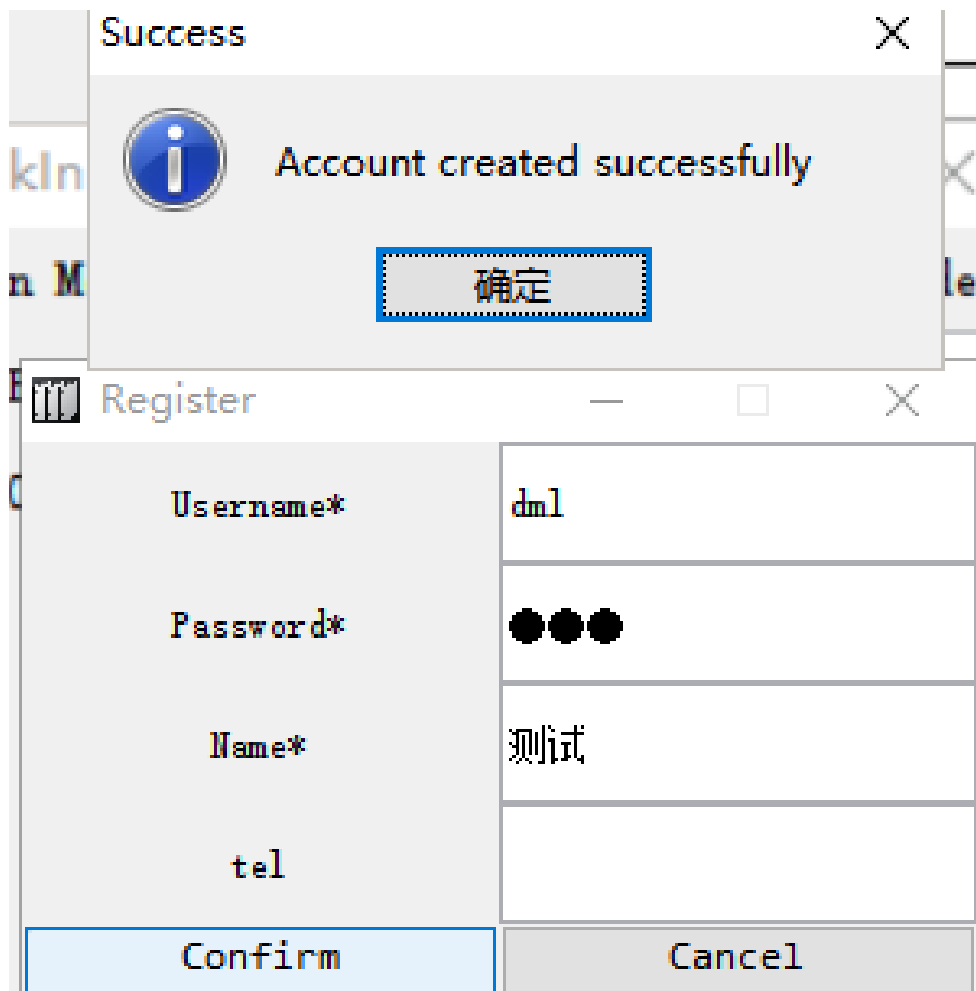


图 4.9

```
1  select * from admin;
2
```

Results: test.sql				
RESULTS				
	ano	pw	name	tel
1	A1	123456	Jeff	18158515687
2	A2	123456	Mary	
3	dml	123	测试	
4	system	123	system	

图 4.10

### 4.1.3 查询

查询图书类别为‘计算机’的书籍，以出版年份、降序排列

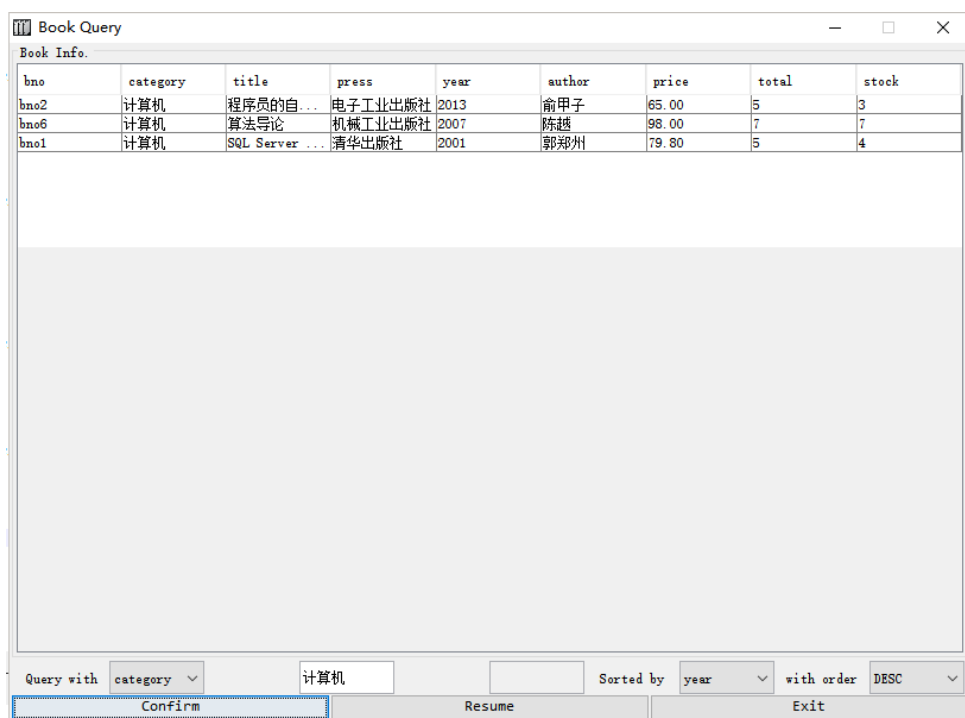


图 4.11

查询出版年份在 2002 与 2008 之间的书籍，以价格、升序排列



The screenshot shows the 'Book Query' application window. It features a table titled 'Book Info.' with the following data:

bno	category	title	press	year	author	price	total	stock
bno7	数学	数学分析	同济大学出版社	2008	李盛弘	20.50	3	3
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
bno5	英语	实用英语写作	高等教育出版社	2008	高云鹏	33.00	3	3
bno6	计算机	算法导论	机械工业出版社	2007	陈越	98.00	7	7

Below the table, there is a query filter section with the following settings:

- Query with: year span
- from: 2002
- to: 2008
- Sorted by: price
- with order: ASC

At the bottom, there are three buttons: 'Confirm', 'Resume', and 'Exit'.

图 4.12

查询价格在 25 与 33.8 之间的书籍，以库存、升序排列

The screenshot shows the 'Book Query' application window after a query. The table 'Book Info.' now displays only the books that match the criteria (price between 25 and 33.8, sorted by stock):

bno	category	title	press	year	author	price	total	stock
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
bno5	英语	实用英语写作	高等教育出版社	2008	高云鹏	33.00	3	3
11223	测试	测试	测试	2018	测试	28.00	3	3

The query filter section at the bottom is updated with the following settings:

- Query with: price
- from: 25
- to: 33.8
- Sorted by: stock
- with order: ASC

The 'Confirm' button is highlighted in blue.

图 4.13

#### 4.1.4 借还书

查询 cno1 的借书情况

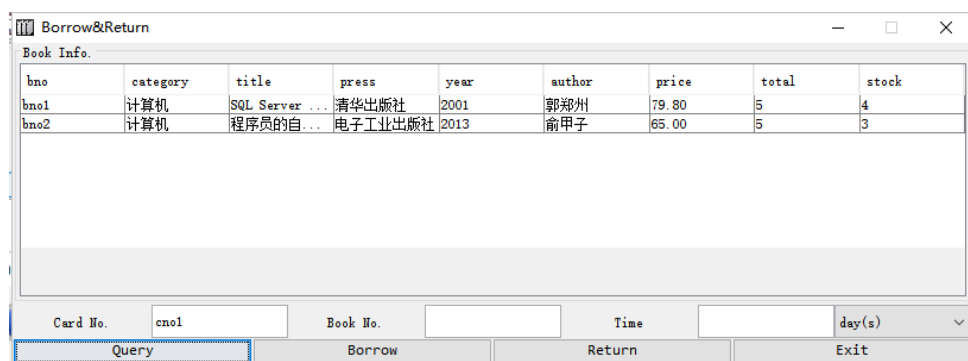


图 4.14

借 bno3 这本书，借 3 周

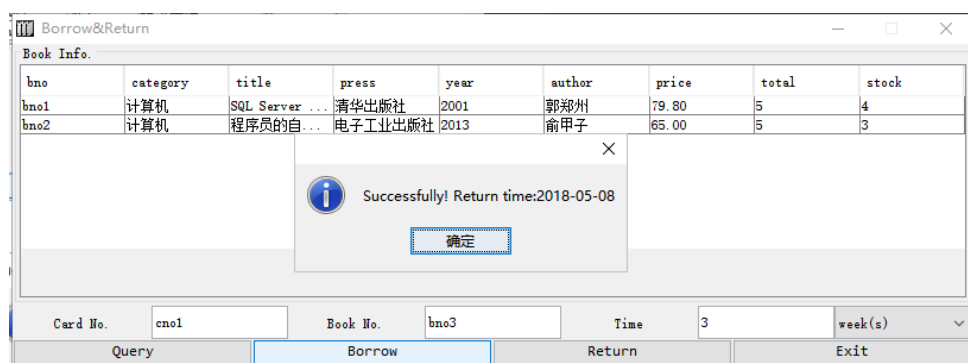


图 4.15

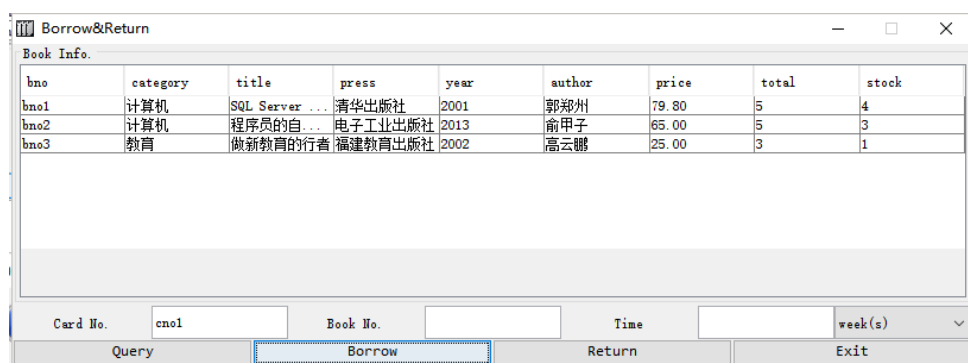


图 4.16

在数据库中查询 borrow 表中的经手人与 book 表的库存

1 select \* from borrow;  
2 select \* from book;

Results: test.sql

RESULTS

	cno	bno	borrow_date	return_date	ano
1	cno1	bno1	2010-06-04	2010-06-10	A1
2	cno1	bno2	2010-06-05	2010-06-10	A1
3	cno2	bno2	2010-07-04	2010-07-10	A1
4	cno3	bno3	2010-08-04	2010-08-10	A2
5	cno4	bno4	2010-09-04	2010-09-10	A2
6	cno1	bno3	2018-04-17	2018-05-08	A1

	bno	cat	tit...	pre	y...	...	price	total	stock
1	11223	测	测...	测	2...	...	28.00	3	3
2	bno1	计	S...	清	2...	...	79.80	5	4
3	bno2	计	程...	电	2...	...	65.00	5	3
4	bno3	教	做...	福	2...	...	25.00	3	1
5	bno4	教	做...	电	2...	...	23.00	5	4
6	bno5	英	实...	高	2...	...	33.00	3	3
7	bno6	计	算...	机	2...	...	98.00	7	7
8	bno7	数	数...	同	2...	...	20.50	3	3

图 4.17

cno1 再借这本书

Borrow&Return

Book Info.

bno	category	title	press	year	author	price	total	stock
bno1	计算机	SQL Server ...	清华大学出版社	2001	郭郑州	79.80	5	4
bno2	计算机	程序员的自...	Failed			65.00	5	3
bno3	教育	做新教育的行者				25.00	3	1

You have borrowed this book!

确定

Card No. cno1 Book No. bno3 Time week(s) v

Query Borrow Return Exit

图 4.18

让 cno2 再借这本书，借 1 天，这本书库存为 0.

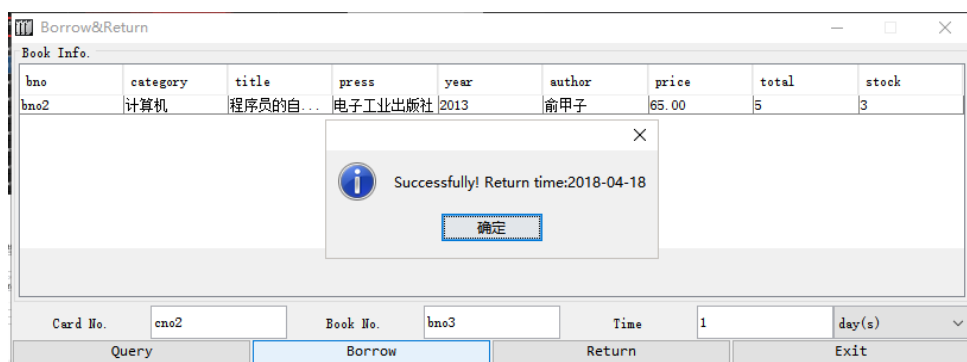


图 4.19

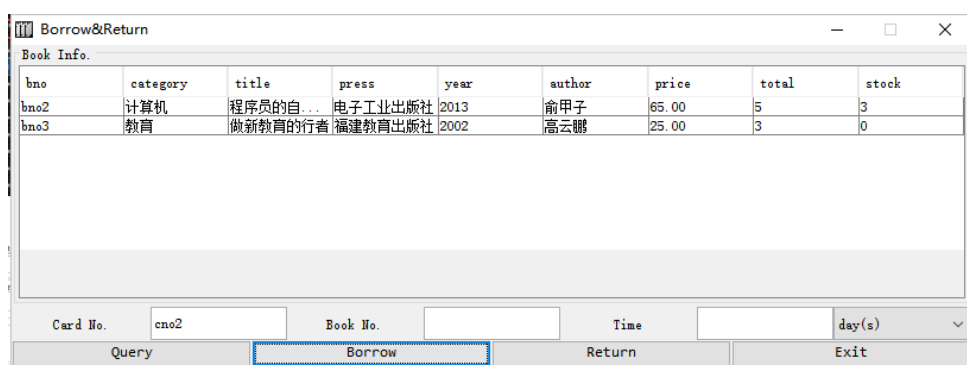


图 4.20

再然 cno4 借这本书，显示了最近的还书日期。

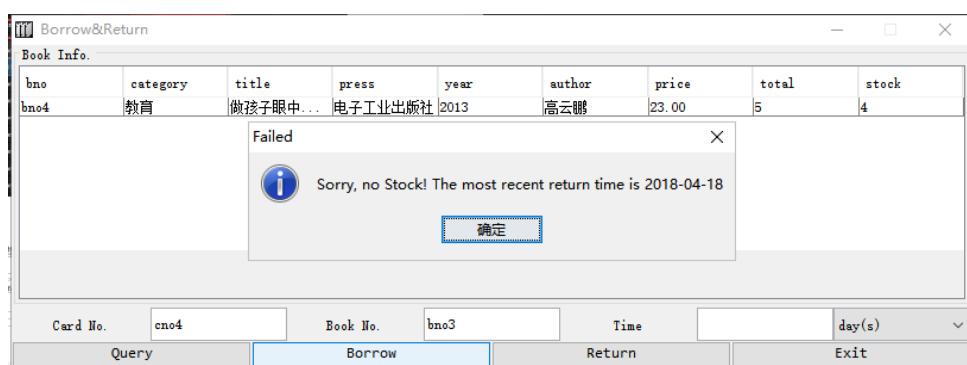


图 4.21

#### 4.1.5 借书证管理

输入 cno = 1111

Card Register

Card Info.

cno	name	department	type
cno1	张三	计算机学院	U
cno2	李四	农学院	U
cno3	王五	计算机学院	T
cno4	朱六	计算机学院	G
cno5	延七	经济学院	O
cno6	凤姐	经济学院	O

Card Type

☐ T ☒ G ☐ U ☐ O

Card No.

1111

Name

测试

Department

测试


Add

Delete

Exit

图 4.22

Success



Card created successfully

确定

图 4.23

Card Register

Card Info.

cno	name	department	type
1111	测试	测试	G
cno1	张三	计算机学院	U
cno2	李四	农学院	U
cno3	王五	计算机学院	T
cno4	朱六	计算机学院	G
cno5	延七	经济学院	O

Card Type

☐ T ☒ G ☐ U ☐ O

Card No.

Name

Department

Add

Delete

Exit

图 4.24

删除这张卡

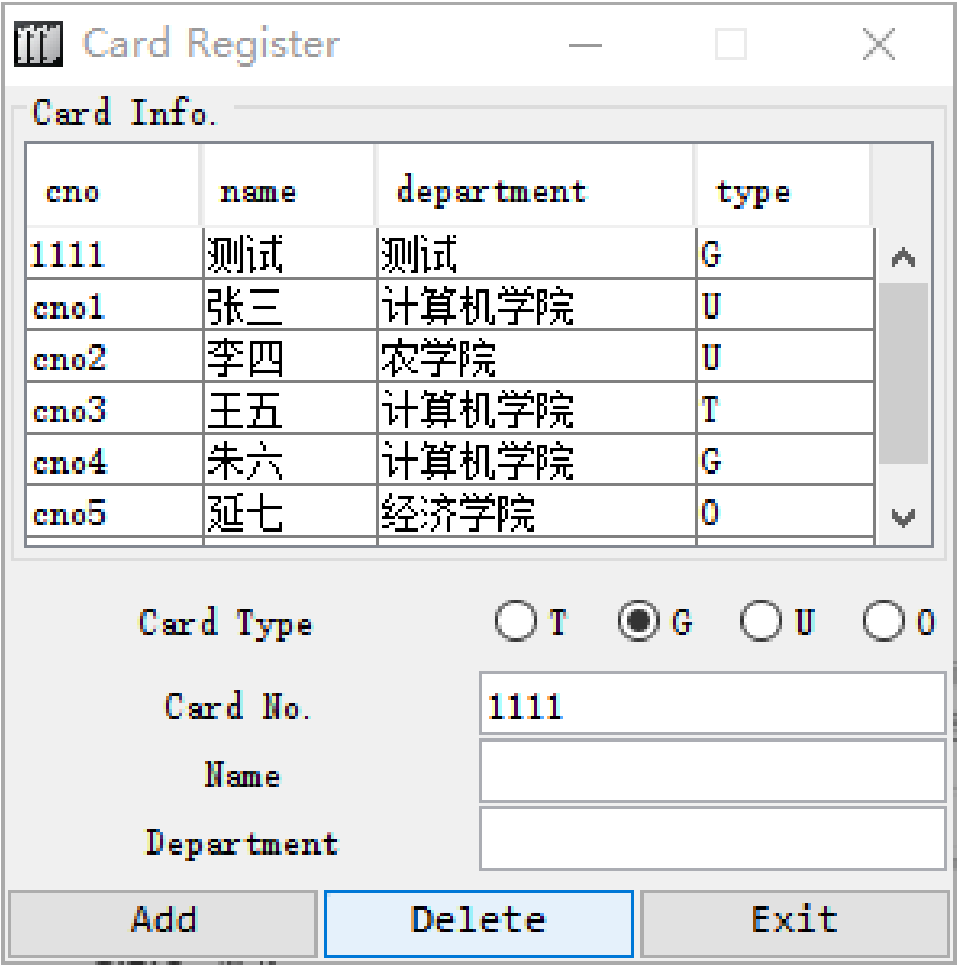


图 4.25

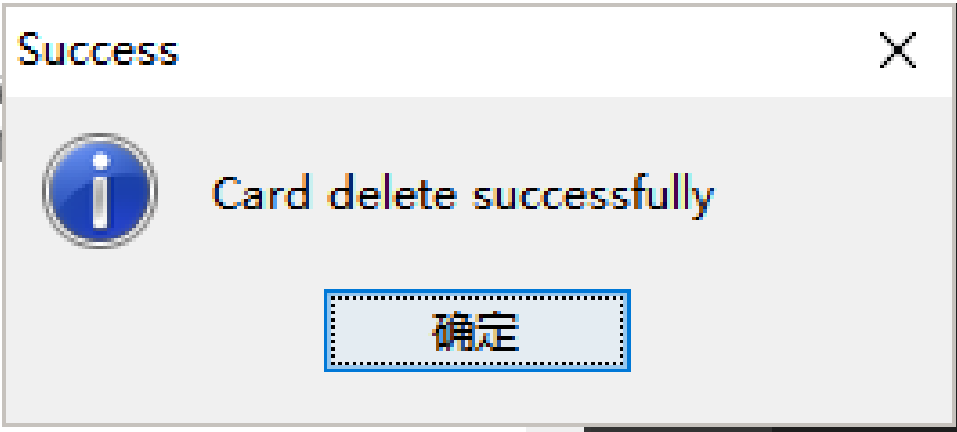


图 4.26

Card Register

Card Info.

cno	name	department	type
cno1	张三	计算机学院	U
cno2	李四	农学院	U
cno3	王五	计算机学院	T
cno4	朱六	计算机学院	G
cno5	延七	经济学院	O
cno6	凤姐	经济学院	O

Card Type

☐ T ☒ G ☐ U ☐ O

Card No.

Name

Department

Add

Delete

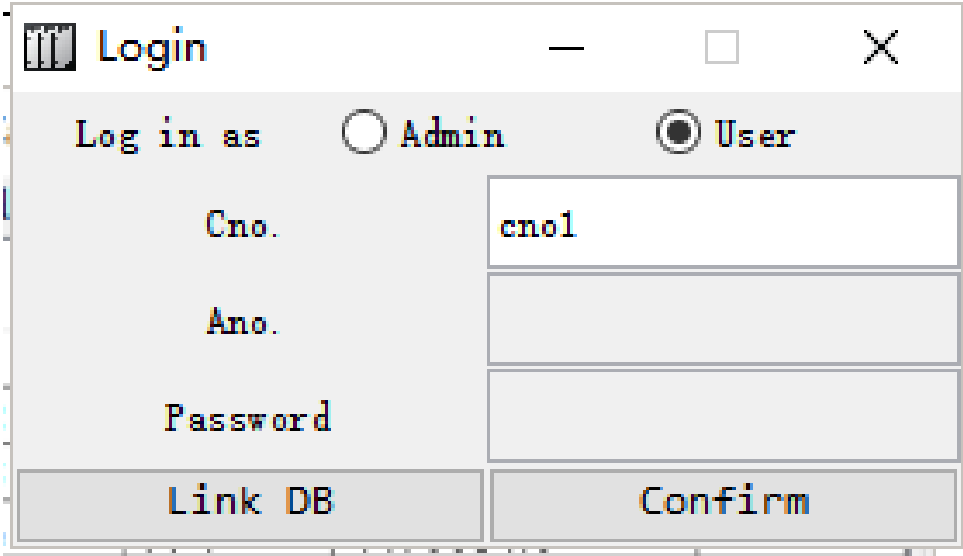
Exit

图 4.27

4.2 普通用户登陆

普通用户以借书证号登陆

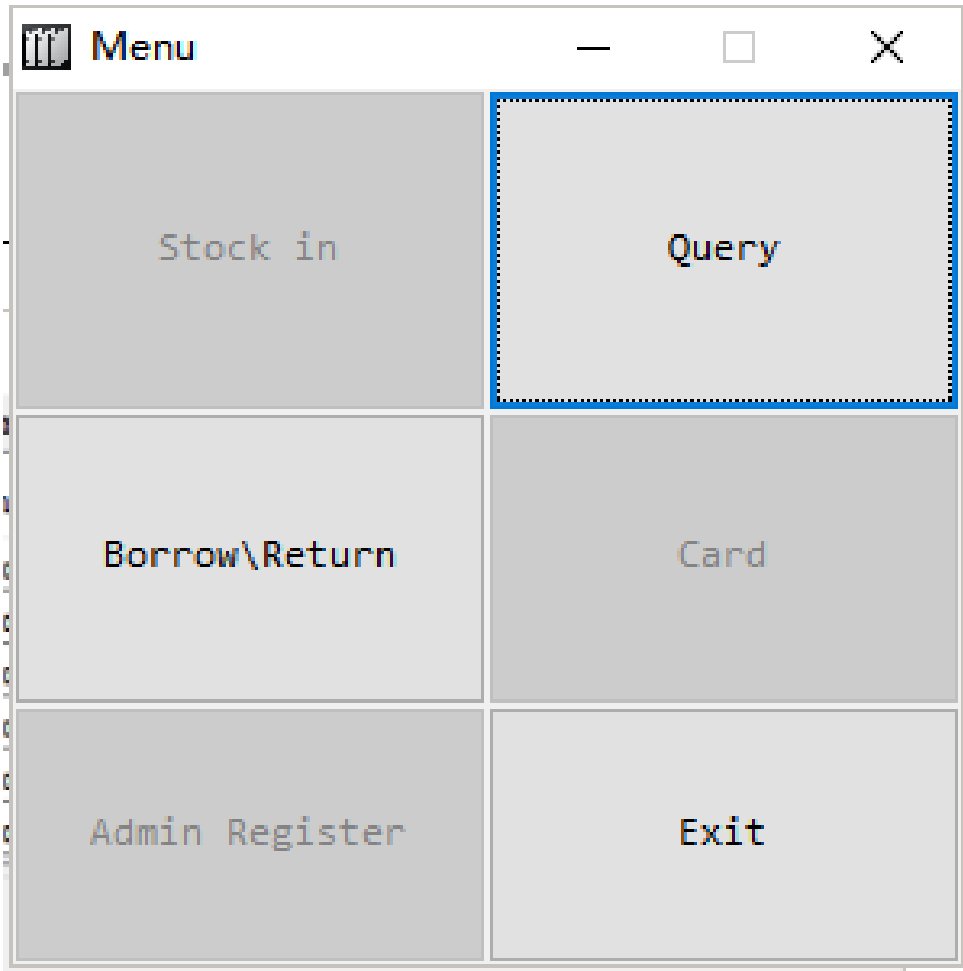




The image shows a 'Login' window with a title bar containing a window icon, the text 'Login', and standard window controls (minimize, maximize, close). The window has a light gray background. At the top, it says 'Log in as' followed by two radio buttons: 'Admin' (unselected) and 'User' (selected). Below this, there are three input fields. The first is labeled 'Cno.' and contains the text 'cnol'. The second is labeled 'Ano.' and is empty. The third is labeled 'Password' and is empty. At the bottom, there are two buttons: 'Link DB' on the left and 'Confirm' on the right.

图 4.28

登录后菜单如下



The image shows a 'Menu' window with a title bar containing a window icon, the text 'Menu', and standard window controls (minimize, maximize, close). The window has a light gray background and contains a 3x2 grid of buttons. The buttons are labeled: 'Stock in' (top-left), 'Query' (top-right), 'Borrow\Return' (middle-left), 'Card' (middle-right), 'Admin Register' (bottom-left), and 'Exit' (bottom-right). The 'Query' button is highlighted with a blue dashed border.

图 4.29

### 4.2.1 借还书

功能与上述相同，但是无法对 cno 进行更改，只能对自己这张卡借书还书

bno	category	title	press	year	author	price	total	stock
bn01	计算机	SQL Server ...	清华大学出版社	2001	郭郑州	79.80	5	4
11223	测试	测试	测试	2018	测试	28.00	3	3
bn02	计算机	程序员的自...	电子工业出版社	2013	俞甲子	65.00	5	3
bn05	英语	实用英语写作	高等教育出版社	2008	高云鹏	33.00	3	3
bn07	数学	数学分析	同济大学出版社	2008	李盛弘	20.50	3	3
bn06	计算机	算法导论	机械工业出版社	2007	陈越	98.00	7	7
bn04	教育	做孩子眼中...	电子工业出版社	2013	高云鹏	23.00	5	4
bn03	教育	做新教育的行者	福建教育出版社	2002	高云鹏	25.00	3	2

Card No.  Book No.  Time  day(s)

图 4.30

### 4.2.2 查询

与上述相同，在此省略

## A SQL Source Code

### lib\_init.sql

```
1 create database LMS;
2 go
3 use LMS;
4
5 create table book
6 (
7 bno char(8) primary key,
8 category char(10),
9 title varchar(40),
10 press varchar(30),
11 year int,
12 author varchar(20),
13 price decimal(7,2),
14 total int,
15 stock int
16 );
17 create table card
18 (
19 cno char(7) not null primary key,
20 name varchar(10) not null,
21 department varchar(40) not null,
22 type char(1) check(type in ('T', 'G', 'U', 'O'))
23 );
24 create table admin
25 (
26 ano varchar(10) not null,
27 pw varchar(20) not null,
28 name varchar(10) not null,
29 tel char(11) null,
30 primary key(ano)
31 );
32 create table borrow
33 (
```

```

34 cno char(7),
35 bno char(8),
36 borrow_date date,
37 return_date date,
38 ano varchar(10),
39 foreign key(cno) references card
40     on update cascade,
41 foreign key(bno) references book
42     on update cascade
43     on delete cascade,
44 foreign key(ano) references admin
45 );
46
47 insert into book values('bno1','计算机',
48 , 'SQL Server 2008完全学习手册', '清华大学出版社', 2001, '郭郑州', 79.80, 5, 4);
49 insert into book values('bno2','计算机',
50 , '程序员的自我修养', '电子工业出版社', 2013, '俞甲子', 65.00, 5, 3);
51 insert into book values('bno3','教育',
52 , '做新教育的行者', '福建教育出版社', 2002, '高云鹏', 25.00, 3, 2);
53 insert into book values('bno4','教育',
54 , '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, 4);
55 insert into book values('bno5','英语',
56 , '实用英文写作', '高等教育出版社', 2008, '庞继贤', 33.00, 3, 3);
57
58
59 insert into card values('cno1','张三','计算机学院','U');
60 insert into card values('cno2','李四','农学院','U');
61 insert into card values('cno3','王五','计算机学院','T');
62 insert into card values('cno4','朱六','计算机学院','G');
63 insert into card values('cno5','延七','经济学院','O');
64 insert into card values('cno6','凤姐','经济学院','O');
65
66 insert into admin values('A1','123456','Jeff','18874274311');
67 insert into admin values('A2','123456','Mary','');
68 insert into admin values('system','123','system','');
69

```

```

70 insert into borrow values( 'cno1', 'bno1'
71 , '2010-6-4', '2010-6-10', 'A1' );
72 insert into borrow values( 'cno1', 'bno2'
73 , '2010-6-5', '2010-6-10', 'A1' );
74 insert into borrow values( 'cno2', 'bno2'
75 , '2010-7-4', '2010-7-10', 'A1' );
76 insert into borrow values( 'cno3', 'bno3'
77 , '2010-8-4', '2010-8-10', 'A2' );
78 insert into borrow values( 'cno4', 'bno4'
79 , '2010-9-4', '2010-9-10', 'A2' );

```

### trigger.sql

```

1 CREATE TRIGGER chec_empty ON admin
2 INSTEAD of INSERT
3 AS
4 BEGIN
5 DECLARE @ano VARCHAR(10)
6 DECLARE @pw VARCHAR(20)
7 DECLARE @name VARCHAR(10)
8 SELECT @ano = ano FROM inserted
9 SELECT @pw = pw FROM inserted
10 SELECT @name = name FROM inserted
11 IF (@ano = '' OR @ano is null
12 OR @pw = '' OR @pw is null
13 OR @name = '' OR @name is null)
14 print ( 'Insert refused ' )
15 ELSE
16 INSERT into admin SELECT * FROM inserted
17 END

```

## B Java Source Code

Omitted here.