



河南工业大学

Java 课程设计报告

题 目：_____ 电脑配件库存管理系统 _____

专 业 班 级：_____ 计科 F1602 _____

学 生 姓 名：_____ 宋广辉 _____

学 号：_____ 201616010113 _____

指 导 教 师：_____ 王 峰 _____

课程设计时间：_____ 2018.6.18—2018.6.29 _____

计算机科学与技术 专业课程设计任务书

学生姓名	宋广辉	专业班级	计科 F1602	学号	201616010113
题 目	电脑配件库存管理系统				
课题性质	工程设计	课题来源	自拟课题		
指导教师	王峰	同组姓名	无		
主要内容	<p>1、课程设计目标：</p> <p>1) 按照面向对象程序设计思想，结合软件工程的系统开发流程，利用相关开发工具，能够系统分析所面对的工程问题，并进行系统概要设计和详细设计；</p> <p>2) 结合信息社会的需求变化，能够积极分析工程技术应用领域出现的社会、健康、安全、法律、文化以及环境等新问题，尝试采取新思路在系统设计方案中积极应对；</p> <p>3) 能根据设计方案的要求，评价设计方案实施后带来的经济与社会效益；</p> <p>4) 能根据设计方案的要求，评价工程项目实施的解决方案对社会、健康、安全等方面的影响。</p> <p>2、课程设计内容：设计一个简单的电脑配件库存管理系统，实现以下主要功能：</p> <p>1) 电脑配件资料的添加、修改与删除；</p> <p>2) 电脑配件的进货、退货与出货管理；</p> <p>3) 电脑配件的库存查询；</p> <p>4) 各种信息统计分析功能。</p>				
任务要求	<p>1、采用 Java 语言和面向对象程序设计方法编写程序，实现问题求解；</p> <p>2、注意养成添加注释等良好编程习惯；代码要求达到一定工作量，并具有一定的深度和难度；</p> <p>3、根据课题程序的分析和设计过程，按规定的格式撰写课程设计报告；</p> <p>4、课程设计结束后提交程序源代码、课程设计报告的电子稿和打印稿。</p>				
参考文献	<p>[1] 耿祥义.Java 2 实用教程(第 5 版) [M]. 北京：清华大学出版社, 2017.</p> <p>[2] 耿祥义. Java 课程设计（第 3 版）[M]. 北京：清华大学出版社, 2018.</p> <p>[3] 明日学院. Java 从入门到精通（项目案例版）[M].北京:水利水电出版社, 2017.</p> <p>[4] 何云清. MySQL 数据库程序设计 [M]. 北京：中国铁道出版社, 2018.</p> <p>[5] 明日科技. MySQL 从入门到精通 [M]. 北京：清华大学出版社, 2017.</p>				
审查意见	<p>指导教师签字：</p> <p>教研室主任签字：2018 年 6 月 18 日</p>				

目录

1 需求分析.....	1
1.1 管理员身份认证	1
1.2 配件进货入库	1
1.3 配件出货出库	1
1.4 配件退货回库	1
1.5 信息统计分析	2
1.6 库存查询	2
1.7 退出和保存信息	2
2 概要设计.....	3
2.1 设计的结构.....	3
2.2 主要类的 UML 图	3
2.2.1 main 类的 UML 图	3
2.2.2 User_sql 类的 UML 图	4
2.2.3 Login_ui 类的 UML 图	5
2.2.4 in 类的 UML 图如下:	6
3 运行环境.....	7
3.1 硬件及操作系统环境:.....	7
3.2 应用软件及编译环境:	7
4 开发工具和编程语言	8
4.1 开发工具:	8
4.2 编程语言:	8
5 详细设计.....	9
5.1 身份认证实现流程图	9
5.2 主要类的定义.....	9
5.2.1 main 包里 main 类的定义.....	9
5.2.2 mysql 包里 User_sql 类的定义	10
5.2.3 UI 包里 Login_ui 类的定义	12
5.2.4 control 包里 in 类的定义.....	16
6 调试分析.....	20
6.1 界面显示的问题	20

6.2 数据库连接失败的问题	20
6.3 程序中其他的问题.....	21
7 测试结果.....	22
7.1 登陆测试.....	22
7.2 入库测试.....	23
7.3 出库测试.....	25
7.4 回库测试.....	27
7.5 信息分析测试.....	28
7.6 自增（减）测试	30
7.7 菜单测试.....	33
参考文献.....	34
心得体会.....	35

1 需求分析

通过和指导老师交流，了解到本系统中的数据主要来源于数据库，通过实现对数据库中配件信息的添加、修改和删除，从而来实现对电脑配件的进货、出货以及退货的管理，根据信息分析的需要也可查询到其他的信息。此外，还需要实现一个管理员身份认证功能模块，使得只有通过管理员身份认证的用户，才可以正常使用本系统，否则，不准许其使用本系统。总之本系统，要具备如下功能：管理员身份认证、配件入货、配件出货、配件退货回库、信息统计分析、库存查询、退出和保存信息。具体如下：

1.1 管理员身份认证

提供可视化的窗口以交互的方式，来获取用户输入的身份信息，并和数据库中的管理员信息进行比对，如果两者相匹配，则允许用户进行信息的管理，否则，提示用户输入合法的管理员，直至信息相匹配或者用户主动关闭系统。

1.2 配件进货入库

能够将用户输入的合法的新进货的配件信息新增至数据库，并进行判断，如果配件总量不等于库存余量加销售量，则对用户进行提醒。

1.3 配件出货出库

根据给定的 number，能够更新该记录的库存余量和销售量。另外，提供批量出库和逐个出库两种方式，以便于用户更加便捷使用。

1.4 配件退货回库

根据给定的 number，能够更新该记录的库存余量和销售量。另外，提供批量回库和逐个回库两种方式，以便于用户更加便捷使用。

1.5 信息统计分析

实现与供应商相关信息、配件相关信息和价格相关信息的独立显示，便于用户更加直观清晰的了解库存数据库的信息。

1.6 库存查询

实现库存相关信息的独立显示，便于用户直观清楚明了的了解库存信息，以使用户合理的安排进出库。

1.7 退出和保存信息

实现安全退出和更新数据库信息后及时保存和刷新，保证信息的同步和安全性，增强系统的稳定性。

2 概要设计

2.1 设计的结构

使用模型-视图-控制器（Model-View-Controller，简称 MVC）结构，从面向对象的角度设计类，主要思想如下：

1. `mysql` 包作模型模块，存储对于数据库的信息以及对数据库进行基本操作的类以及方法。
2. `ui` 包作视图模块，提供给用户可视化的，便于交互的视图界面，为模型提供可视化显示的对象。
3. `control` 包做控制器模块，处理用户和系统的交互操作，对于用户的操作做出响应，让视图和模型进行交互，通过视图操作获取模型中的数据，同时当模型中的数据变化时，让视图更新显示。
4. `main` 包做程序开始模块，做程序主类，用于开始运行整个系统。

2.2 主要类的 UML 图

2.2.1 `main` 类的 UML 图

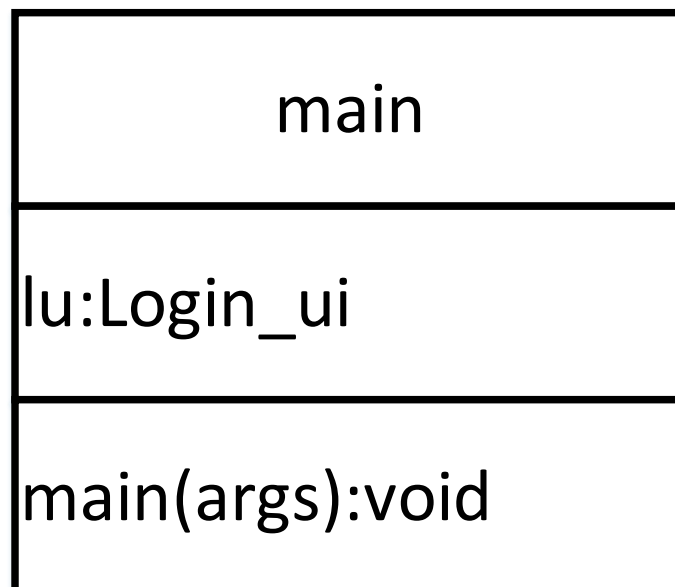


图 1 `main` 类的 UML 图如上

2.2.2 User_sql 类的 UML 图

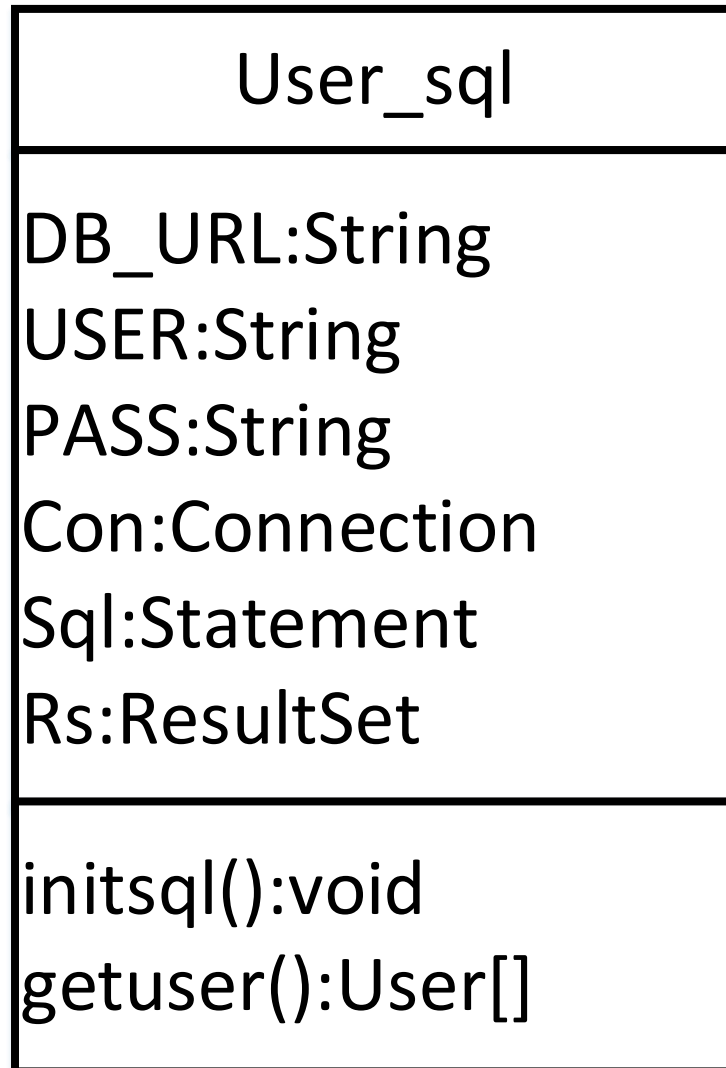


图 2 User_sql 类的 UML 图如上

2.2.3 Login_ui 类的 UML 图

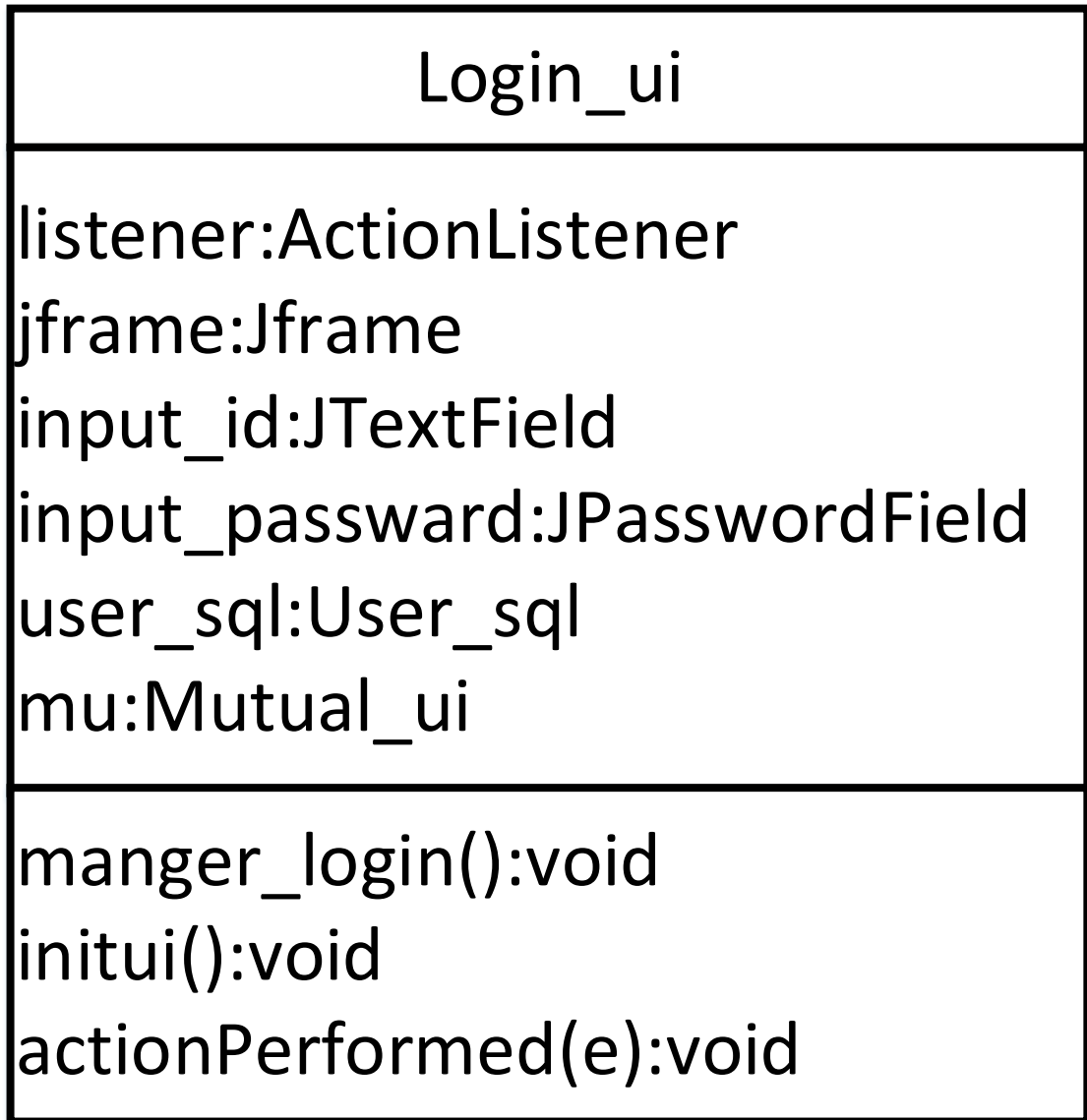


图 3 Login_ui 类的 UML 图

2.2.4 in 类的 UML 图如下：

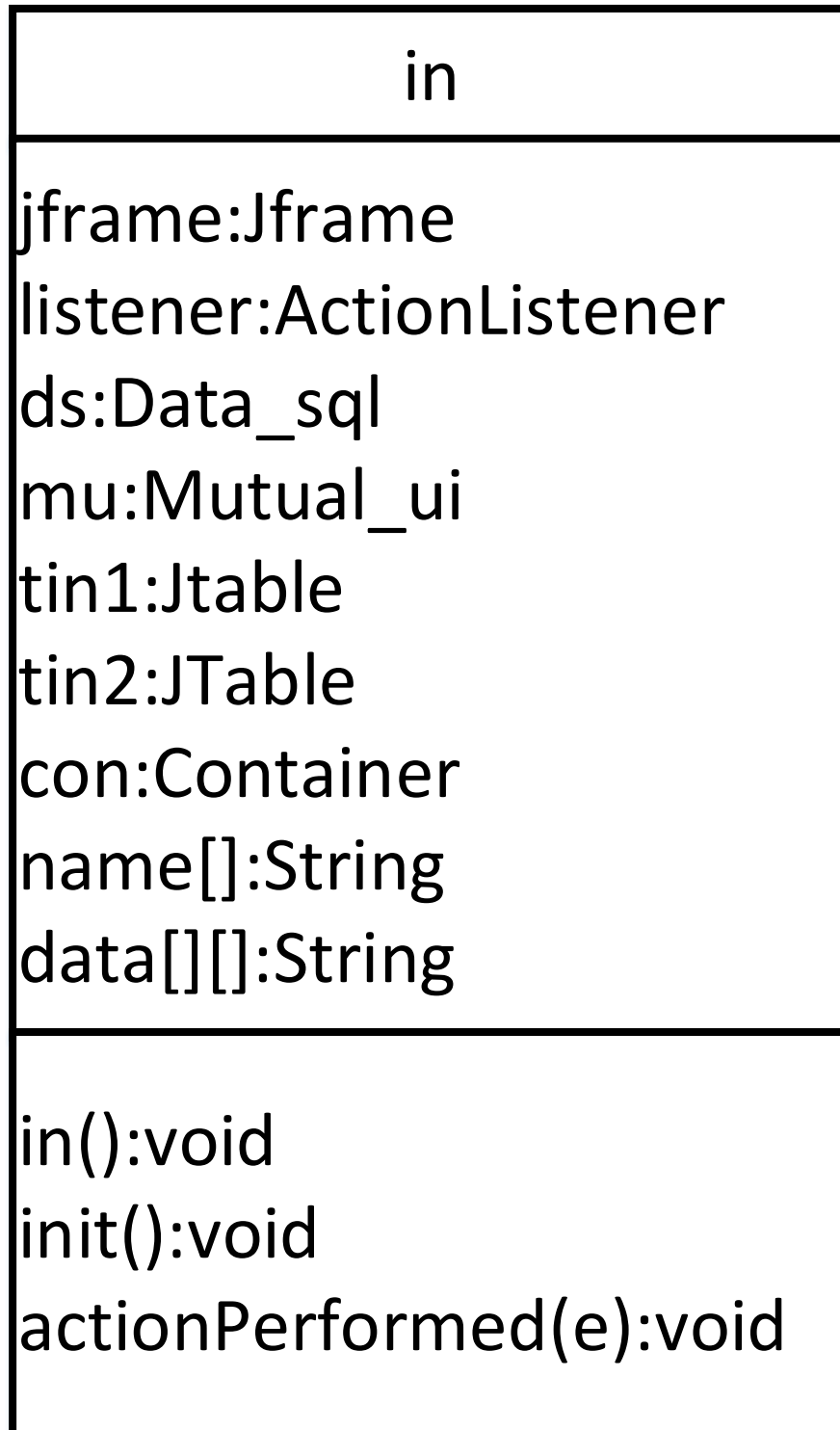


图 4 in 类的 UML 图

3 运行环境

3.1 硬件及操作系统环境:



图 5 电脑硬件配置详细信息

3.2 应用软件及编译环境:

eclipse.exe

Navicat 12 for MySQL.exe

GitHub Desktop.exe

jdk-9.0.4

jre-9.0.4

mysql-8.0.11-winx64

mysql-connector-java-8.0.11

4 开发工具和编程语言

4.1 开发工具：

Eclipse.exe

Navicat 12 for MySQL.exe

GitHub Desktop.exe

4.2 编程语言：

JAVA 语言

SQL 语句

5 详细设计

5.1 身份认证实现流程图

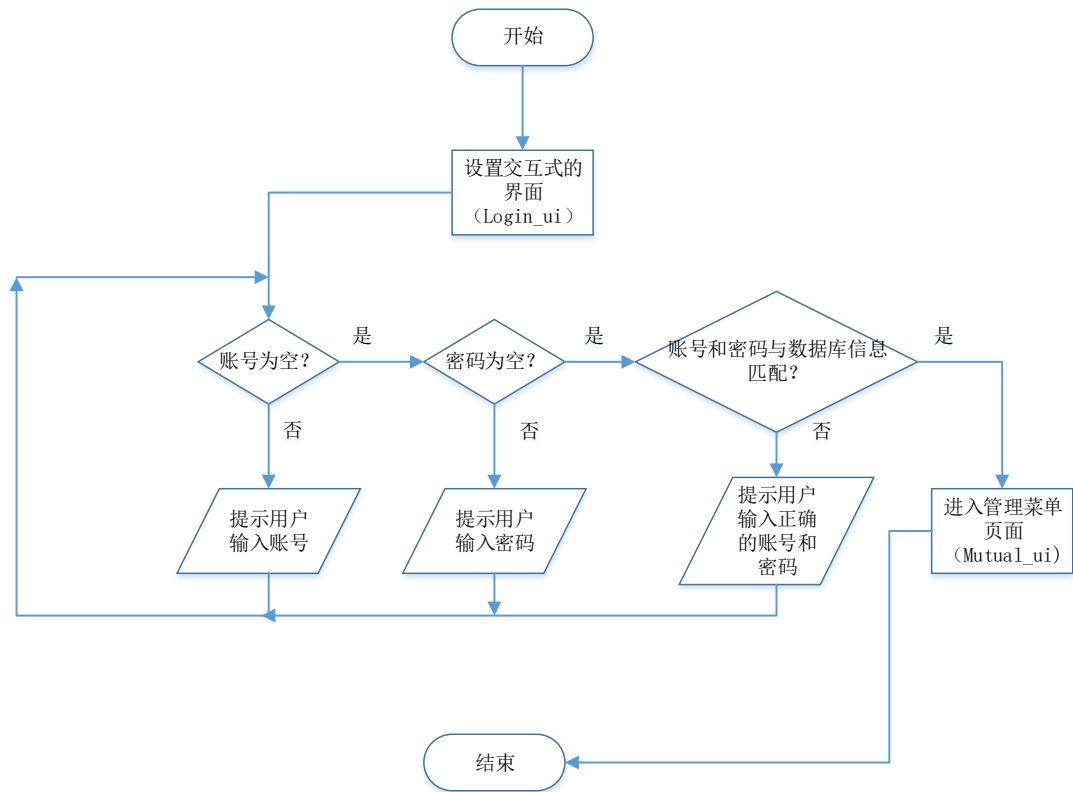


图 6 身份认证流程图

5.2 主要类的定义

5.2.1 main 包里 main 类的定义

```
package main;

import UI.*;

/**
 * 主类
 * @author 广辉
```

```

*/
public class main_codes{
    static Login_ui lu=new Login_ui();
    public static void main(String[] args) {
        lu.manger_login();
    }
}

```

5.2.2 mysql 包里 User_sql 类的定义

```

package mysql;
import java.sql.*;
/**
 * 用户类数据库
 */
public class User_sql{
    // JDBC 驱动名及数据库 URL
    Static final String DB_URL = "jdbc:mysql://localhost:3306/user?&useSSL=false
    &serverTimezone=UTC";
    // 数据库的用户名与密码
    static final String USER = "root";
    static final String PASS = "";
    Connection con = null;
    Statement sql;
    ResultSet rs;
    //初始化连接数据库
    public void initsql() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver"); //加载 MYSQL

```

JDBC 驱动程序

```
//System.out.println("成功加载驱动， 请继续！ ");
}catch (Exception e) {
    System.out.print("Error loading Mysql Driver!");
    e.printStackTrace();
}
try{
    // 打开链接
    // System.out.println("连接管理员数据库...");
    con = DriverManager.getConnection(DB_URL,USER,PASS);
    //System.out.println("连接管理员数据库成功!!! ");
}catch(SQLException e){    // 处理 JDBC 错误
    System.out.println(e);
}
}

/**
 * 从数据库获取 user 表中的 user
 * @return
 */
public User[] getuser() {
    int i=0;
    User user[];
    user=new User[12];
    try {
        sql=con.createStatement();//创建 SQL 语句对象
        rs=sql.executeQuery("select id,password,birthday from `register`");
        while(rs!=null && rs.next()) {
            user[i]=new    User(rs.getString("id"),  rs.getString("password"),
rs.getDate("birthday"));
        }
    }
}
```

```

        //user[i].show();

        i++;

        //System.out.println("I can run to here!\n");
    }

    rs.close();

    //con.close();

}catch(SQLException e) {

    System.out.println(e);

}

return user;

}

}

```

5.2.3 UI 包里 Login_ui 类的定义

```

package UI;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import mysql.*;

/**
 * 登陆界面类如下
 * @author 广辉
 */
@SuppressWarnings("serial")
public class Login_ui extends JFrame implements ActionListener{

    ActionListener listener;

```



```

JFrame jframe=new JFrame();

private JTextField input_id=new JTextField(12);

private JPasswordField input_password=new JPasswordField(12);

User_sql usersql=new User_sql();

Mutual_ui mu=new Mutual_ui();


public void manger_login() {

    initui();

    usersql.initsql();

}


private void initui() {

    jframe.setTitle("欢迎来到电脑配件库存管理系统");

    jframe.setLayout(new FlowLayout());

    Box box;

    Box box1,box2,box3;

    box=Box.createHorizontalBox();

    box1=Box.createVerticalBox();

    box2=Box.createVerticalBox();

    box3=Box.createVerticalBox();

    JLabel jlabel1,jlabel2;


    jlabel1=new JLabel("账户： ");

    jlabel2=new JLabel("密码： ");

    JCheckBox cb=new JCheckBox("记住密码");

    JButton bt=new JButton("登陆");


    //利用嵌套实现好看的布局

    box1.add(jlabel1);

    box2.add(input_id);

```

```

        box1.add(jlabel2);
        box2.add(input_password);
        box.add(box1);
        box.add(Box.createHorizontalStrut(10));
        box.add(box2);
        box3.add(box);
        box3.add(cb);
        box3.add(bt);

        bt.addActionListener(this);

        jframe.add(box3);
        jframe.pack();

        Container con1=this.getContentPane();
        con1.setBackground(Color.white);
        jframe.setBounds(580,300,425,280);
        jframe.setResizable(false);
        jframe.setVisible(true);
        jframe.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }

    /**
     * 点击登录按钮后的事件处理函数
     */
    public void actionPerformed(ActionEvent e) {
        User [] user;
        user=usersql.getuser();
        if(input_id.getText().isEmpty())
        {

```

```

        JOptionPane.showMessageDialog(this, "请输入您要登陆的账户!");
        this.dispose();
    }
    else if(new String(input_password.getPassword()).isEmpty())
    {
        JOptionPane.showMessageDialog(this, "请输入密码!");
        this.dispose();
    }
    else if (!(input_id.getText().equals(null)) && !(new
String(input_password.getPassword()).equals(null)))
    {
        int i=0;
        for(;i<User.total;i++) {
            if(input_id.getText().equals(user[i].getid()) && new
String(input_password.getPassword()).equals(user[i].getpassword()))
            {
                //System.out.println("登陆成功\n");
                jframe.dispose();
                mu.manger_menu();
                mu.center_table();
                break;
            }
        }
        if(i>=User.total)
        JOptionPane.showMessageDialog(this, "账户或者密码输入错误!");
    }
}
}
}

```

5.2.4 control 包里 in 类的定义

```
package control;

import mysql.*;

import UI.Mutual_ui;

import UI.Win_table;

import javax.swing.*.*;

import javax.swing.table.DefaultTableModel;

import java.awt.BorderLayout;

import java.awt.Button;

import java.awt.Container;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

@SuppressWarnings({ "serial", "unused" })

public class in extends JFrame implements ActionListener{

    JFrame jframe;

    ActionListener listener;

    Data_sql ds=new Data_sql();

    Mutual_ui mu=new Mutual_ui();

    JTable tin1,tin2;

    Container con;

    String name[]= ds.getcolumnname();

    String [][] data=new String[1][8];

    public in() {

        jframe=new JFrame();

        this.setLayout(new GridLayout(3,1));//使用网格布局

        init();

    }

}
```

```

ds=new Data_sql();
tin2=new JTable(ds.getrecord(),ds.getcolumnname());
tin2.setCellSelectionEnabled(true);
this.add(tin2);
add(new JScrollPane(tin2),BorderLayout.CENTER);

```

```

JScrollPane js=new JScrollPane(tin1);
js.setLocation(0,0);
js.setSize(670,150);
add(js,BorderLayout.NORTH);

```

```

con =getContentPane();
JButton bt=new JButton("确保表格（除 Others 外），均为正确数据时再
点击 ‘入库’，即可成功添加！");

```

```

bt.addActionListener(this);
bt.setSize(670,50);
setLocation(0,400);
con.add(bt,BorderLayout.SOUTH);

```

```

setTitle("配件入库");
setLocation(480,200);
setSize(670,455);
setVisible(true);
validate();
setDefaultCloseOperation(DISPOSE_ON_CLOSE);

```

```

}

```

```

private void init() {
    int value=0;
    for(int i=0;i<data.length;i++){
        for (int j=0;j<data.length;j++){
            data[i][j]=String.valueOf(value++);
        }
    }

    tin1=new JTable(data,name);
    tin1.setCellSelectionEnabled(true);
    this.add(tin1);
}

public void actionPerformed(ActionEvent e) {
    JOptionPane.showMessageDialog(this, "是否确认添加新配件? ");
    this.dispose();

    int i=0;
    for(;i<7;i++) {
        if(data[0][i]==null)
        {
            JOptionPane.showMessageDialog(this, "输入不合法! ");
            //this.dispose();

            i=10;
            break;
        }
    }

    if(i==10) {
        try {
            Thread.sleep(5000);

```

```
        } catch (InterruptedException e1) {  
            // TODO 自动生成的 catch 块  
            e1.printStackTrace();  
        }  
    }  
    if(data[0][7]==null)  
        data[0][7]=" ";  
    ds.in(name, data[0]);  
    this.dispose();  
    new repaint();  
}  
}
```

6 调试分析

6.1 界面显示的问题

在对界面进行必要的初始化和设置之后，程序可以正常运行，但是设置的按钮和表格组件均无法显示，经过仔细的调试也并没有发现 **bug**，每一个语句也都可以正常执行，但是仍然无法显示。一次意外地拖动界面的大小，发现拖动之后可以正常显示组件。百般解决无果后，去百度搜索之后发现好多人也都遇到了这个问题，而且上面有不同的解决方案，在仔细尝试所有的解决方案，并对方案加以理解和分析之后，才最终明白，之所以会出现这个情况，是因为自己把设置函数放在了代码的前面，放在设置方法的最后即可解决这一问题。

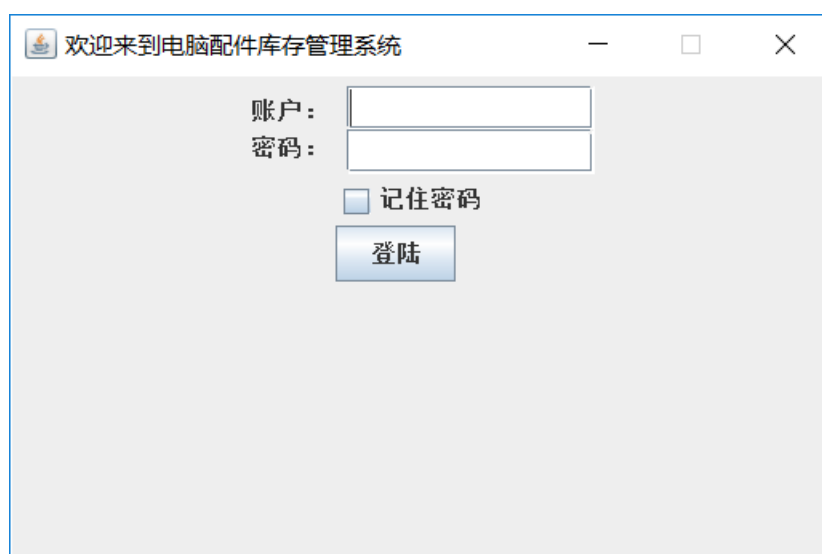


图 7 运行之后弹出的正常页面

6.2 数据库连接失败的问题

在确定安装完成数据库，同时利用 Navicat 12 for MySQL.exe 可以正常连接数据库，并且可以完成创建表格等操作后，尝试利用 Eclipse.exe 进行连接，虽然连接 Mysql 驱动程序按照课本上的操作一次就成功连接，但是之

后无论如何也连接不上数据库，经过调试并利用 Navicat 12 for MySQL.exe 修改信息之后发现，即使账户和密码一定正确，连接仍是失败，于是便把问题锁定在了数据库 URL (DB_URL) 上，经过多次去百度搜索，多次尝试各种解决方案，最终发现，把 URL (DB_URL) 中数据库名字后面的字符串修改为 “?&useSSL=false&serverTimezone=UTC” 即可解决这一问题，成功连接上 Mysql 数据库。

```
com.mysql.cj.jdbc.exceptions.CommunicationsException: Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the
exception in thread "AWT-EventQueue-0" com.mysql.cj.jdbc.exceptions.CommunicationsException: Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the
java.lang.NullPointerException
    at mysql.Data_sql.setColumnName(Data_sql.java:168)
    at mysql.Data_sql.setrecord(Data_sql.java:185)
    at mysql.Data_sql.judge(Data_sql.java:57)
    at UI.Mutual_ui.center_table(Mutual_ui.java:119)
    at UI.Login_ui.actionPerformed(Login_ui.java:111)
    at java.desktop/javaw.swing.AbstractButton.fireActionPerformed(Unknown Source)
    at java.desktop/javaw.swing.AbstractButton$Handler.actionPerformed(Unknown Source)
    at java.desktop/javaw.swing.DefaultButtonModel.fireActionPerformed(Unknown Source)
    at java.desktop/javaw.swing.DefaultButtonModel.setPressed(Unknown Source)
    at java.desktop/javaw.swing.plaf.basic.BasicButtonListener.mouseReleased(Unknown Source)
    at java.desktop/java.awt.Component.processMouseEvent(Unknown Source)
    at java.desktop/javaw.swing.JComponent.processMouseEvent(Unknown Source)
    at java.desktop/java.awt.Component.processEvent(Unknown Source)
    at java.desktop/java.awt.Container.processEvent(Unknown Source)
    at java.desktop/java.awt.Component.dispatchEventImpl(Unknown Source)
    at java.desktop/java.awt.Container.dispatchEventImpl(Unknown Source)
    at java.desktop/java.awt.Component.dispatchEvent(Unknown Source)
    at java.desktop/java.awt.LightweightDispatcher.retargetMouseEvent(Unknown Source)
    at java.desktop/java.awt.LightweightDispatcher.processMouseEvent(Unknown Source)
```

图 8 数据库连接失败错误的截图

6.3 程序中其他的问题

在详细设计程序的过程中，主要是写代码阶段出现了各种各样的问题，比较难以解决的是上述的两个，其他的类似 SQL 语句中的特殊字符要用反引号 (‘) 包括起来等等的小问题，在百度上搜索一下，在经尝试、分析和对比即可解决的小问题就不在此一一赘述了。

7 测试结果

7.1 登陆测试

打开软件后，出现如下界面：

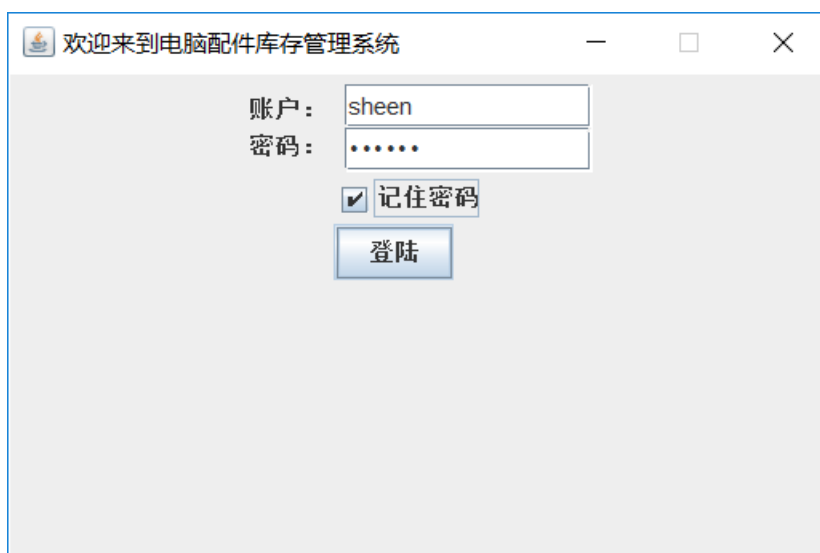


图 9 sheen 管理员登陆截图

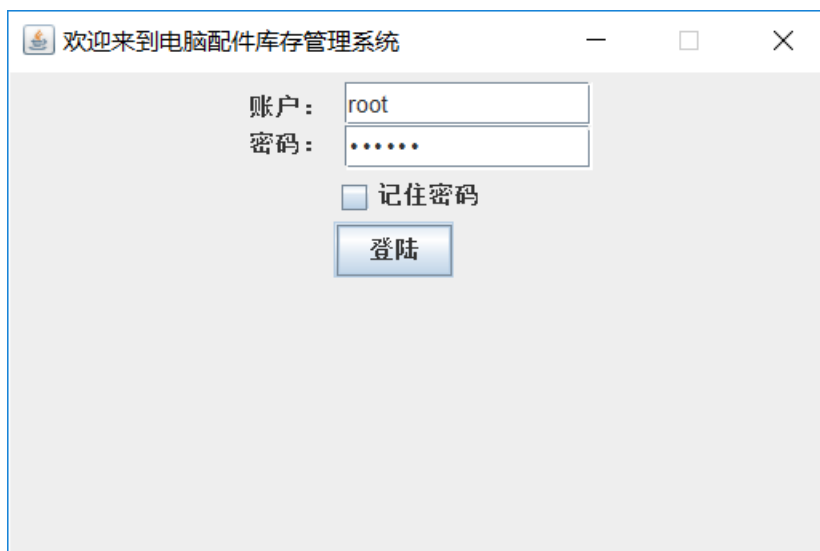


图 10 root 管理员登陆截图

7.2 入库测试

点击菜单下面的配件入库后，出现如下界面：

配件入库

number	type	total	allowance	sales	price	supplier	others
1	处理器	600	521	79	199	Intel	
2	主板	500	470	30	600	asus	
3	内存	1000	800	200	500	samsung	
4	声卡	1000	500	500	10	asus	
5	网卡	1000	450	550	12	intel	
6	显示器	600	330	270	350	aoc	
7	电源	550	350	200	100	长城	
13	机械硬盘	600	599	1	329	WD	

确保表格（除Others外），均为正确数据时再点击‘入库’，即可成功添加！

图 11 入库操作截图

点击下面的按钮后，出现如下界面：



图 12 确定入库消息对话框截图

点击确定后，出现如下界面：

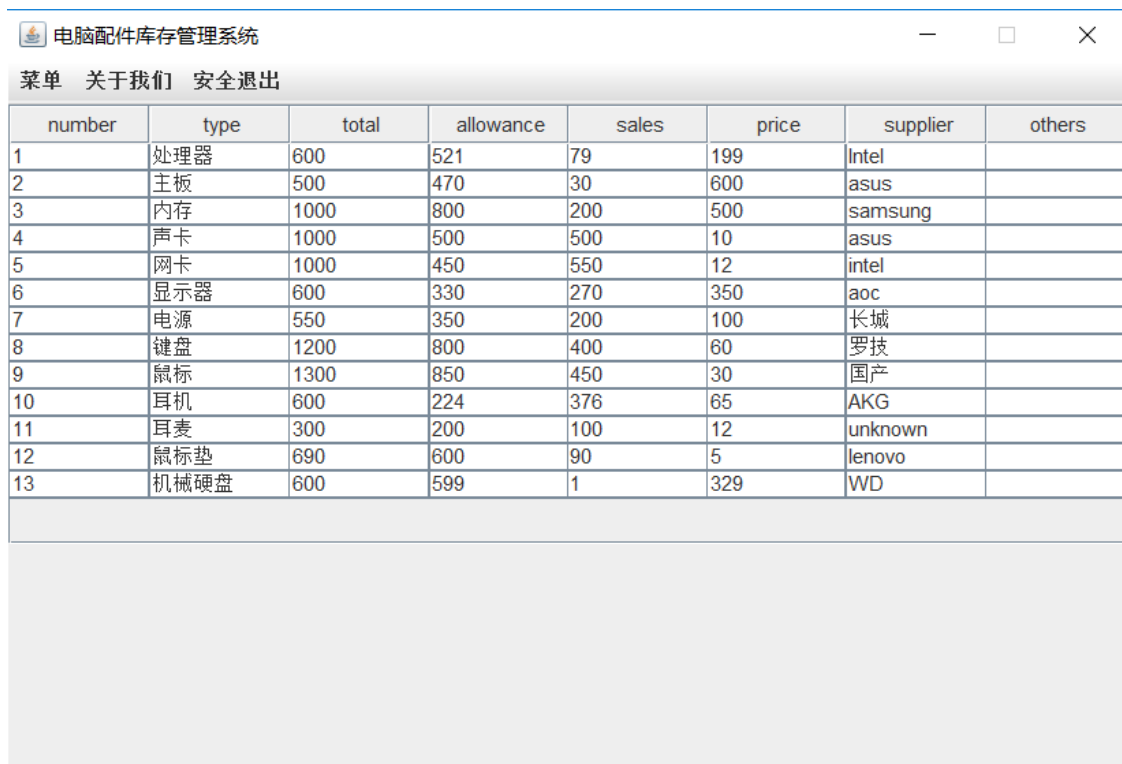


图 13 入库完成后截图



图 14 多次入库完成后截图

7.3 出库测试

点击菜单下面的配件出库后，出现如下界面：

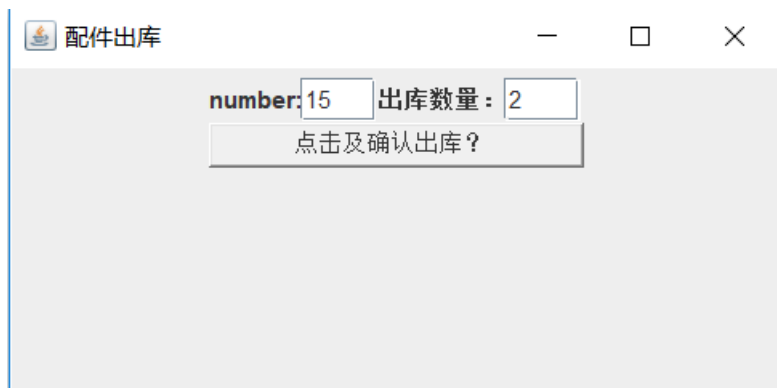


图 15 出库操作截图

点击下面的按钮后，出现如下界面：

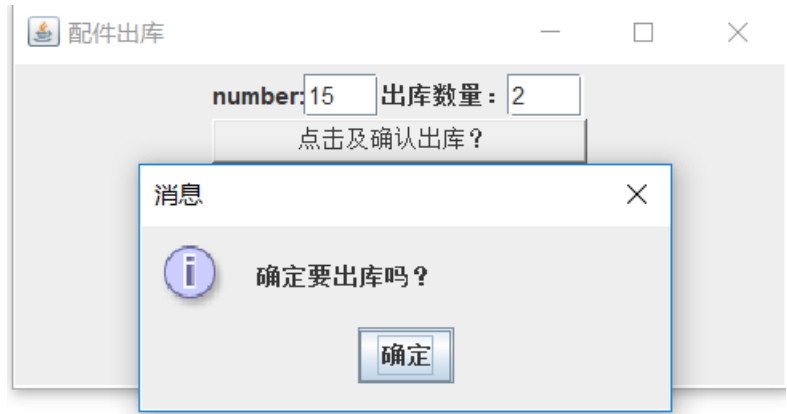


图 16 确定出库消息对话框截图

点击确定按钮后，出现如下界面：



图 17 出库完成后截图

7.4 回库测试

点击菜单下面的退货回库后，出现如下界面：

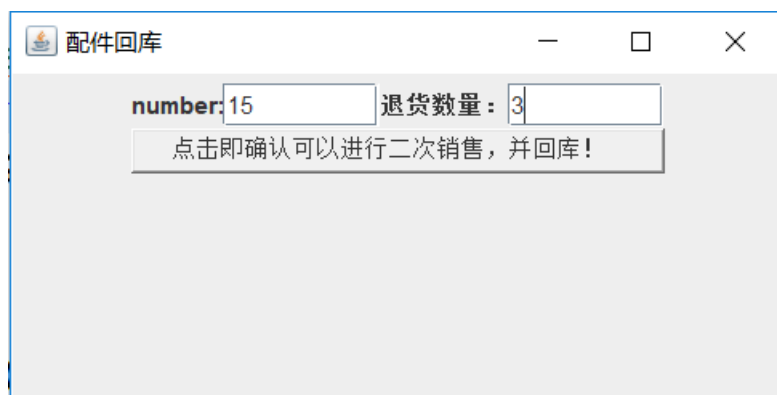


图 18 回库操作截图

点击下面的按钮后，出现如下界面：

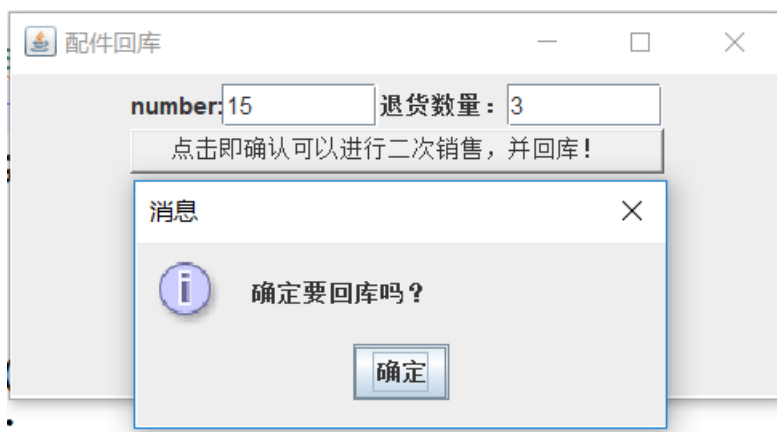


图 19 确定回库消息对话框截图

点击确定后，出现如下界面：



图 20 回库完成后截图

7.5 信息分析测试

点击菜单下面的仓库信息后，出现如下界面：

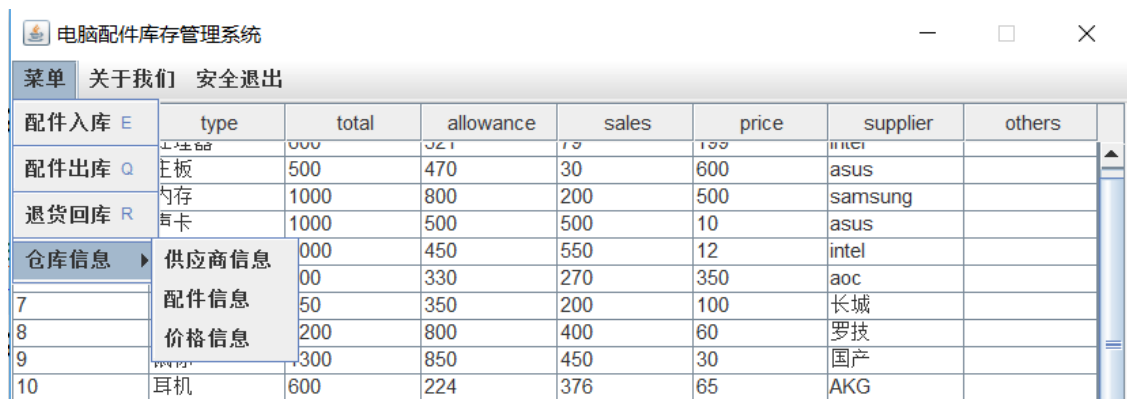
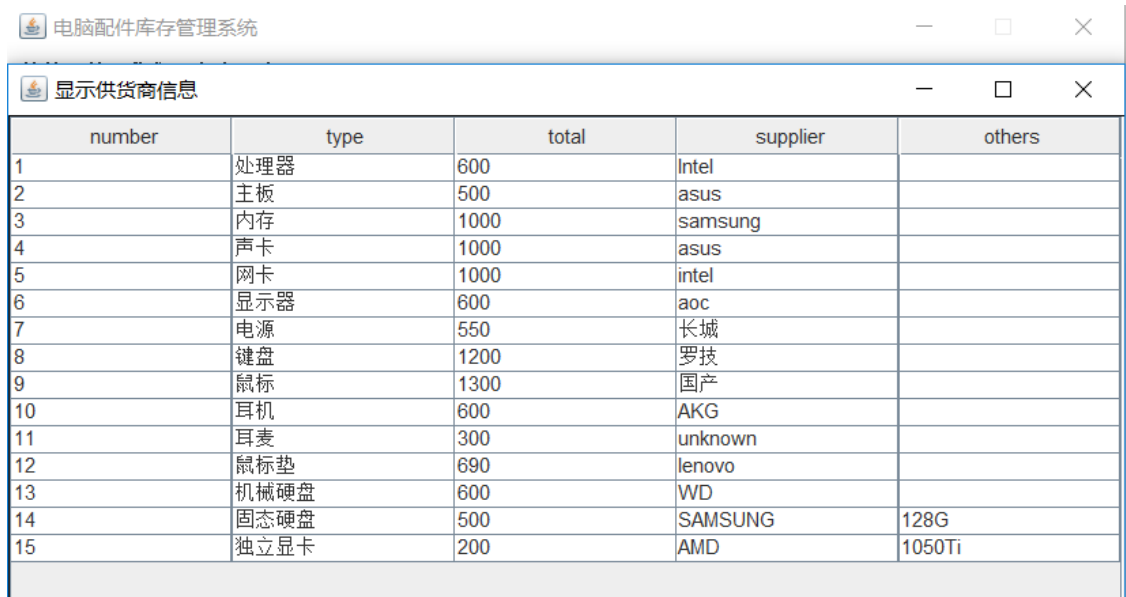


图 21 子菜单截图

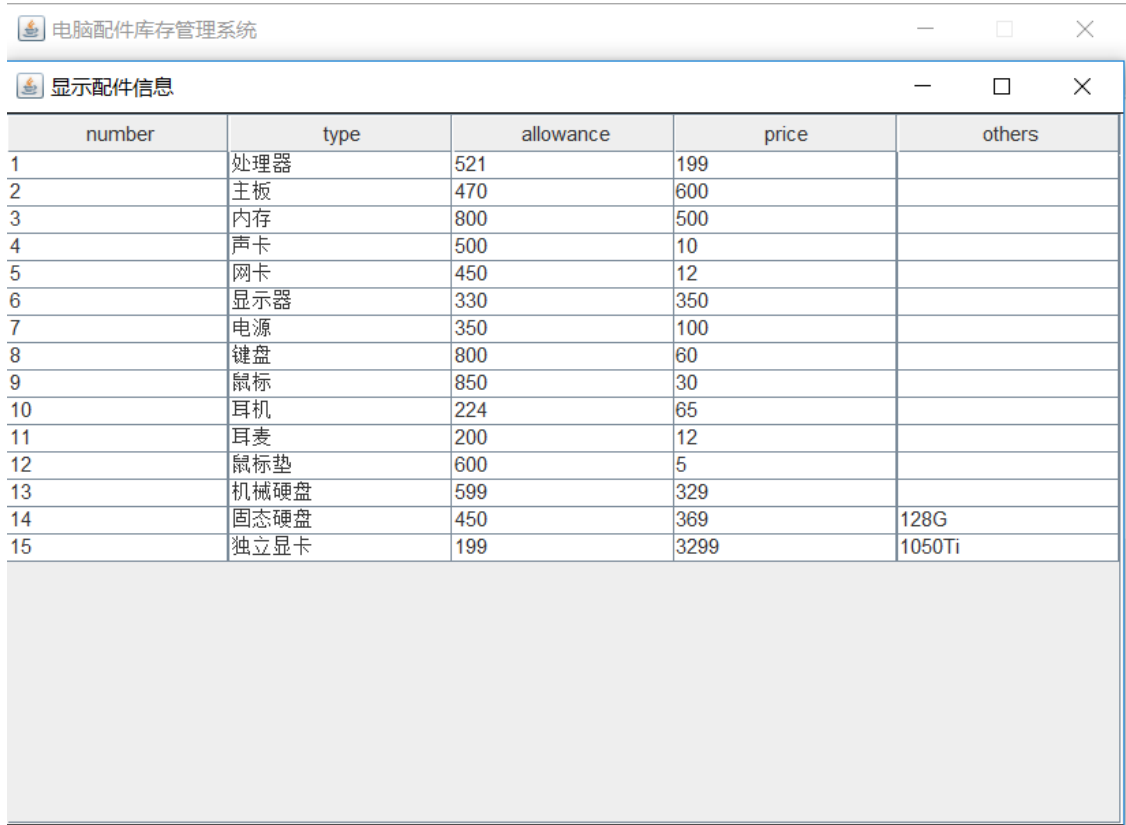
点击供应商信息后，出现如下界面：



number	type	total	supplier	others
1	处理器	600	Intel	
2	主板	500	asus	
3	内存	1000	samsung	
4	声卡	1000	asus	
5	网卡	1000	intel	
6	显示器	600	aoc	
7	电源	550	长城	
8	键盘	1200	罗技	
9	鼠标	1300	国产	
10	耳机	600	AKG	
11	耳麦	300	unknown	
12	鼠标垫	690	lenovo	
13	机械硬盘	600	WD	
14	固态硬盘	500	SAMSUNG	128G
15	独立显卡	200	AMD	1050Ti

图 22 供应商信息截图

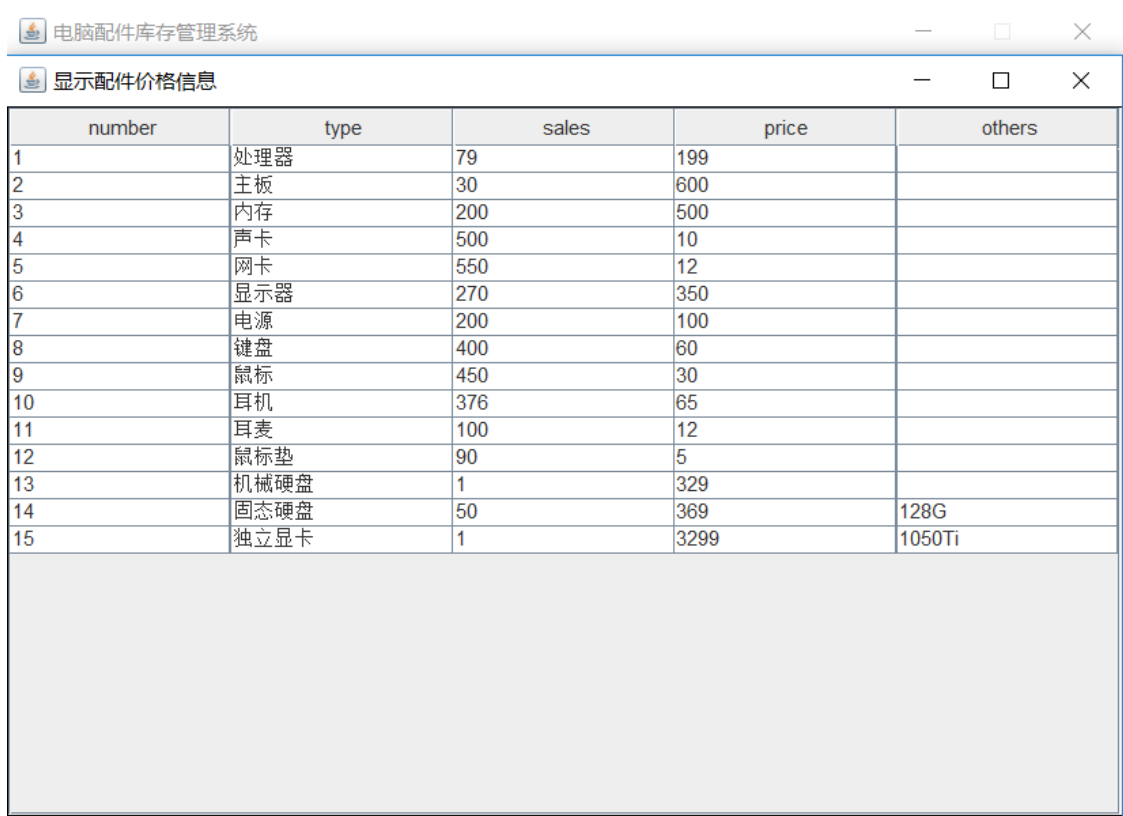
点击配件信息后，出现如下界面：



number	type	allowance	price	others
1	处理器	521	199	
2	主板	470	600	
3	内存	800	500	
4	声卡	500	10	
5	网卡	450	12	
6	显示器	330	350	
7	电源	350	100	
8	键盘	800	60	
9	鼠标	850	30	
10	耳机	224	65	
11	耳麦	200	12	
12	鼠标垫	600	5	
13	机械硬盘	599	329	
14	固态硬盘	450	369	128G
15	独立显卡	199	3299	1050Ti

图 23 配件信息截图

点击价格信息后，出现如下界面：



number	type	sales	price	others
1	处理器	79	199	
2	主板	30	600	
3	内存	200	500	
4	声卡	500	10	
5	网卡	550	12	
6	显示器	270	350	
7	电源	200	100	
8	键盘	400	60	
9	鼠标	450	30	
10	耳机	376	65	
11	耳麦	100	12	
12	鼠标垫	90	5	
13	机械硬盘	1	329	
14	固态硬盘	50	369	128G
15	独立显卡	1	3299	1050Ti

图 24 价格信息截图

7.6 自增（减）测试

在文本框区输入 14，

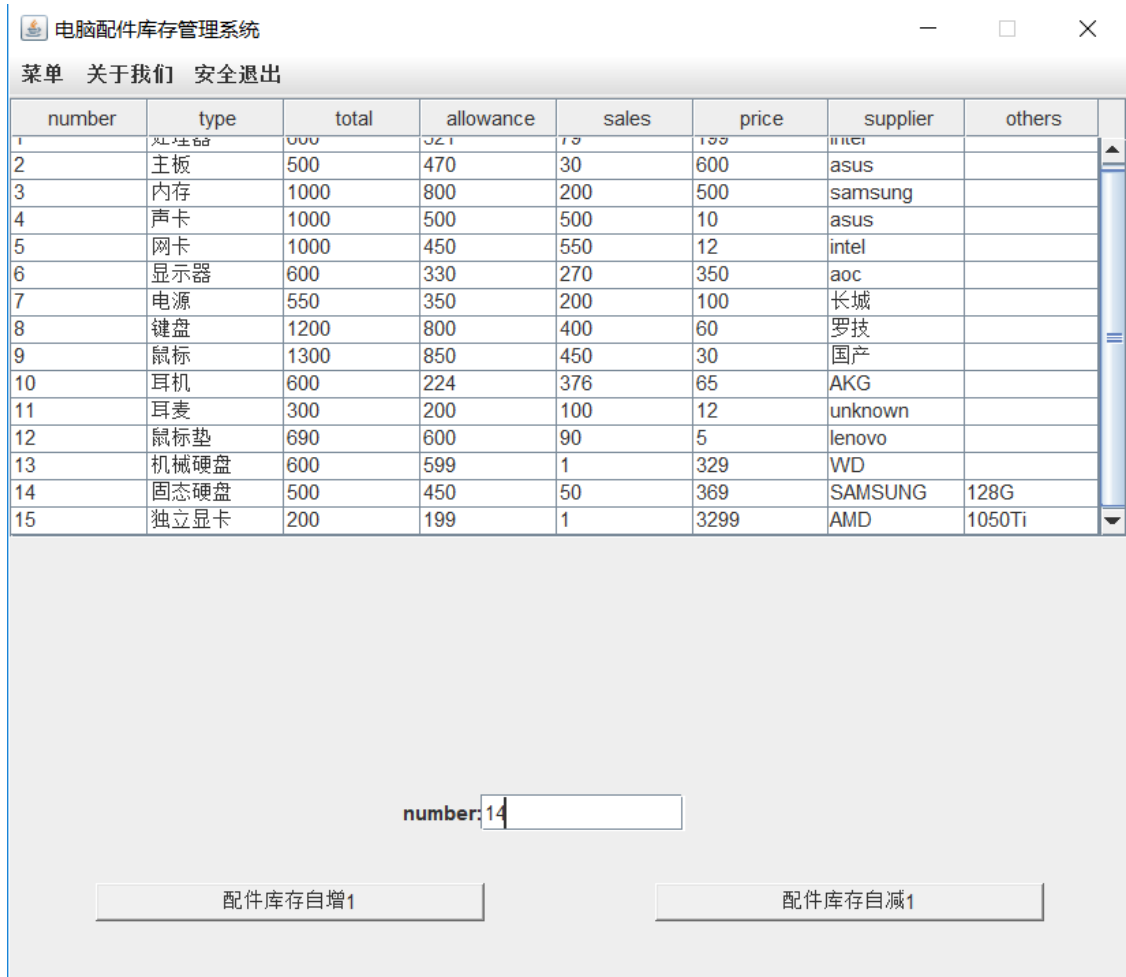


图 25 自增按钮截图

点击下面自增 1 按钮后，变成如下界面：

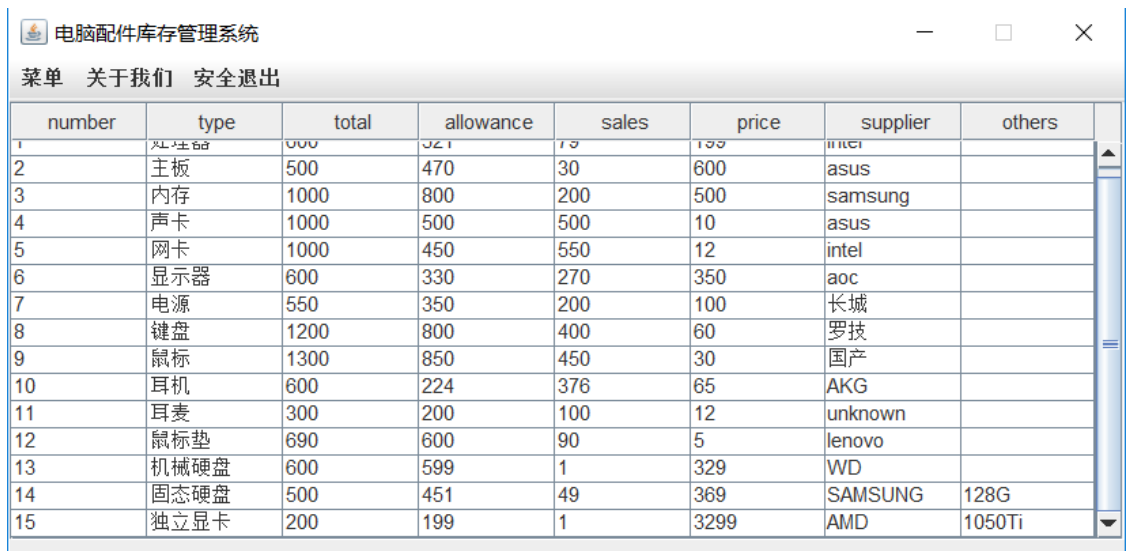


图 26 自增后截图

在文本框区输入 13，



图 27 自减按钮截图

点击下面自减 1 按钮后，变成如下界面：

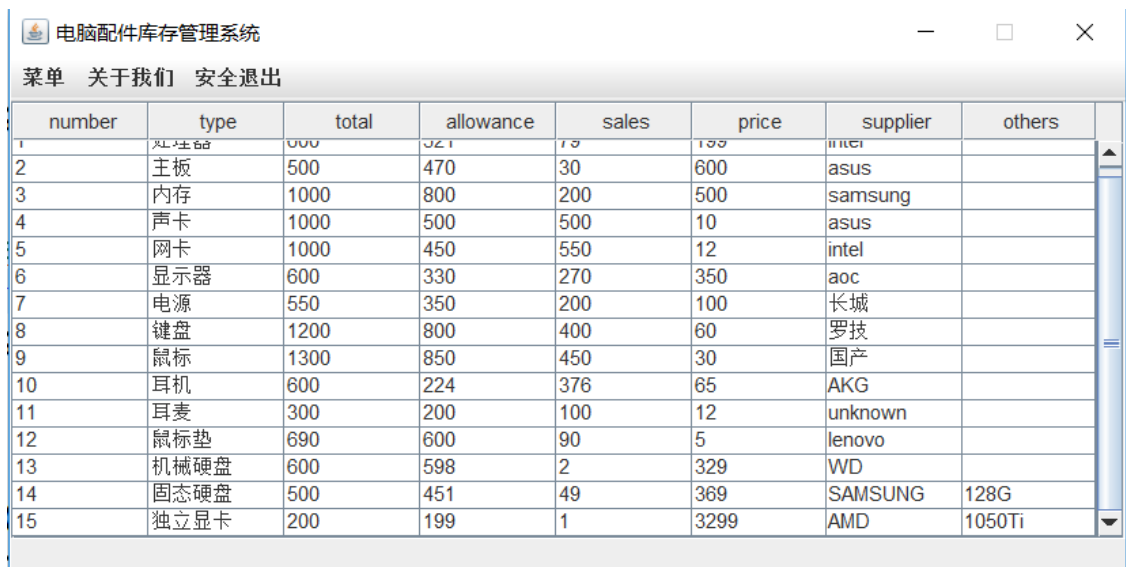


图 28 自减后截图

7.7 菜单测试

点击‘关于我们’后，出现如下界面：



图 29 ‘关于我们’ 菜单截图

点击‘安全退出’后，出现如下界面：



图 30 ‘安全退出’ 菜单截图

参考文献

- [1] 耿祥义. Java 2 实用教程(第 5 版) [M]. 北京:清华大学出版社, 2017.
- [2] 耿祥义. Java 课程设计(第 3 版) [M]. 北京:清华大学出版社, 2018.
- [3] 明日学院. Java 从入门到精通(项目案例版) [M]. 北京:水利水电出版社, 2017.
- [4] 何云清. MySQL 数据库程序设计 [M]. 北京:中国铁道出版社, 2018.
- [5] 明日科技. MySQL 从入门到精通 [M]. 北京:清华大学出版社, 2017.

心得体会

Java 语言是当今世界上最受程序开发人员使用的语言之一，连续多年在程序设计语言中排名前三；Mysql 语句也是当今世界上最受欢迎的数据库处理语句之一；面向对象的思想以及通过界面和窗口的交互方式，更是目前全球 IT 届比较成熟和广为接受思想和方式。此次的 Java 课程设计不仅要求我们将所学的知识学以致用，还尽力让我们去接触当今全球 IT 届比较热门思想和技术。这些思想和技术对自己以后的发展将会有很大的帮助。

通过这次课程设计，使我灵活应用所学数据结构知识，独立完成问题分析，结合数据结构理论知识，编写程序解决现实生活中的问题。初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；提高综合运用所学的理论知识和方法独立分析和解决问题的能力；并用系统的观点和软件开发一般规范进行软件开发，巩固、深化学生的理论知识，提高编程水平，并在此过程中培养了我严谨的科学态度和良好的工作作风。熟练掌握数据的存储表示和基本操作的实现，能够利用数据库语句和 Java 语言实现电脑配件管理系统。同时在这学期的实验中，培养了独立思考、动手操作的能力，在各种其它能力上也都有了提高。在实验课上，我学会了很多学习的方法。要面对社会的挑战，只有不断学习、实践，再学习、再实践。这对于我的将来也有很大的帮助。

虽然过程辛苦是不可避免，但收获还是令人感到尤其的欣慰。在这次的课程设计中不仅检验了我所学习的知识，也培养了我的实践能力，让我知道遇到一个问题，如何去寻找思路，如何去解决问题，最终完成整个事情。

课程设计是我们专业课程知识综合应用的实践训练，是我们迈向社会，从事职业工作前一个必不可少的过程。实验过程中，也十分感谢实验指导老师王峰老师的指点与教导。这次课程设计不仅是对这学期所学知识的一种综合检验，而且也是对自己动手能力的一种提高，增强了自己实践能力。通过这次课程设计使我明白了自己知识还比较欠缺，只是学习书本知识还是远远不够的，自己不会的东西还有太多，学习需要自己长期的积累，在以后的学习、工作中都应该不断的学习，将课本的理论知识与生活中的实践知识相结合，不断提高自己文化知识和实践能力。