

## 报告二：商品销售管理系统

### 第7部分 设计的内容要求

#### 7.1 设计内容

商品(图书)销售管理系统，后台连接数据库，前端 有操作友好的界面设计。

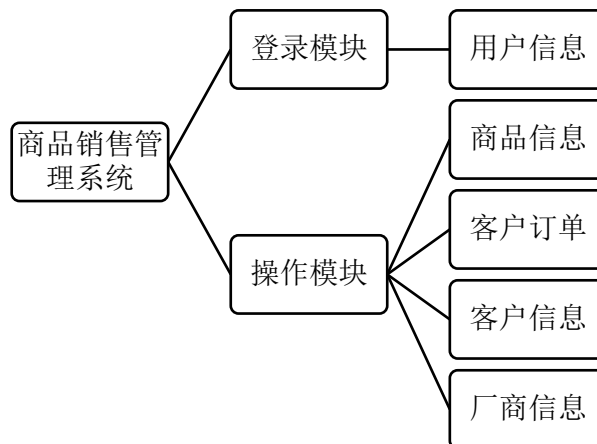
#### 7.2 设计要求

1. 系统各种信息的输入，包括商品信息、客户订单、厂商信息的输入等。
2. 商品销售管理系统各种信息的查询、修改和维护。
3. 商品销售报表的生成。
4. 用户对信息的管理。

### 第8部分 设计思想

#### 8.1 主要思想

将整个系统划分为以下几个模块



## 8.2 算法及技术

1. 登录模块主要运用了数据库的连接，字符串的读取等技术。
2. 操作模块主要运用了数据库的连接，SQL 语句对数据库的操作等技术。

## 第9部分 详细设计

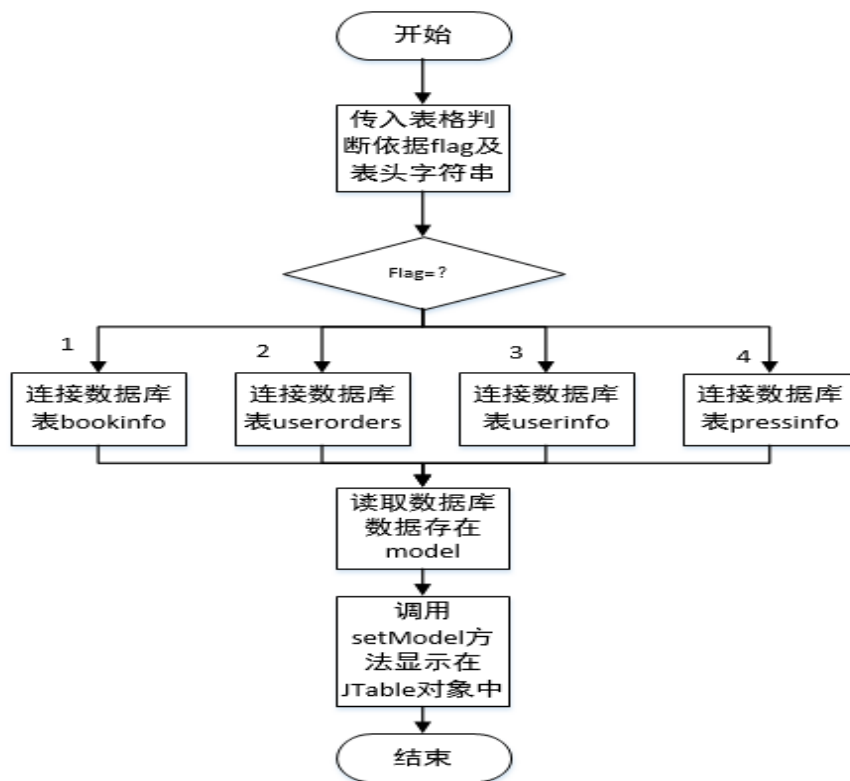
### 9.1 登录模块

1. 详细实现过程  
获取文本框字符串与数据库中账户姓名对比，相同则进入操作模块。
2. 流程图



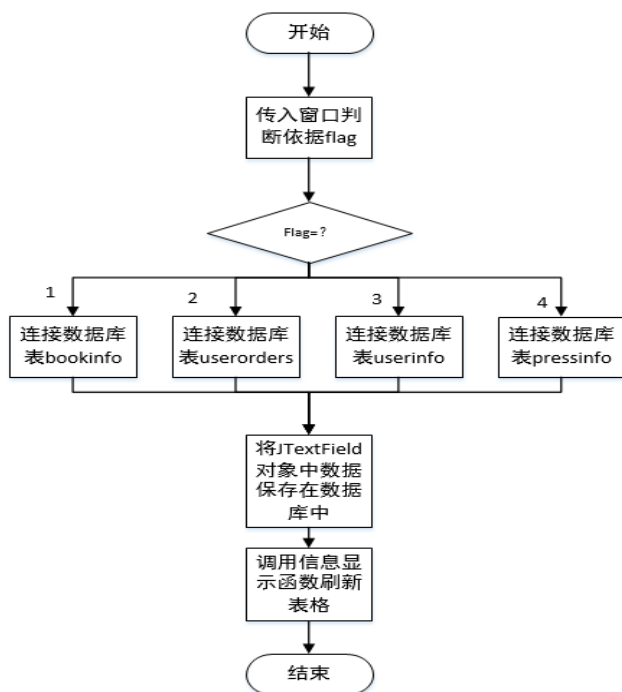
### 9.2 信息显示模块

1. 详细实现过程  
先将 DefaultTableModel 类对象 model 初始化，连接数据库，在利用 SQL 语句对数据库进行遍历，一行一行的读取数据存在 model 中，最后将调用 JTable 中的 setModel 方法实现表格的显示。
2. 流程图



### 9.3 信息增加模块

1. 详细实现过程  
新建了一个继承 JFrame 的 addinfo 类，通过传入的 flag 来判断对哪个表单进行操作，直接修改数据库，在调用信息显示方法。
2. 流程图

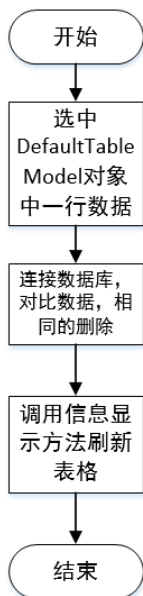


## 9.4 信息删除模块

### 1. 详细实现过程

选中表格中一行，对比数据库中数据，相同的移除，再调用显示方法刷新表格。

### 2. 流程图

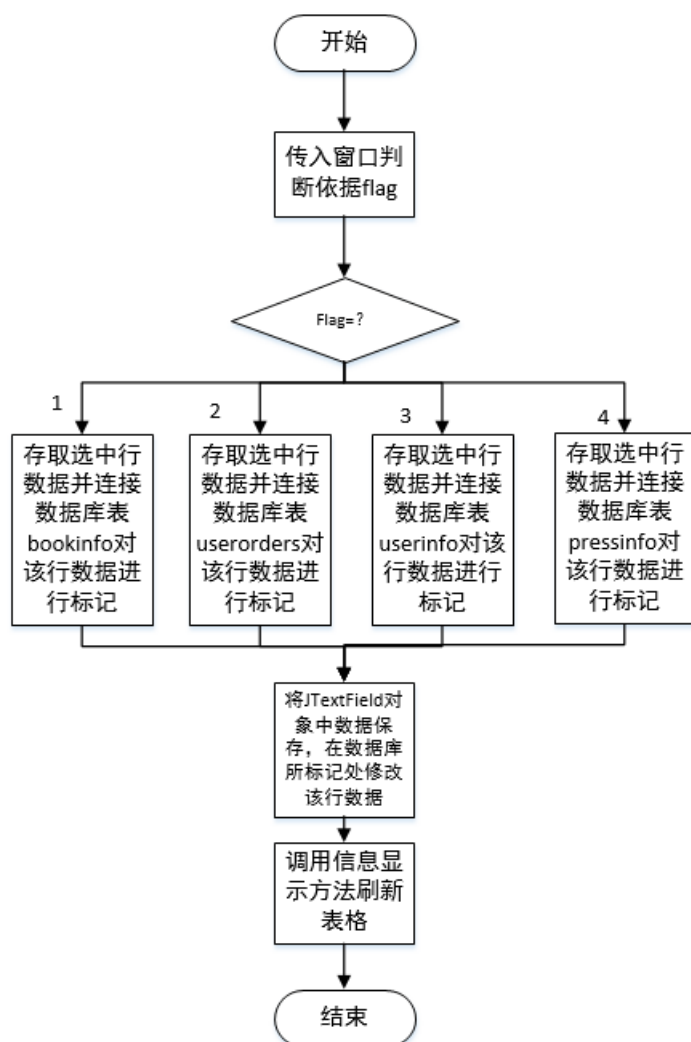


## 9.5 信息修改模块

### 1. 详细实现过程

新建继承 JFrame 的 changinfo 类，选中 DefaultTableModel 对象一行读取数据并在数据库中对这一行数据进行标记，点击确定按钮以后，读取 JTextField 对象数据对已标记的这一行进行修改，最后调用显示方法显示。

## 2. 流程图

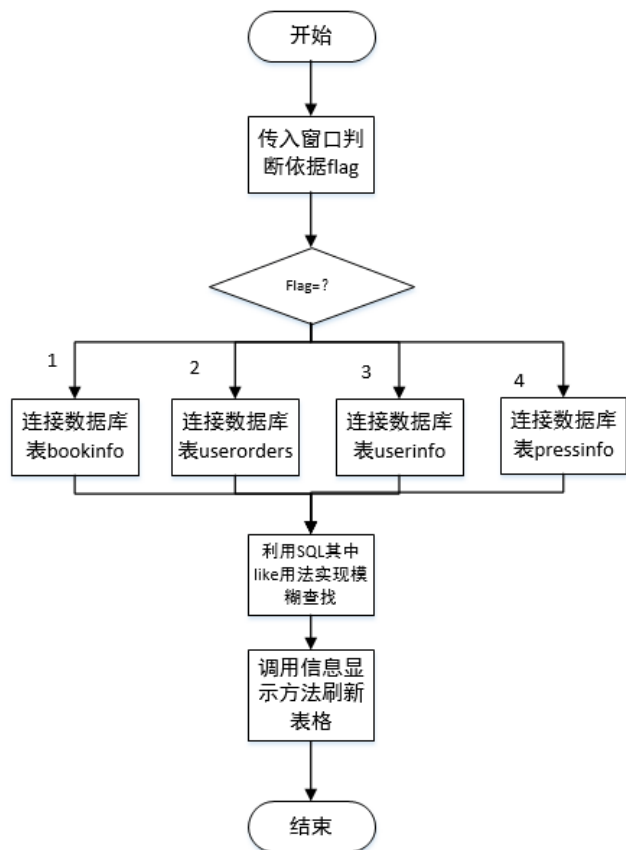


## 9.6 信息查找模块

### 1. 详细实现过程

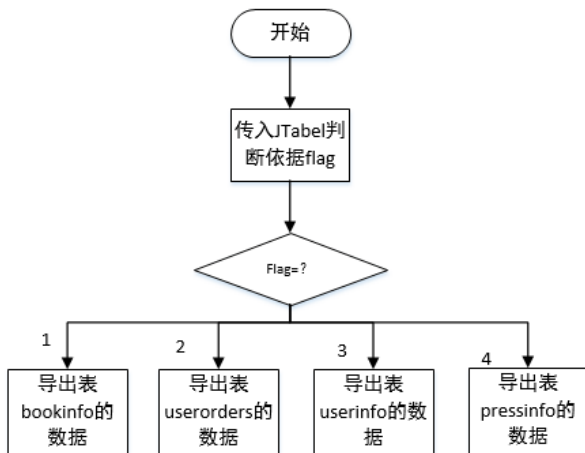
新建继承 JFrame 的 search 类作为查找的弹出窗口，根据传入的 flag 判断表单并连接对应数据库，利用 SQL 中的 like 语句进行每一列的模糊查找，数据传入 model 实现表格的刷新。

### 2. 流程图



## 9.7 生成报表模块

1. 详细实现过程  
利用 `FileDialog` 类创建对象，传入所要保存的 `JTable` 对象。
2. 流程图



# 第10部分 运行效果图

1. 登录界面
- 输入用户名，密码点击确认。

图书销售管理系统

用户名

Admin

密码

.....

确定

取消

2. 表单切换选项界面
- 点击表单，点击任意表单进行切换。

图书销售管理系统

生成报表

表单

关于

商品信息

客户订单

客户信息

厂商信息

	作者	出版社	价格	ISBN	库存
第一行代码	郭鑫	人民邮电出版社	79	978-7-115-43978-9	200
数字逻辑第	阎守谦，孙圣铸	国防工业出版社	49	978-7-118-07647-9	200
为版十二载	所罗门·诺瑟普	北京大学出版社	26	978-7-301-23812-7	150

增加

删除

修改

查询

返回当前表单

图书销售管理系统

生成报表

表单

关于

书名	作者	出版社	价格	数量	总价	客户	下单时间
白夜行	东野圭吾	南海出版社	38.5	10	385	罗逍遥	2017-6-3
摆渡人	克莱尔·麦克福尔	百花洲文艺出版社	36.00	150	5400	andy	2017-06-10

增加

删除

修改

查询

返回当前表单

3. 添加信息界面

图书销售管理系统

生成报表 表单 关于

书名	作者
第一行代码Android	郭霖
数学建模算法与应用	司守奎，孙玺菁
为奴十二载	所罗门 诺瑟普

添加商品信息

书名

国城

作者

钱钟书

出版社

人民文学出版社

价格

39

ISBN

978-7-02-009000-6

库存

500

确定

取消

增加

删除

修改

查询

返回当前表单

4. 删除界面

选中某一行以后点删除。

图书销售管理系统

生成报表 表单 关于

书名	作者	出版社	价格	ISBN	库存
第一行代码Android	郭霖	人民邮电出版社	79	978-7-115-43978-9	200
数学建模算法与应用	司守奎，孙玺菁	国防工业出版社	49	978-7-118-07647-9	200
为奴十二载	所罗门 诺瑟普	北京大学出版社	26	978-7-301-23812-7	150
国城	钱钟书	人民文学出版社	39	978-7-02-009000-6	500

增加

删除

修改

查询

返回当前表单

图书销售管理系统

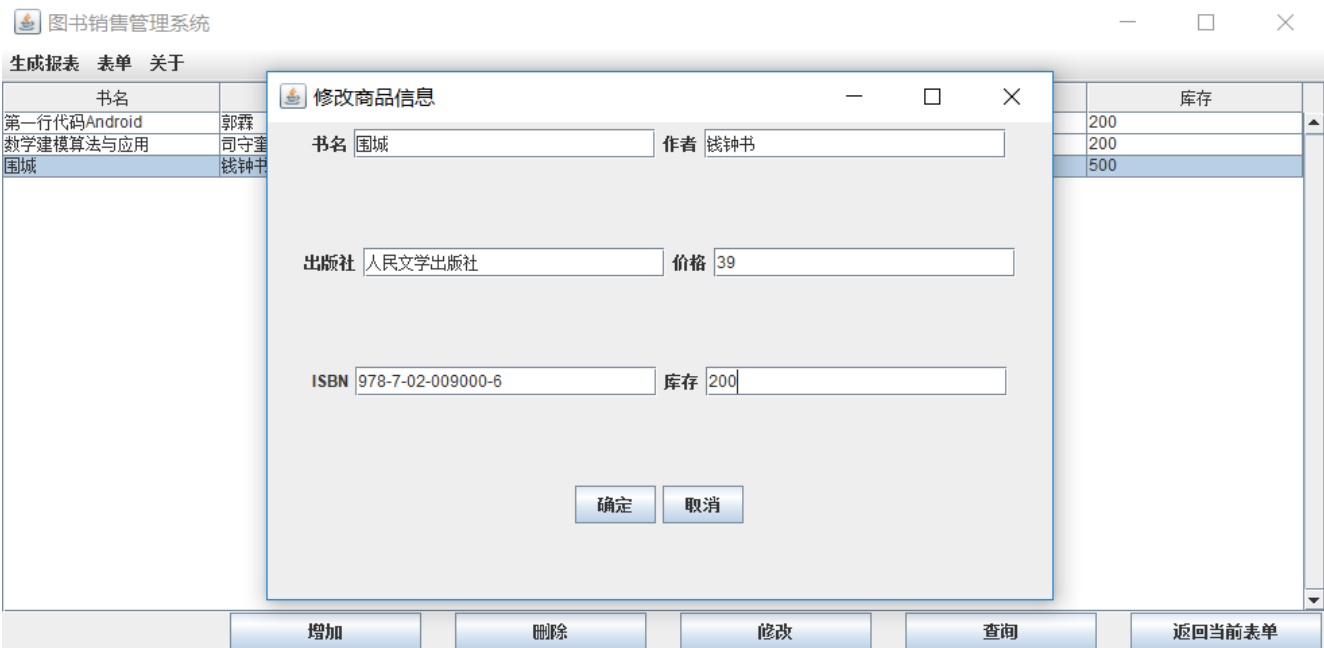
生成报表 表单 关于

书名	作者	出版社	价格	ISBN	库存
第一行代码Android	郭霖	人民邮电出版社	79	978-7-115-43978-9	200
数学建模算法与应用	司守奎，孙玺菁	国防工业出版社	49	978-7-118-07647-9	200
国城	钱钟书	人民文学出版社	39	978-7-02-009000-6	500



5. 修改界面

选中某一行后，点击修改。



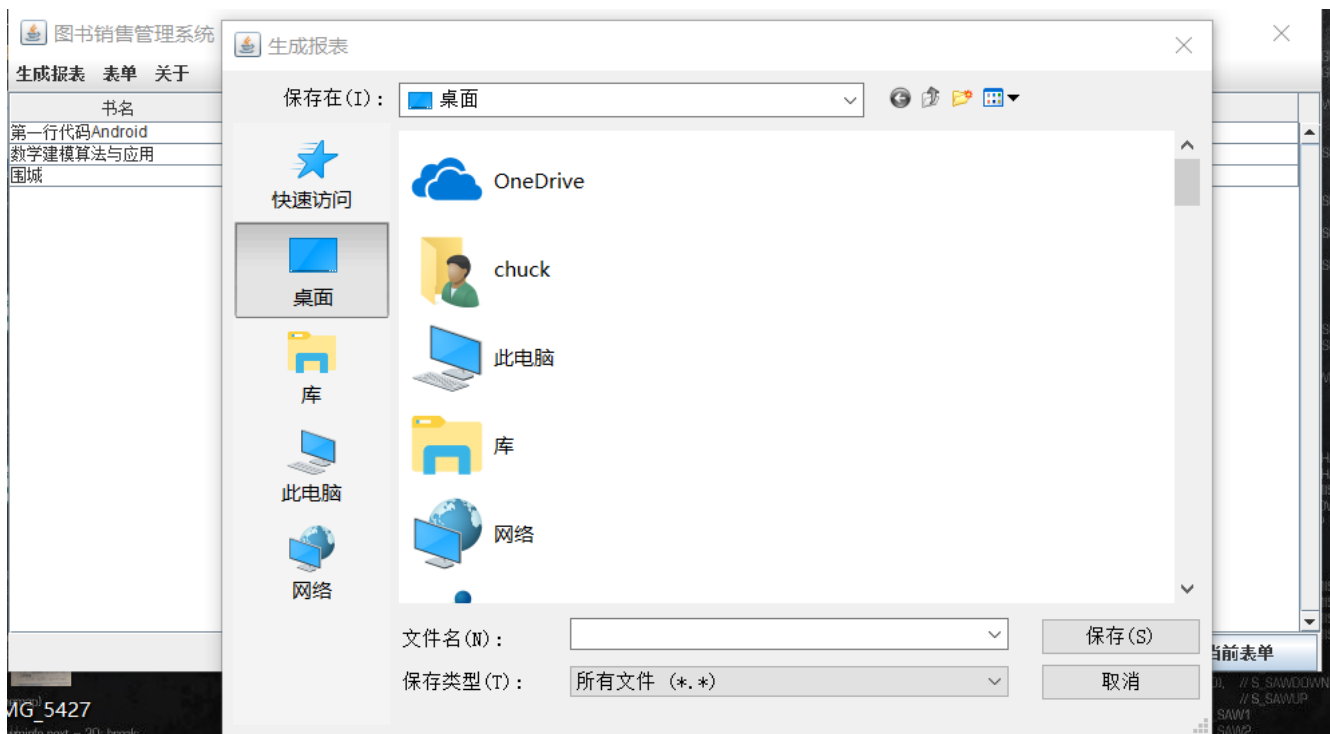
6. 查询界面

任意输入关键信息，只要被包含都可以被查出来，而且可以直接在其上进行修改。



7. 生成报表界面

选择路径以后，直接打文件名，不带后缀名，不保存即可。



## 第11部分 心得与体会

在本课设的制作过程中，第一个遇到的问题是由于 Java 的版本变化导致原本驱动的不存在，因此专门导入了另外的 Jar 包来实现与 access 的连接。最开始考虑的使用 JTextArea 组件来显示信息，但在上网查询相关信息以后，找到了更为简洁方便的组件 JTable，在 JTable 中数据基本操作的编写中，有两种方案，一种是同步对数据库刷新数据，另一种是先在 JTable 的 model 中直接更改，最后将表中数据全部存取数据库，第一种方式更有效的避免了程序中途出错无法保存的情况，但是代码量较于第二种大大增加，出于多方原因，我还是选择的第一种方法，也因此代码很长，但重复率也相对高。在增加信息程序的编写上，监听窗口关闭的方法最为关键，同步更新数据库以及读取都要在增加信息窗口的关闭以后进行。

再有一些编程时候遇到的小问题，比如本来在 JTable 组件中可以直接在表头上增加排序的功能，但是由于编写方法的缘故，这个方法会使 model 的行数固定，重新读取数据库的方法不能用，但如果直接运用 DefaultTableModel 类中的方法就可以解决，但和程序主体方法有偏差，故去掉了这个方法。

在写查找函数的时候，本来打算通过精确查找来实现，但意外的发现了 SQL 语句中 like 的用法，实现了模糊查找。

类似的问题出现了一些，通过同学的讨论或者是自己对程序的不断调试从而逐个解决，虽然系统还是存在些许不足，但总的来说，已经比较满意，收获良多。

## 第12部分 附录

//登录部分

```
try{
    Connection con=DriverManager.getConnection(conURL);
    Statement s=con.createStatement();
    ResultSet rs=s.executeQuery("select * from user");
    while(rs.next()){

        if(s1.equals(rs.getString("user_id"))&&s2.equals(rs.getString("password")))
        {
            JOptionPane.showMessageDialog(null,"登录成功");
            this.setVisible(false);
            new Mainmenu(1);
        }
        else
        {
            flag=0;
            continue;
        }
    }
    if(flag==0)
        JOptionPane.showMessageDialog(null,"用户名或密码错误");
}catch(SQLException e1) {
    System.out.println("SQLException: " +e1.getMessage( ));
}
```

//显示方法核心代码

```
Connection con=DriverManager.getConnection(Login.conURL);
Statement s=con.createStatement( );
String r1="select * from bookinfo";
ResultSet rs=s.executeQuery(r1);
count = 0;
while(rs.next())
{
    count++;
}
Object[][] bookInfo = new Object[count][6];
count = 0;
model.setRowCount(0);
rs=s.executeQuery(r1);
while(rs.next())
{
    bookInfo[count][0]=rs.getString("bookName");
    bookInfo[count][1]=rs.getString("author");
    bookInfo[count][2]=rs.getString("press");
    bookInfo[count][3]=rs.getString("price");
    bookInfo[count][4]=rs.getString("ISBN");
    bookInfo[count][5]=rs.getString("stock");
    model.addRow(bookInfo[count]);
    count++;
}
s.close( );
con.close( );
}
catch(SQLException ex)
{
    System.out.println("SQLException: " +ex.getMessage( ));
}
```

//增加信息

```
public void actionPerformed(ActionEvent e) {
```

```

        if(e.getSource()==b2)
            dispose();
        else if(i==1)
        {

            try
            {
                Connection con=DriverManager.getConnection(Login.conURL);
                Statement s=con.createStatement( );

                bookName=t1.getText() ;
                author=t2.getText() ;
                press=t3.getText() ;
                price=t4.getText() ;
                ISBN=t5.getText() ;
                stock=t6.getText() ;
                String r2="insert into bookinfo values('"+bookName+"','"
                    +author+"','"+press+"','"+price+"','"+ISBN
+ "','"+stock+"')";

                s.executeUpdate(r2);
                s.close( );
                con.close( );
            }
            catch(SQLException ex)
            {
                System.out.println("SQLException: " +ex.getMessage( ));
            }
            dispose();
        }
        //删除部分
        try
        {
            Connection con=DriverManager.getConnection(Login.conURL);
            Statement s=con.createStatement( );

            for(int i=0;i<8;i++)
                data[i]=(String) model.getValueAt(rowNumber, i);

            System.out.println(data[4]);
            String r1="delete from userorders where bookName='"+data[0]+"'"
                + "and press='"+data[2]+"'"
                + "and price='"+data[3]+"'"
                + "and total='"+data[5]+"'"
                + "and time='"+data[7]+"'";
            author='"+data[1]+"'"
            number='"+data[4]+"'"
            costume='"+data[6]+"'"
            s.executeUpdate(r1);

            s.close( );
            con.close( );
        }
        catch(SQLException ex)
        {
            System.out.println("SQLException1: " +ex.getMessage( ));
        }
        //修改部分
        try
        {
            Connection con=DriverManager.getConnection(Login.conURL);
            Statement s=con.createStatement( );

```

```

        String name = t1.getText();
        String address=t2.getText();
        String phone=t3.getText();
        String email=t4.getText();
        String r="update userinfo set name='"+name+"',address
='"+address+"',phone='"+phone+"'"
                                + ",email='"+email+"'where name='change'";
        s.executeUpdate(r);

        s.close( );
        con.close( );
    }
    catch(SQLException ex)
    {
        System.out.println("SQLException: " +ex.getMessage( ));
    }
    dispose();    }

//查找部分
try
{
    Connection con=DriverManager.getConnection(Login.conURL);
    Statement s=con.createStatement( );
    String r1="select * from userinfo where name like '%" +aa+"%'or address
like '%" +aa+"%'
                                + "or phone like '%" +aa+"%' or email like '%" +aa+"%'";
    ResultSet rs=s.executeQuery(r1);
    count = 0;
    while(rs.next())
    {
        count++;
    }
    String[] bookInfo = new String[4];
    rs=s.executeQuery(r1);
    while(rs.next())
    {
        bookInfo[0]=rs.getString("name");
        bookInfo[1]=rs.getString("address");
        bookInfo[2]=rs.getString("phone");
        bookInfo[3]=rs.getString("email");
        model.addRow(bookInfo);
    }
    s.close( );
    con.close( );
}
catch(SQLException ex)
{
    System.out.println("SQLException: " +ex.getMessage( ));
}

//生成报表
FileDialog fd = new FileDialog(this, "生成报表", FileDialog.SAVE);
    fd.setLocation(400, 250);
    fd.setVisible(true);
    String stringfile = fd.getDirectory()+fd.getFile()+".xls";
    try {
        table1.exportTable(table1.t1, new File(stringfile));
    } catch (IOException ex) {
        System.out.println(ex.getMessage());
        ex.printStackTrace();
    }
}

```