

** 工 程 学 院

课 程 设 计

设计题目： 面向对象程序设计课程设计
小超市管理系统

系 别 信息工程系 班级 计本 081

学生姓名 _____

学 号 _____

指导教师 _____ 职称 副教授

起止日期：2010 年 6 月 21 日起——至 2010 年 7 月 2 日止

沈 阳 工 程 学 院

课程设计任务

1.设计的原始资料及依据

通过课程设计使学生进一步加深对课程内容的理解，利用程序设计的基本思想进行程序的开发，进一步掌握编程方法和技巧，提高学生用程序的思想来解决实际问题的能力。

2. 设计的主要内容及要求

本组设计题目：小超市管理系统

需要完成的功能：

- (1) 数据库的创建、登陆界面
- (2) 销售子系统
- (3) 权限管理子系统
- (4) 销售管理子系统
- (5) 商品管理子系统

3.对设计说明书撰写内容、格式、字数的要求

(1) 学生应撰写的内容为：中文摘要和关键词、目录、正文、参考文献等。课程设计说明书（论文）的结构及各部分内容要求可参照《沈阳工程学院毕业设计（论文）撰写规范》执行。应做到文理通顺，内容正确完整，书写工整，装订整齐。

(2) 装订格式

封面、任务书、成绩评审意见表、摘要和关键词、目录、正文、结论、致谢、参考文献。

(3) 课程设计说明书（论文）是体现和总结课程设计成果的载体，一般不应少于 3000 字。打印时采用 A4 纸，页边距均为 20mm，正文采用宋体小四号字，行间距 18 磅。文中大标题采用黑体小三号字，一级节标题采用黑体四号字，二级节标题采用黑体小四号字，三级节标题采用黑体小四号字，表题与图题采用宋体五号字。

4. 设计完成后应提交成果的种类、数量、质量等方面的要求

- (1) 课程设计任务书
- (2) 课程设计报告
- (3) 设计成品（光盘）

5.时间进度安排

顺序	阶段日期	计划完成内容	备注
1	第 1 周第 1 天	查阅资料，充分解读题目内容，小组分工，确定每个人的具体任务。	1 天
2	第 1 周第 2 天	数据库设计及登陆界面的实现。	1 天
3	第 1 周第 3, 4, 5 天, 第 2 周第 1 天	编制程序，单元测试。	4 天
4	第 2 周第 2, 3 天	程序整合，综合调试。	2 天
5	第 2 周第 4 天	答辩。	1 天
6	第 2 周第 5 天	撰写课程设计报告。	1 天

6.主要参考资料（文献）

- [1] 耿祥义,张跃平.Java2 实用教程(第三版).北京:清华大学出版社,2006
- [2] 张小波,曾锡山.Java 程序设计实训教程.冶金工业出版社.2006
- [3] 孙一林,彭波.Java 数据库编程实例.北京:清华大学出版社,2003
- [4] 严桂兰,刘甲耀,刘波.Java 编程模式与范例.机械工业出版社.2002
- [5] 朱仲杰.Java2 全方位学习. 机械工业出版社.2006
- [6] 张思民.JAVA 程序设计实践教程.北京：清华大学出版社，2006
- [7] 汤一平.Java 语言程序设计.北京：科学出版社，2006
- [8] 刘艺.Java 程序设计大学教程. 北京：机械工业出版社，2006

沈阳工程学院

面向对象程序设计课程设计成绩评定表

系（部）：信息工程系 班级：计本081 学生姓名：王浩 魏兴龙 王潇伯 夏恩亮

指导教师评审意见							
评价内容	具体要求	权重	评分				加权分
调研论证	能独立查阅文献,收集资料; 能制定课程设计方案和日程安排。	0.1	5	4	3	2	
工作能力态度	工作态度认真, 遵守纪律, 出勤情况是否良好, 能够独立完成设计工作。	0.2	5	4	3	2	
工作量	按期圆满完成规定的设计任务, 工作量饱满, 难度适宜。	0.2	5	4	3	2	
说明书的质量	说明书立论正确, 论述充分, 结论严谨合理, 文字通顺, 技术用语准确, 符号统一, 编号齐全, 图表完备, 书写工整规范。	0.5	5	4	3	2	
指导教师评审成绩 (加权分合计乘以 8)		分	加权分合计				
指导教师签名:		年 月 日					
评阅教师评审意见							
评价内容	具体要求	权重	评分				加权分
查阅文献	查阅文献有一定广泛性; 有综合归纳资料的能力	0.2	5	4	3	2	
工作量	工作量饱满, 难度适中。	0.5	5	4	3	2	
说明书的质量	说明书立论正确, 论述充分, 结论严谨合理, 文字通顺, 技术用语准确, 符号统一, 编号齐全, 图表完备, 书写工整规范。	0.3	5	4	3	2	
评阅教师评审成绩 (加权分合计乘以 4)		分	加权分合计				
评阅教师签名:		年 月 日					
答辩小组评审意见							
评价内容	具体要求	权重	评分				加权分
学生汇报	汇报准备充分, 思路清晰; 语言表达准确, 概念清楚, 论点正确, 有层次, 有重点, 基本上反映了所完成任务的全部内容; 时间符合要求。	0.5	5	4	3	2	
答辩	思路清晰; 回答问题有理论依据, 基本概念清楚; 主要问题回答准确, 深入, 有说服力。	0.5	5	4	3	2	
答辩小组评审成绩 (加权分合计乘以 8)		分	加权分合计				
答辩小组教师签名:		年 月 日					
课程设计总评成绩		分					

摘要

随着小型超市规模的发展不断扩大，商品数量急剧增加，有关商品的各种信息量也成倍增长，传统的人工记忆方式也慢慢的无法适应形势的变化。随着信息技术的发展，计算机已被广泛的用于社会的各个领域，成为推动社会发展的技术动力。而在计算机应用中，软件的作用十分突出，软件已经发展成为信息技术的核心，主导着信息产品的开发和信息技术市场的进一步的开拓。软件产业已成为社会信息化进程中的一个战略性产业。在软件技术的应用中软件的开发技术尤其是应用型软件产品的开发技术成了重中之重。不断开发适应用户需求、市场需要的新型软件产品。随着社会的发展，软件也在不断的更新换代。

由于有些超市里对商品的进货、售出及商品的基本信息等的一些繁琐处理都是通过手工进行记载并处理的，对商品信息的处理工作量特别大，不仅浪费大量的人力物力，而且还很容易出错，更重要的是为了方便顾客，超市需要时时刻刻对商品对各种信息进行统计分析。一直以来人们使用传统人工的方式管理超市的各项工作，这种管理方式存在着许多缺点，如：效率低、保密性差，另外时间一长，将产生大量的冗余文件和数据，这给查找、统计、更新和维护工作都带来了很大的困难。然而大型的超市管理系统功能过于强大而造成操作繁琐降低了小超市的工作效率。如果有这样一套完整的小型超市管理系统软件，只需点击几下鼠标就可完成所需操作，那样就会大大的提高超市员工的工作效率和管理水平，并能进一步提高超市的工作效率并使超市实施规范化管理。鉴于小型超市管理中存在的诸多问题，我们在此次课程设计中以小型超市管理系统为研究课题，对此系统进行了全面的设计。

小型超市管理系统由销售界面、销售管理、商品管理、权限管理四部分组成。它的内容对于超市的销售人员和管理者乃至顾客来说都至关重要，所以小型超市管理系统应该能够为用户提供充足的信息和快捷的查询手段。

一个完整系统的设计，当然就离不开后台环境的支持，在此次设计中，我们主要以 Access 数据库作为基础，实行此系统的开发。

关键字 小型超市管理，商品销售，java 版小超市管理系统， Access 数据库

目 录

摘 要.....	I
第 1 章 绪 论.....	1
第 2 章 系统功能介绍.....	2
2.1 系统功能总框图.....	2
2.2 本系统模块功能介绍.....	2
2.2.1 登录功能介绍.....	2
2.2.2 商品销售功能介绍.....	3
2.2.3 权限管理功能介绍.....	3
2.2.4 销售管理功能介绍.....	4
2.2.5 商品管理功能介绍.....	4
2.3 组内任务分工.....	5
第 3 章 数据库设计.....	6
3.1 数据库需求分析.....	6
3.2 数据库概念结构设计.....	6
3.3 数据库表的设计.....	7
第 4 章 系统功能实现.....	8
4.1 登录功能实现.....	8
4.2 销售界面功能介绍.....	11
4.2.1 搜索商品的信息.....	11
4.2.2 购买商品.....	11
4.2.3 模块的核心代码.....	12
4.3 权限管理功能介绍.....	18
4.3.1 添加用户功能介绍.....	18
4.3.2 修改用户功能介绍.....	19
4.3.3 删除用户功能介绍.....	19
4.3.4 显示用户信息与刷新按钮介绍.....	20
4.3.5 用户管理的实现代码.....	20
4.4 销售管理功能介绍.....	27
4.4.1 编号查询功能.....	27
4.4.2 账单查询功能.....	27
4.4.3 销售管理模块的核心代码.....	28
4.4 商品管理功能介绍.....	31
4.4.1 查询信息.....	31
4.4.2 添加信息及提交添加.....	32
4.4.3 删除信息及提交删除.....	32
4.4.4 修改信息及提交修改.....	33
4.4.5 关于.....	33
4.4.6 模块的核心代码.....	34
总 结.....	39
致 谢.....	40
参考文献.....	41

第1章 绪 论

随着人们生活水平的不断提高，购物已成为一种时尚。每天都有大量的消费者在各大商场中留下消费信息，所以作为商场的管理人员就需要有一个自动化、智能化的管理系统来完成这些信息的处理。由此我们设计开发小超市管理系统。

目前市场上的超市管理系统，大部分为基于分布式数据库的网络管理系统，对于规模较小的超市没有必要花巨资来购买这样功能全面的管理系统，此外，对于那样功能齐全的管理系统也需要花大精力来维护。基于这种情况，我们用所学的 java 知识，可以开发一种既能节约资金，又能完成小超市日常的管理任务。

本系统所包括的功能主要有：限于不同身份的人登录的登录界面；权限管理功能；商品销售功能；商品管理功能；销售管理功能。

在系统的设计中，用户管理功能可以管理登录本系统的人员，如：管理员、用户和超级用户等。在最初的系统使用中只有技术管理员一种身份。商品销售功能：可以完成商品信息的查询。如：输入一种商品的编号在商品信息栏就显示该商品的所有信息，名称、价格、生产日期、生产地。然后选择购买功能，就可以将商品添加到购物信息栏，最后选择提交，所购买的商品信息同时也添加到数据库中。在商品管理功能中，可以把一种新引进的商品信息添加到小超市管理系统中。在销售管理功能中，可以查看超市的销售情况，所有商品的购买信息都存储在数据库中。

在系统的安全性方面，我们规定了不同权限的登录用户，管理员（主要负责用户管理）可以登录到任何一个管理模块，以及后台的数据库，能够改变任一个登录者的用户名和密码。用户只能进入到销售界面。老板超级主要查看商品的销售情况和商品管理，他可以进入到商品销售管理查看商品的销售情况。通过这样的权限限制就可以方便的控制系统的安全性。

总之，小超市管理系统是一个经济、实惠的应用软件，适合小规模的小商店和超市。操作难度小，易学易用。

第2章 系统功能介绍

2.1 系统功能总框图

小超市管理系统是一个专门针对小型的商店和超市的智能化、自动化的管理系统。其功能总框图如图 2-1 所示。

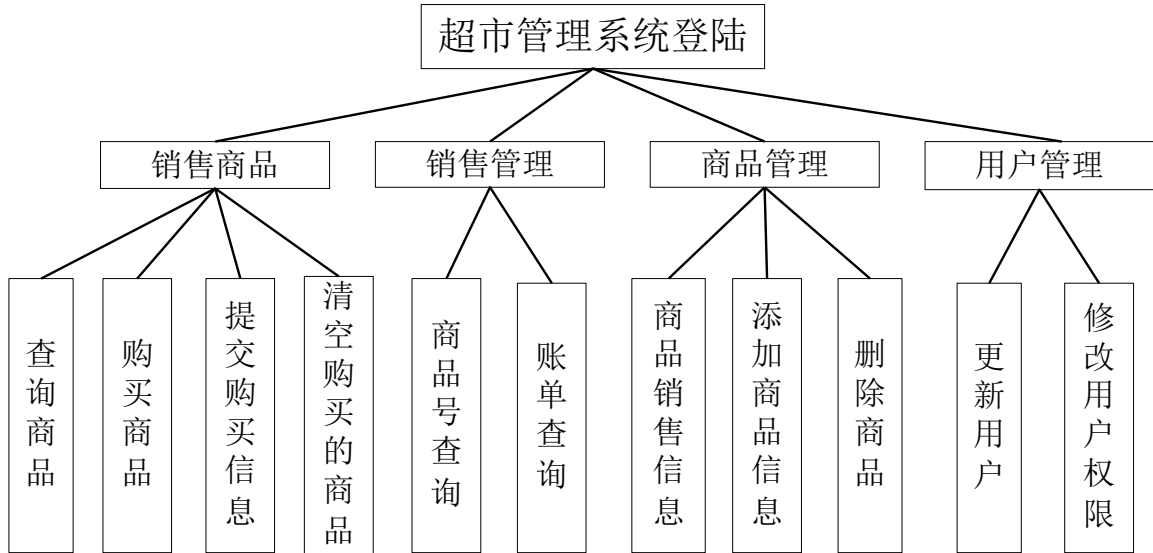


图 2-1 系统功能总框图

2.2 本系统模块功能介绍

2.2.1 登录功能介绍

登录功能是进入系统必须经过的验证过程，其主要功能是验证使用者的身份，确认使用者的权限，从而在使用软件过程中能安全地控制系统数据，即不同的工作人员有不同的权限，每个使用人员不得跨越其权限操作软件，可以避免不必要的数据库丢失事件发生。

登录的界面如图 2-2 所示：

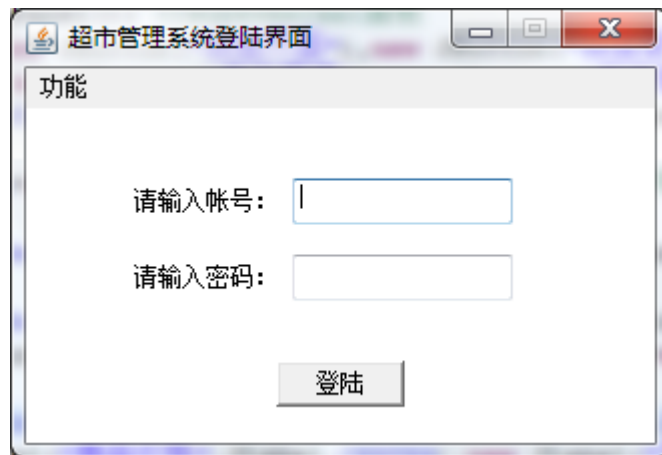


图 2-2 登录的界面

2.2.2 商品销售功能介绍

销售界面是本系统直接供销售的店员应用的专管销售功能的界面，首先要查询欲购买的商品，在左侧的查询框内输入商品代号，然后点击搜索按钮，相应的商品信息将显示在“商品信息”一栏中，确认是此商品后点击购买，购买的商品将出现在下面的购物信息一栏中，当所有欲购买的商品都已输入好，点击提交，购买的商品信息将记录到数据库中。若输入有误，点击清除按钮，刚输入的信息将被清除，可重新输入。界面的右下方有商品总价的标签，可以时时地显示已购买商品的总价，以供参考。商品销售功能的主界面如图 2-3 所示：

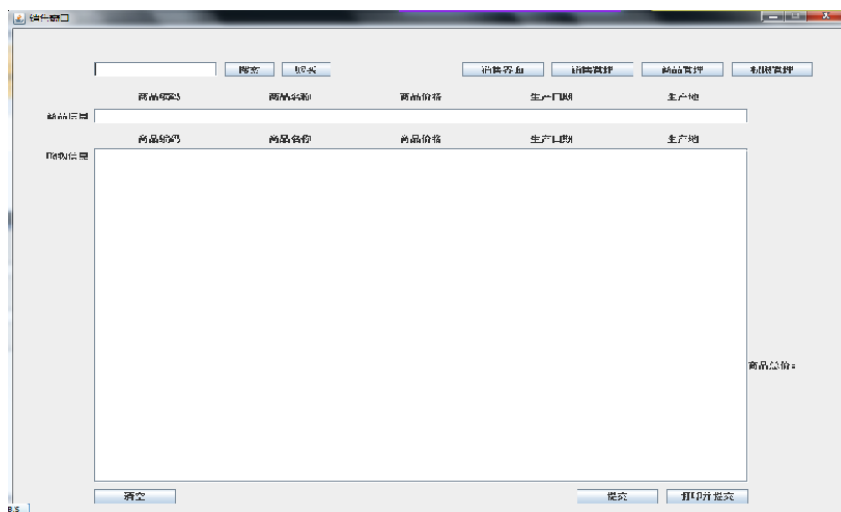


图 2-3 销售窗口的主界面

2.2.3 用户管理功能介绍

用户管理功能是管理人员使用的界面。管理人员可以使用此功能管理所有登录用户的信息。可以修改所有用户的帐号和密码，设置用户的权限，删除用户的帐号；对于已存在的用户技术管理员可以查询他们的信息，在下面的显示用户信息处显示。添加、修改、删除等操作后，单击“刷新”后，新的更改信息才可以在下面的用户信息显示处显示。

用户管理的主界面如图 2-4 所示：

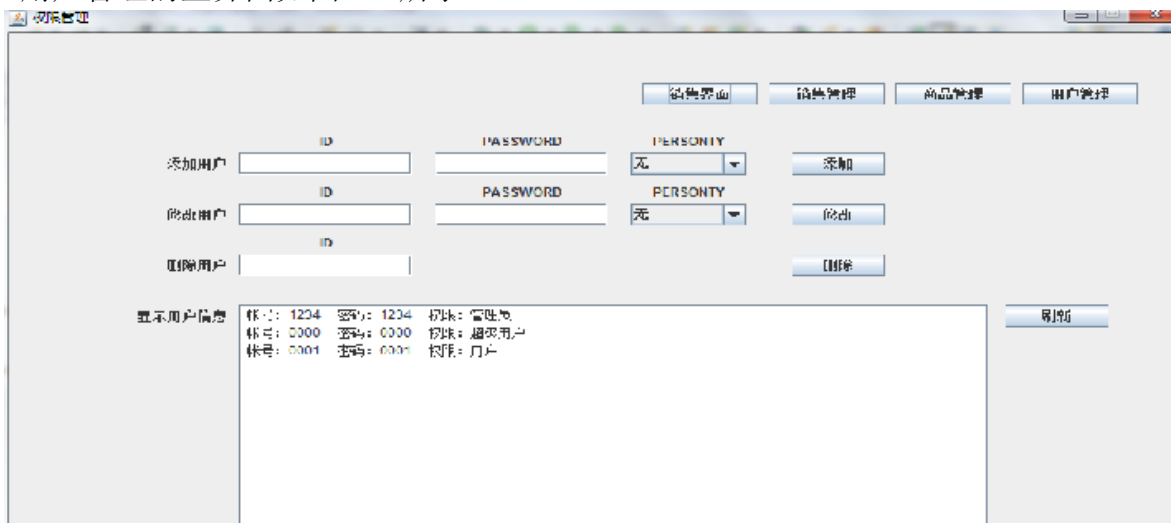


图 2-4 用户管理的主界面

2.2.4 销售管理功能介绍

销售管理功能主要是老板查看超市商品销售情况。老板可以根据已售出的商品号查询该商品的信息，如可以查询该商品已售出的总数量。还可以根据销售的账单号查询，就是在消费者每一次购买商品时，就分配一个账单号，如出现消费者退货情况时，老板可以查询账单号解决。

“按编号查询”输入商品的编号，在下面的商品信息处显示该商品的信息；“按账单号查询”输入商品的账单号，在下面的商品信息处显示该商品的信息。

销售管理功能的主界面如图 2-5 所示：

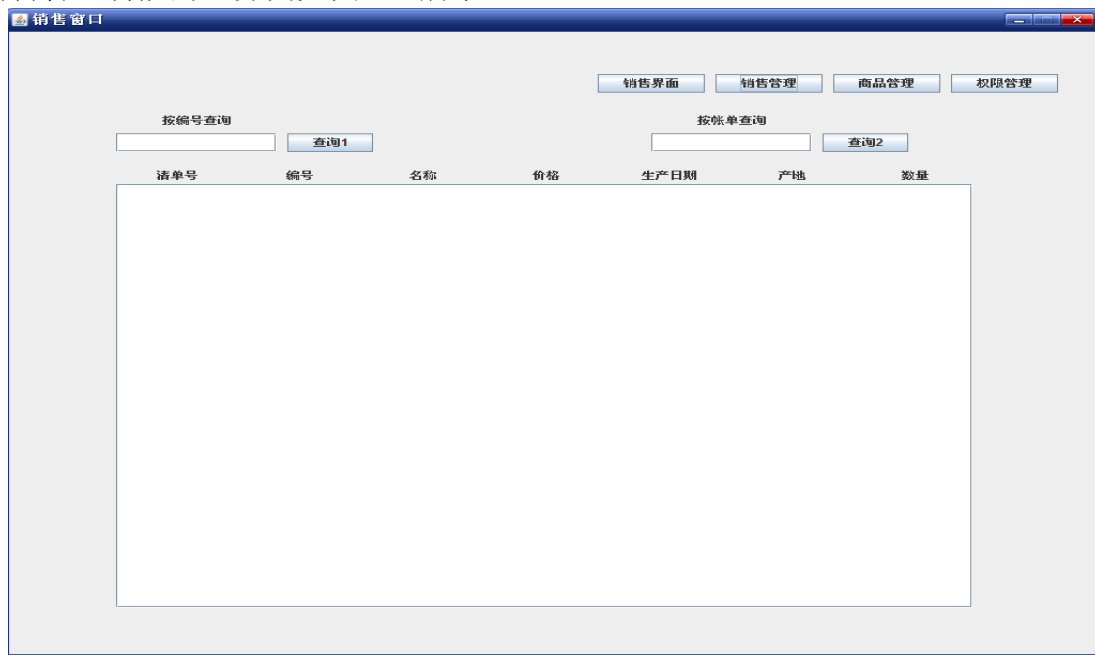


图 2-5 商品管理功能主界面

2.2.5 商品管理功能介绍

商品管理的功能是店员在进货、商品信息维护时所需要的界面，此界面功能是管理所有商品的信息，在这里可以添加、修改、删除任意商品信息，做到对商品信息的及时维护。

应用时，对应按钮的功能，进行应用，维护方便。

商品管理功能的主界面如图 2-6 所示：



图 2-6 商品管理系统的主界面

2.3 组内任务分工

组内人员分工如表 2-1 所示：

表 2-1 组内分工

组内分工	姓名	主要任务	成绩
组长	王浩	数据库的设计、商品管理和用户管理等功能	
组员	魏兴龙	销售管理界面及功能	
组员	王潇伯	商品销售界面及功能	
组员	夏恩亮	权限销售界面及功能	

第3章 数据库设计

在每个管理系统中都一定有储存机制，若单单以文件形式储存，查找和存储时速度比较缓慢，会降低系统的整体运行速度，一定要运用数据库的进行信息管理，所以我们的“小超市管理系统”运用了 Microsoft Access 数据库，存储其后台数据。下面将介绍本系统的数据库设计。

3.1 数据库需求分析

此系统需要有两个实体，账户和商品。
数据项如表 3-1 和 3-2 所示。

表 3-1 用户信息数据项

数据项名	数据项含义说明	别名	数据类型	长度
ID	用户登录的帐号	Id	字符型	50
PASSWORD	用户登录帐号对应的密码	Psw	字符型	50
PERSONTY	登录用户的权限	Pst	字符型	50

表 3-2 商品信息数据项

数据项名	数据项含义说明	别名	数据类型	长度
NUM	商品的编号	编号	字符型	50
NAME	商品的名称	名称	字符型	50
TIME	商品的进货时间	货时	字符型	50
ADD	商品的生产地址	产地	字符型	50
PRICE	商品的单价	价格	整型	50
COUNT	商品的库存数量	数量	整型	

3.2 数据库概念结构设计

根据需求分析设计出 E-R 图如图 3-1 所示:

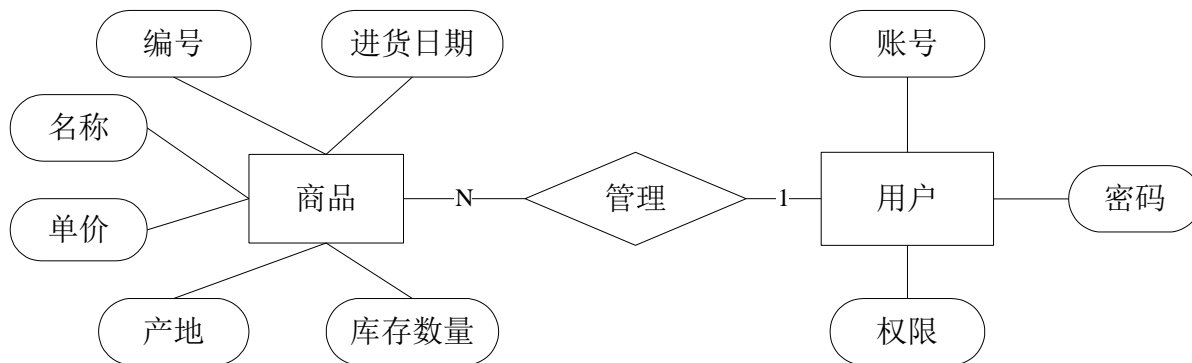


图 3-1 系统 E-R 图

3.3 数据库表的设计

Password 表，其中存储的是用户的帐号、密码及权限，用于登陆系统时确认身份。此表设计如 3-3 表所示：

表 3-3 Password 信息表

字段名称	数据类型	长 度	备 注
ID	文 本	50	
PASSWORD	文 本	50	
PERSONTY	文 本	50	

Goods 表，其中存储的是商品的详细信息，作用是记录商品的信息及在本超市中的数量。此表设计如 3-4 表所示。

表 3-4 Goods 信息表

字段名称	数据类型	长 度	备 注
NUM	文 本	50	
NAME	文 本	50	
PRICE	整 型		
TIME	文 本	50	
ADD	文 本	50	
COUNT	文 本	50	

Information 表，其中存储的是顾客在超市中的消费情况，其中有帐单号、商品代号、数量及总价，用来记录超市中销售情况，此表设计如 3-5 表所示。

表 3-5 Information 信息表

字段名称	数据类型	长 度	备 注
编号	长 整 型		销售的每件商品有不同的编号
LISTNUM	文 本	50	每提交一次帐单帐单号增加 1
GOODSNUM	文 本	50	
ALLPRICE	整 型		每个帐单的最后合出此帐单的总价

第4章 系统功能实现

4.1 登录功能实现

登录界面是使用者在使用此软件时需要输入自己的账号和密码，从而使用自己的权限来管理超市的运行。

小超市登录界面

如 4-1 所示。

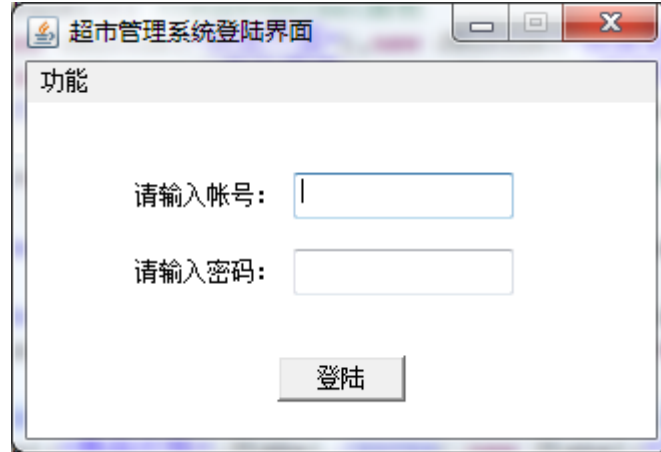


图 4-1 小超市管理系统登录界面

模块核心代码:

```
public class Login extends Frame implements ActionListener,WindowListener //登陆界面
{
    public int screanWidth,screanHeight;
    String idin=null;
    String keyin=null;
    String passwordD=null;
    public static String persontyD=null;
    MenuBar menubar;
    Menu menu;
    MenuItem quet,help;
    Label id,key;
    TextField idtf,keytf;
    Box boxV1,boxV2,boxV3,baseBox,boxx;
    Button enterB;
    Login()
    {
        setTitle("超市管理系统登陆界面");
        Toolkit tool=getToolkit();
        Dimension dim=tool.getScreenSize();
        screanWidth=dim.width;
        screanHeight=dim.height;
        setBounds(dim.width/3,dim.height/3,320,215);
        menubar=new MenuBar();
        menu=new Menu("功能");
        help=new MenuItem("帮助");
```

```
quet=new MenuItem("退出");
quet.setShortcut(new MenuShortcut(KeyEvent.VK_E));
quet.addActionListener(new ActionListener()           //匿名类实例控制
    {
        public void actionPerformed(ActionEvent p){
            System.exit(0);
        }
    });

menu.add(help);
menu.add(quet);
menubar.add(menu);
setMenuBar(menubar);
id=new Label("请输入帐号: ",Label.RIGHT);
key=new Label("请输入密码: ",Label.RIGHT);
idtf=new TextField(10);
keytf=new TextField(10);
keytf.setEchoChar('*');
enterB=new Button("登陆");
boxV1=Box.createVerticalBox();
boxV1.add(Box.createVerticalStrut(35));
boxV1.add(id);
boxV1.add(Box.createVerticalStrut(15));
boxV1.add(key);
boxV1.add(Box.createVerticalStrut(15));
boxV2=Box.createVerticalBox();
boxV2.add(Box.createVerticalStrut(35));
boxV2.add(idtf);
boxV2.add(Box.createVerticalStrut(15));
boxV2.add(keytf);
boxV2.add(Box.createVerticalStrut(15));
baseBox=Box.createHorizontalBox();
baseBox.add(Box.createHorizontalStrut(25));
baseBox.add(boxV1);
baseBox.add(Box.createHorizontalStrut(5));
baseBox.add(boxV2);
baseBox.add(Box.createHorizontalStrut(70));
boxV3=Box.createHorizontalBox();
boxV3.add(Box.createHorizontalStrut(125));
boxV3.add(enterB);
boxV3.add(Box.createHorizontalStrut(125));
boxx=Box.createVerticalBox();
boxx.add(baseBox);
boxx.add(Box.createVerticalStrut(15));
boxx.add(boxV3);
boxx.add(Box.createVerticalStrut(70));
add(boxx);
enterB.addActionListener(this);
```



```

        addWindowListener(this);
        setResizable(false);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        Connection con;
        Statement sql;
        ResultSet rs;
        if (e.getSource()==enterB)
        {
            idin=idtf.getText();
            keyin=keytf.getText();
            try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); }
            catch (ClassNotFoundException f)
            {System.out.println(""+f);}
            try
            {con=DriverManager.getConnection("jdbc:odbc:q","", "");}
            sql=con.createStatement();
            rs=sql.executeQuery("SELECT * FROM password where ID='"+idin+"'");
            while(rs.next()){
                passwordD=rs.getString(2);
                persontyD=rs.getString(3);}
            con.close();}
            catch (SQLException g)
            {System.out.println(g);}
            if (keyin.equals(passwordD))
            {View2 frame=new View2();
                this.setVisible(false); }
            else {JOptionPane.showMessageDialog(this,"
请重新输入","提示",JOptionPane.WARNING_MESSAGE); } } }
        public void windowActivated(WindowEvent o)
        {validate();}
        public void windowDeactivated(WindowEvent o)
        {setBounds(screanWidth/3,screanHeight/3,320,215);
            validate();}
        public void windowClosing(WindowEvent o)
        {dispose();}
        public void windowClosed(WindowEvent o)
        {System.exit(0);}
        public void windowIconified(WindowEvent o)
        {}
        public void windowDeiconified(WindowEvent o)
        {setBounds(screanWidth/3,screanHeight/3,320,215);
            validate();}
        public void windowOpened(WindowEvent o){ }
    }
}

```

帐户或密码错误 \n

4.2 销售界面功能介绍

该功能是本系统的最关键的模块，是消费者直接使用的功能。消费者可以根据已看到的商，在搜索栏中打入商品的编号，进行查询商品的信息，然后可以依据自己的需要购买。最后单击提交，购买的商品就记录到系统的数据库中。

销售的主界面如图 4-2 所示：

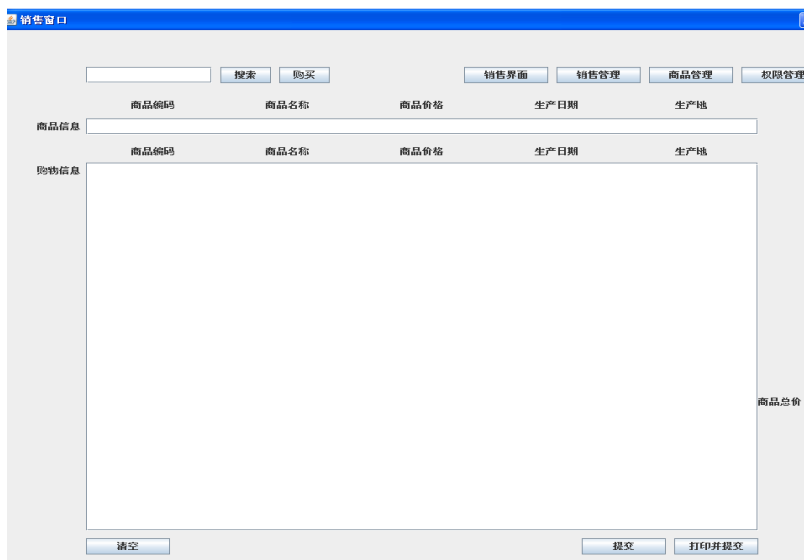


图 4-2 销售的主界面

4.2.1 搜索商品的信息

消费者可根据自己所需要的商品编号，在销售功能中搜索，查看商品的信息无误后，确定是否购买。如搜索 001 号商品，查看它的信息，如图 4-3 所示。

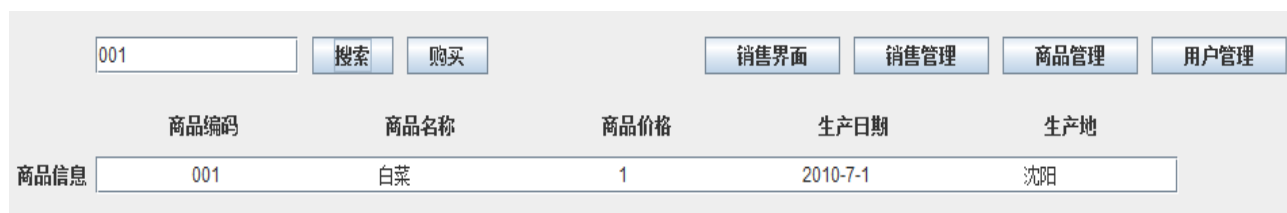


图 4-3 搜索 001 号商品的信息

4.2.2 购买商品

在确定自己需要的商品后，然后就是购买商品，消费者可以按“购买”键进行购买物品，购买一份就单击一次“购买”键，购买物品的信息显示在购物信息框中，最后单击“提交”，消费者所购买的物品就作为一个账单号存储在数据库中。

如购买 004、005、006、008、010 和 011 号商品时，购买信息如图 4-4 下：

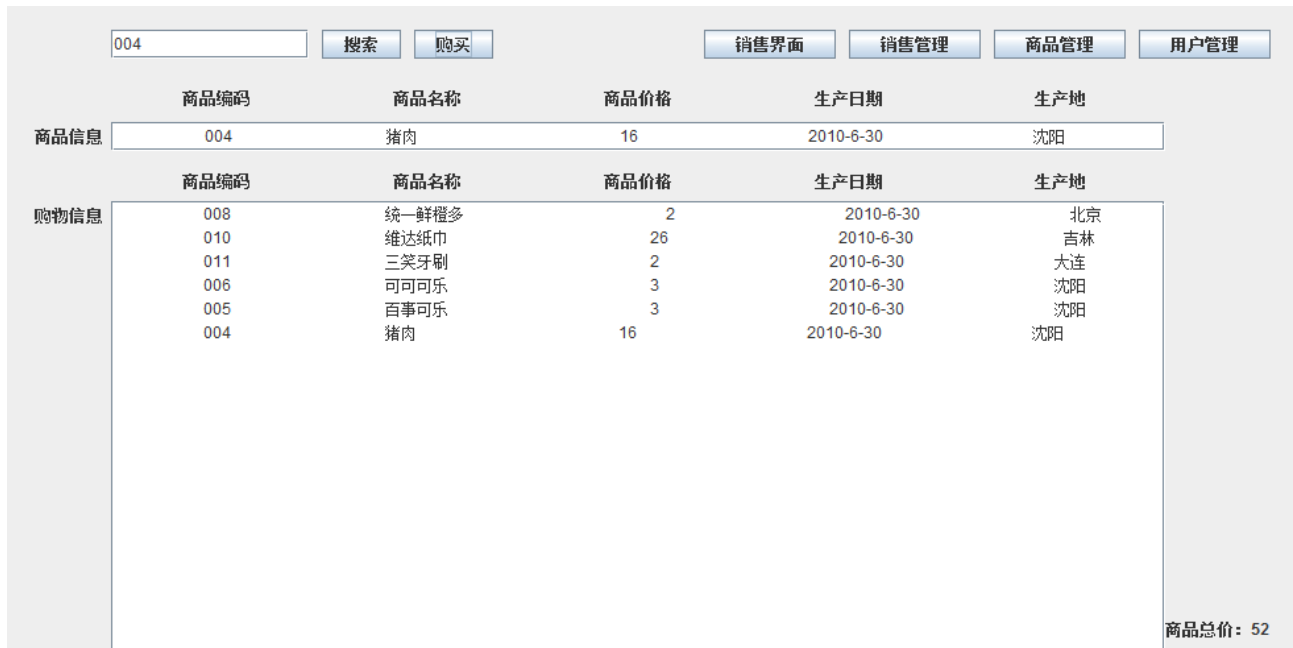


图 4-4 购买信息

当购买人不想购买这些商品时，可以点击窗口左下角的清除按钮，来清除刚才购买的商品，数据库中也不会有此记录。

此外，本系统是每购买一件商品向数据库提交一次记录，当未点击提交按钮之前，若要结束此窗口进程，我们在代码中嵌入了一段删除当前帐单的数据库记录代码，以达到没有误提交的操作。

4.2.3 模块的核心代码

(1) 销售界面的实现

```
public class View2 extends JFrame implements ActionListener //销售界面
{
    String num=null,name=null,time=null,add=null,t1=null,t2=null,t3=null;
    int price=0;
    int n1,n2,ap,n3;
    private JPanel sM=new JPanel(); //supermarket 面板
    private JButton[] pS={new JButton("销售界面"),new JButton("销售管理"),
        new JButton("商品管理"),new JButton("权限管理")}; //personty 按钮
    private JTextArea goodsShow=new JTextArea(), searchShow=new JTextArea(); //文本区

    private JScrollPane jsp=new JScrollPane(goodsShow); //滚动区域 设定内容为 商品显示的文本区
    private JTextField numSearch=new JTextField(),showGoods=new JTextField(); //商品查询/显示商品
    private JButton searchGoods=new JButton("搜索"),buyGoods=new JButton("购买"),
        clean=new JButton("清空"), ok=new JButton("提交"),print=new JButton("打印并提交"); //查询
    按钮
    private JLabel[] goodsInformation={new JLabel("商品编码",JLabel.CENTER),new JLabel("商品名称
```

```
    ",JLabel.CENTER),
        new JLabel(" 商品 价格 ",JLabel.CENTER),new JLabel(" 生 产 日 期
",JLabel.CENTER),new JLabel("生产地",JLabel.CENTER) };
    private JLabel[] goodsI={new JLabel(" 商品 编 码 ",JLabel.CENTER),new JLabel(" 商品 名 称
",JLabel.CENTER),
        new JLabel(" 商品 价格 ",JLabel.CENTER),new JLabel(" 生 产 日 期
",JLabel.CENTER),new JLabel("生产地",JLabel.CENTER) } ;
    private JLabel allPrice=new JLabel(" 商品 总 价 : \n ",JLabel.LEFT),label1=new JLabel(" 商品 信 息
",JLabel.RIGHT),label2=new JLabel("购物信息  ",JLabel.RIGHT);
    public View2()
    {
        sM.setLayout(null);
        for(int i=0;i<4;i++)
        {
            pS[i].setBounds(550+i*110,50,100,22);
            sM.add(pS[i]);
            pS[i].addActionListener(this);
        }
        jsp.setBounds(100,180,800,500);
        goodsShow.setLineWrap(true)
        numSearch.setBounds(100,50,150,22);
        searchGoods.setBounds(260,50,60,22);
        searchGoods.addActionListener(this);
        buyGoods.setBounds(330,50,60,22);
        buyGoods.addActionListener(this);
        showGoods.setBounds(100,120,800,22);
        for (int b=0;b<5 ;b++ )
        {
            goodsInformation[b].setBounds(100+160*b,87,160,30);
            sM.add(goodsInformation[b]);
        }
        for (int b=0;b<5 ;b++ )
        {
            goodsI[b].setBounds(100+160*b,150,160,30);
            sM.add(goodsI[b]);
        }
        clean.setBounds(100,690,100,22);
        clean.addActionListener(this);
        ok.setBounds(690,690,100,22);
        ok.addActionListener(this);
        print.setBounds(800,690,100,22);
        print.addActionListener(this);
        allPrice.setBounds(900,480,100,50);
        label1.setBounds(20,120,80,22);
        label2.setBounds(20,180,80,22);
        sM.add(label1);
```

```
sM.add(label2);
sM.add(allPrice);
sM.add(print);
sM.add(ok);
sM.add(showGoods);
sM.add(clean);
sM.add(numSearch);
sM.add(searchGoods);
sM.add(buyGoods);
sM.add(jsp);
addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
    Connection con;
    Statement sql;
    ResultSet rs;
    try
    {
        if (n3!=0)
        {
            int a1=0;
            con=DriverManager.getConnection("jdbc:odbc:q","","");
            sql=con.createStatement();
            rs=sql.executeQuery("SELECT MAX(编号) from information");
            while(rs.next())
            {
                a1=rs.getInt(1);}
            for (int i=0;i<n3 ;i++ )
            {
                sql.executeUpdate("delete from information where 编号="+a1-i);
            }
            con.close();
            ap=0;
            num=null;name=null;price=0;time=null;add=null;n3=0;
            goodsShow.setText(null);
        }
    }
    catch (SQLException g)
    {System.out.println(g);
    }
    System.exit(0);
}
});
this.add(sM);
this.setTitle("销售窗口");
this.setResizable(false);
this.setBounds(0,0,1024,768);
```

```

    this.setVisible(true);
    validate(); }

```

(2) 购买商品功能的代码

```

if(p.getSource()==buyGoods) //购买按钮
{
    try
    {
        if (num!=null)
        { con=DriverManager.getConnection("jdbc:odbc:q","","");
          sql=con.createStatement();
          sql.executeUpdate("insert into information(listnum,goodsnum) values("+n2+", "+num+"");
          rs=sql.executeQuery("SELECT * FROM information where listnum="+n2);
          n3++;//记录购买商品个数
          con.close();
          goodsShow.append("                "+num+"
"+name+"                "+price+"
"+time+"                "+add+"\n"); }
        else
            {JOptionPane.showMessageDialog(this," 请 选 择 商 品 "," 提 示
",JOptionPane.WARNING_MESSAGE);}

        ap=ap+price;//总价
        allPrice.setText("商品总价: \n"+ap);
    }
    catch (SQLException g)
    { System.out.println(g);
    }
}
if(p.getSource()==ok)
{
    try
    {
        if (n3!=0)
        {
            con=DriverManager.getConnection("jdbc:odbc:q","","");
            sql=con.createStatement();
            sql.executeUpdate("insert into information(listnum,allprice) values("+n2+", "+ap+"");
            rs=sql.executeQuery("SELECT * FROM information where listnum="+n2);
            con.close();
            ap=0;
            num=null;name=null;price=0;time=null;add=null;n3=0;
            goodsShow.setText(null);
            n2++;//帐单号
        }
    }
    else

```

```

        JOptionPane.showMessageDialog(this," 请 购 买 商 品 "," 提 示
",JOptionPane.WARNING_MESSAGE);}
    }
    catch (SQLException g)
    {System.out.println(g);
    }
}
if(p.getSource()==print)
{
    try
    {
        if (n3!=0)
        {
            con=DriverManager.getConnection("jdbc:odbc:q","","");
            sql=con.createStatement();
            sql.executeUpdate("insert into information(listnum,allprice) values("+n2+"",""+ap+"");
            rs=sql.executeQuery("SELECT * FROM information where listnum="+n2);
            con.close();
            ap=0;
            num=null;name=null;price=0;time=null;add=null;n3=0;
            goodsShow.setText(null);
            n2++;//帐单号
        }
        else
            JOptionPane.showMessageDialog(this," 请 购 买 商 品 "," 提 示
",JOptionPane.WARNING_MESSAGE);}
    }
    catch (SQLException g)
    {System.out.println(g);
    }
}
if(p.getSource()==clean)
{ try
    { if (n3!=0)
        { int a1=0;
            con=DriverManager.getConnection("jdbc:odbc:q","","");
            sql=con.createStatement();
            rs=sql.executeQuery("SELECT MAX(编号) from information");
            while(rs.next()) {
                a1=rs.getInt(1);}
            for (int i=0;i<n3 ;i++ )
            { sql.executeUpdate("delete from information where 编号="+a1-i); }
            con.close();
            ap=0;
            num=null;name=null;price=0;time=null;add=null;n3=0;
            goodsShow.setText(null);} }
}

```

```

catch (SQLException g)
    {System.out.println(g);} }

```

(3) 搜索商品的代码实现

```

if (p.getSource()==searchGoods)

```

```

    {
        num=numSearch.getText();
        try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        }
        catch (ClassNotFoundException f)
        {System.out.println(""+f);}
        try {
            con=DriverManager.getConnection("jdbc:odbc:q","","");
            sql=con.createStatement();
            rs=sql.executeQuery("SELECT * FROM goods where num='"+num+"'");
            while(rs.next()){
                name=rs.getString(2);
                price=rs.getInt(3);
                time=rs.getString(4);
                add=rs.getString(5);
                n1=1;}
            con.close();}
        catch (SQLException g)
            {System.out.println(g); }

```

```

if (n1==1)

```

```

    {
        showGoods.setText(" "+num+" "+name+" "+price+" "+time+" "+add);
        n1=0;    }

```

```

else

```

```

    {
        showGoods.setText(" ");
        num=null;name=null;price=0;time=null;add=null;
        JOptionPane.showMessageDialog(this," 输入错误\n 请重新输入","提示",JOptionPane.WARNING_MESSAGE);
    }

```

(4) 提交商品的代码

```

if(p.getSource()==ok)

```

```

    {
        try
        {
            if (n3!=0)
            { con=DriverManager.getConnection("jdbc:odbc:q","","");
            sql=con.createStatement();
            sql.executeUpdate("insert into information(listnum,allprice) values '"+n2+"','"+ap+"'");
            rs=sql.executeQuery("SELECT * FROM information where listnum='"+n2);
            con.close(); ap=0; num=null;name=null;price=0;time=null;add=null;n3=0;
            goodsShow.setText(null);

```



```

        n2++;//帐单号 }
    else
        {JOptionPane.showMessageDialog(this," 请 购 买 商 品 "," 提 示
",JOptionPane.WARNING_MESSAGE);}
    catch (SQLException g)
        {System.out.println(g);}

```

4.3 用户管理功能介绍

在用户管理部分，主要功能是控制小超市内用户的权限，不同的职位具有不同的权限，具有不同的 ID 和密码。在这部分当中，可以针对超市的情况对工作人员的信息实现添加用户、修改用户和删除用户的功能，并且超市内所有人员的信息都可以通过显示用户信息区域显示出来。界面的设计简单明了，将 `setLayout` 布局设为空，然后再向面板上添加组件，通过坐标值控制各组件的位置。权限管理界面如图 4-5 所示。

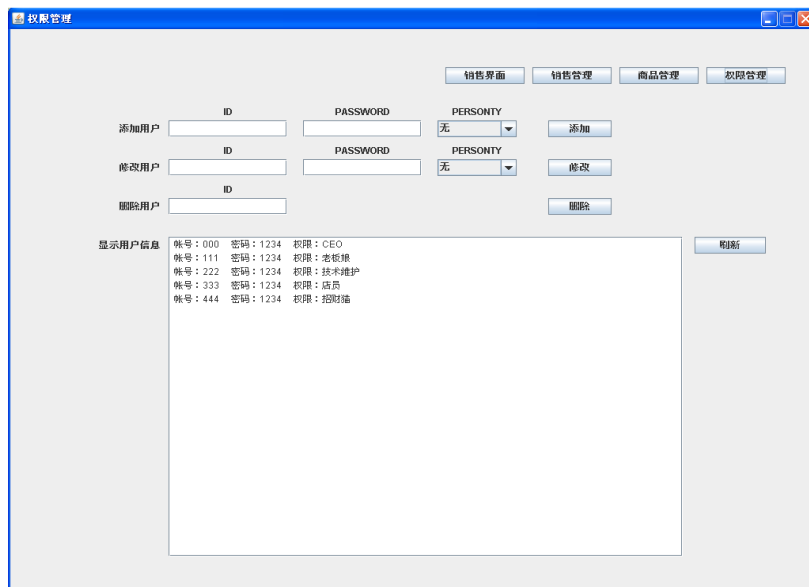
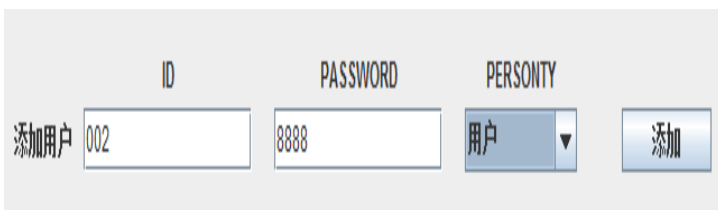


图 4-5 用户管理界面

4.3.1 添加用户功能介绍

在添加用户行所对应的文本框中输入要添加用户的 ID、PASSWORD、PERSONTY，单击添加按钮，“添加”按钮上已添加了监听器，在按钮被触发后，将会向“ID”、“PASSWORD”、“PERSONTY”所对应的文本框索取数据，只有当该用户在数据库中没有相应记录，并且添加的信息完整时才会显示添加成功，若所添加的信息不完整，则显示“请输入完整信息”，若信息完整，但该用户已存在，则显示“添加失败！原因：可能数据库添加失败或已有此 ID”。添加用户功能如图 4-6 所示。



(a)



(b)

图 4-6 添加用户

4.3.2 修改用户功能介绍

修改部分的操作方法与添加部分相类似，但是输入的是修改后的用户信息，“修改”按钮也添加了监控器，单击“修改”按钮后，会将新输入的“ID”、“PASSWORD”、“PERSONTY”信息返回到数据库中，替代原来的信息。同样的，输入的信息必须完整，修改的 ID 数据库中必须存在。修改用户功能如图 4.7 所示。

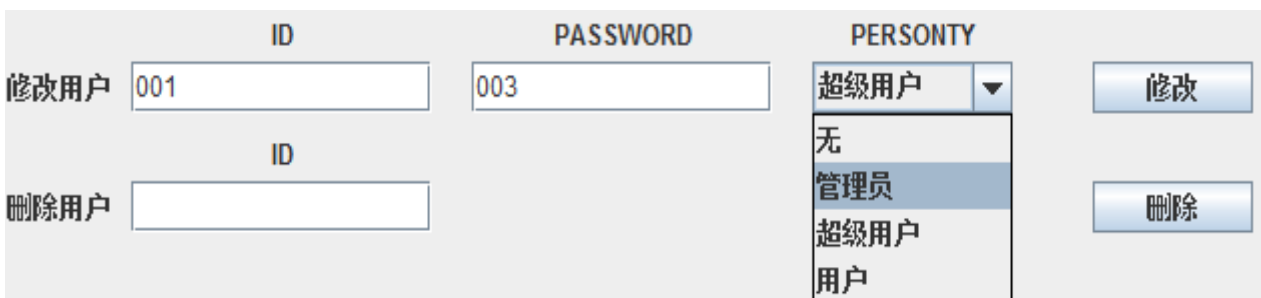
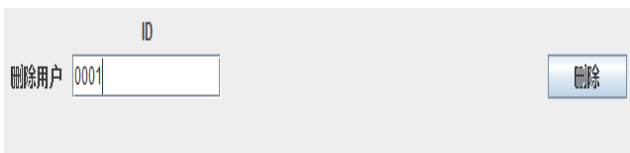


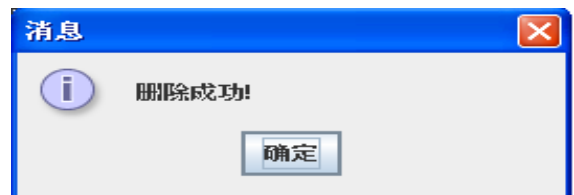
图 4.7 修改用户

4.3.3 删除用户功能介绍

删除用户部分的操作，只需要输入用户的 ID，然后点击“删除”按钮，由于删除按钮上也添加了监控器，当按钮被触发后，会从删除用户对应的文本框获取数据（即 ID 号码），再返回数据库，将该 ID 对应的信息删除。删除用户信息如图 4-8 所示。



(a)



(b)

图 4-8 删除用户

4.3.4 显示用户信息与刷新按钮介绍

设置显示用户信息区域的主要目的是将数据库中的用户信息显示出来，然而该区域只能显示在进入系统时的数据库信息，对于之后的修改等操作后的信息无法更新，所以添加了“刷新”按钮，使心事的信息更新，刷新的实质是将该界面关闭再重新打开，从而达到更新数据的目的。

4.3.5 用户管理的实现代码

用户管理的实现代码如下：

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
import javax.swing.*;
import javax.swing.event.*;

public class View3 extends JFrame implements ActionListener
{
    String url = "jdbc:odbc:driver={Microsoft Access Driver (*.mdb)};DBQ=D:\\supermarket.mdb";
    private JPanel v3=new JPanel();
    private JLabel idl1=new JLabel("ID",JLabel.CENTER),idl2=new JLabel("ID",JLabel.CENTER),idl3=new
    JLabel("ID",JLabel.CENTER),passwordl1=new JLabel("PASSWORD",JLabel.CENTER),passwordl2=new
    JLabel("PASSWORD",JLabel.CENTER),perl1=new JLabel("PERSONTY",JLabel.CENTER),perl2=new
    JLabel("PERSONTY",JLabel.CENTER);
    private JLabel[] cho={new JLabel("添加用户",JLabel.RIGHT),new JLabel("修改用户",JLabel.RIGHT),new
    JLabel("删除用户",JLabel.RIGHT),new JLabel("显示用户信息",JLabel.RIGHT)};
    private JTextField id1=new JTextField(),id2=new JTextField(),id3=new JTextField(),password1=new
    JTextField(),password2=new JTextField();
    private JButton[] pS={new JButton("销售界面"),new JButton("销售管理"),
        new JButton("商品管理"),new JButton("用户管理")};
    private JButton[] cB={new JButton("添加"),new JButton("修改"),new JButton("删除")};
    private JButton addin=new JButton("添加"),updatein=new JButton("修改"),dele=new JButton("修改");
    private String[] str={"无","管理员","超级用户","用户"};
    private JComboBox per1=new JComboBox(str),per2=new JComboBox(str);
    private JTextArea jtas=new JTextArea();
    private JScrollPane jsp=new JScrollPane(jtas);
    private JButton reShow=new JButton("刷新");
    Connection con;//连接对象
```

```
ResultSet rs;//结果集
```

```
Statement st;
```

```
View3()
```

```
{ v3.setLayout(null);
```

```
for(int i=0;i<4;i++)
```

```
{
```

```
pS[i].setBounds(550+i*110,50,100,22);
```

```
v3.add(pS[i]);
```

```
pS[i].addActionListener(this);
```

```
}
```

```
for (int i=0;i<4 ;i++ )
```

```
{ cho[i].setBounds(50,120+i*51,140,22);
```

```
v3.add(cho[i]);
```

```
}
```

```
for (int i=0;i<3 ;i++ )
```

```
{ cB[i].setBounds(680,120+i*51,80,22);
```

```
v3.add(cB[i]);
```

```
cB[i].addActionListener(this);
```

```
}
```

```
idl1.setBounds(200,98,150,22);password1.setBounds(370,98,150,22);per1.setBounds(540,98,100,22);
```

```
id1.setBounds(200,120,150,22);password1.setBounds(370,120,150,22);per1.setBounds(540,120,100,22);
```

```
idl2.setBounds(200,149,150,22);password2.setBounds(370,149,150,22);per2.setBounds(540,149,100,22);
```

```
id2.setBounds(200,171,150,22);password2.setBounds(370,171,150,22);per2.setBounds(540,171,100,22);
```

```
idl3.setBounds(200,200,150,22);
```

```
id3.setBounds(200,222,150,22);
```

```
reShow.setBounds(865,273,90,22);
```

```
reShow.addActionListener(this);
```

```
try
```

```
{
```

```
Connection con;
```

```
Statement sql;
```

```
ResultSet rs;
```

```
con=DriverManager.getConnection(url);
```

```
sql=con.createStatement();
```

```
rs=sql.executeQuery("SELECT * from password");
```

```
while(rs.next())
```

```
        {
            String a1=rs.getString(1);
            String a2=rs.getString(2);
            String a3=rs.getString(3);
            jtas.append(" 帐号: "+a1+"      密码: "+a2+"      权限: "+a3+"\n");
        }
                con.close();
    }

    catch (SQLException g)
    {System.out.println(g);
    }
jsp.setBounds(200,273,650,420);
v3.add(reShow);
v3.add(jsp);
v3.add(per1);
v3.add(per2);
v3.add(per11);
v3.add(per12);
v3.add(id1);
v3.add(id1);
v3.add(id2);
v3.add(id3);
v3.add(id3);
v3.add(password1);
v3.add(password1);
v3.add(password2);
v3.add(password2);
addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
    System.exit(0);
}
});
this.add(v3);
this.setTitle("权限管理");
this.setResizable(false);
this.setBounds(0,0,1024,768);
this.setVisible(true);
validate();
```

```
    }
//连接数据库
public void conDB() {
    String url = "jdbc:odbc:driver={Microsoft Access Driver (*.mdb)};DBQ=D:\\supermarket.mdb";
    try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

    } catch (ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, "数据库操作错误或失败!");
    }
    try {    con = DriverManager.getConnection(url);
    st = con.createStatement();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "数据库连接失败!");
    }
}
//关闭数据库
public void closeDB() {
    try {
        st.close();
        con.close();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "数据库关闭失败!");
    }
}
public void actionPerformed(ActionEvent e)
    {
    if(e.getSource()==reShow)
        {
            View3 fr=new View3();
            dispose();
            //提交添加用户操作
            if (e.getSource() == cB[0])
                {
                    if ((id1.getText().trim()).equals("") ||
                    (password1.getText().trim()).equals("") ||
                    (per1.getSelectedIndex()==0
                    ))
                        {
                            JOptionPane.showMessageDialog(null, "请输入完整信息! ");
                        }
                    }
                }
            }
```

```
} else {
String userid = id1.getText();
String username =password1.getText();
String userpwd =per1.getSelectedItem().toString();
System.out.println(userpwd+"11111111111111111111111111111111");

String strSQL =
"insert into password values('"+userid+" ','"+username+" ','"+ userpwd+"')";
try {
    conDB();
st.executeUpdate(strSQL);
closeDB();
} catch (Exception exx) {
    System.out.println(exx);
OptionPane.showMessageDialog(this, "添加失败! 原因: 有可能数据库联接失败或以由此ID, 尝试换个ID
添加");
return;
}
OptionPane.showMessageDialog(this, "添加成功!");
id1.setText("");
password1.setText("");
per1.setSelectedIndex(0);

}
}
//提交修改用户操作
if (e.getSource() == cB[1])
    {
if ((id2.getText().trim()).equals("") ||
(password2.getText().trim()).equals("") ||
(per2.getSelectedIndex()==0
))
    {
OptionPane.showMessageDialog(null, "请输入完整信息! ");
} else {
String userid = id2.getText();
String username =password2.getText();
String userpwd =per2.getSelectedItem().toString();
System.out.println(userpwd+"11111111111111111111111111111111");
```

```
String strSQL ="update password set personty=" + userpwd + ",password=" + username +"where ID=" + userid
+""";
try {
    conDB();
int a=st.executeUpdate(strSQL);
closeDB();
if (a==0)
{JOptionPane.showMessageDialog(this, "无此id信息， 修改失败!");
return;
}
} catch (Exception exx) {
    System.out.println(exx);
JOptionPane.showMessageDialog(this, "数据库原因， 修改失败!");
return;
}
JOptionPane.showMessageDialog(this, "修改成功!");
id2.setText("");
password2.setText("");
per2.setSelectedIndex(0);

}
}
//提交删除用户操作
if (e.getSource() == cB[2])
    {
if ((id3.getText().trim()).equals(""))
)
{
JOptionPane.showMessageDialog(null, "请输入完整信息! ");
} else {
String userid = id3.getText();

String strSQL ="delete from password where ID=" + userid +""";

try {
    conDB();
int a=st.executeUpdate(strSQL);
closeDB();
if (a==0)
{JOptionPane.showMessageDialog(this, "没有此用户      删除失败!");
```



```
return;
}
} catch (Exception exx) {
    System.out.println(exx);
OptionPane.showMessageDialog(this, "没有此用户    删除失败!");
return;
}
OptionPane.showMessageDialog(this, "删除成功!");
id3.setText("");
}}
if(e.getSource()==pS[0])
{View2 ve=new View2();
dispose();}
if(e.getSource()==pS[1])
{
    if(Login.persontyD.equals("管理员")||Login.persontyD.equals("超级用户")||Login.persontyD.equals("用户"))
    {
        sale2 frame1=new sale2();
        dispose();
    }
    else
        JOptionPane.showMessageDialog(null, "对不起,您没有进入权限!");
}
if(e.getSource()==pS[2])
{
    if(Login.persontyD.equals("管理员")||Login.persontyD.equals("超级用户"))
    {
        Frm_Main frame=new Frm_Main();
        dispose();
    }
    else
        JOptionPane.showMessageDialog(null, "对不起,您没有进入权限!");
}
}
```

4.4 销售管理功能介绍

销售管理功能主要是老板了解商品的销售情况，他可以登录进来，按已销售商品的编号和商品的账单号来查询商品的销售情况。也可以查询商品的售出的数量，当按编号查询时，后面的数量表示已售出商品的数量；当按账单号进行查询时，后面的数量表示消费者一次购买时，不同商品的总数。从而了解哪些商品销售的好，对商品的进货做出决定。

商品管理功能的界面如图 4-9 所示。

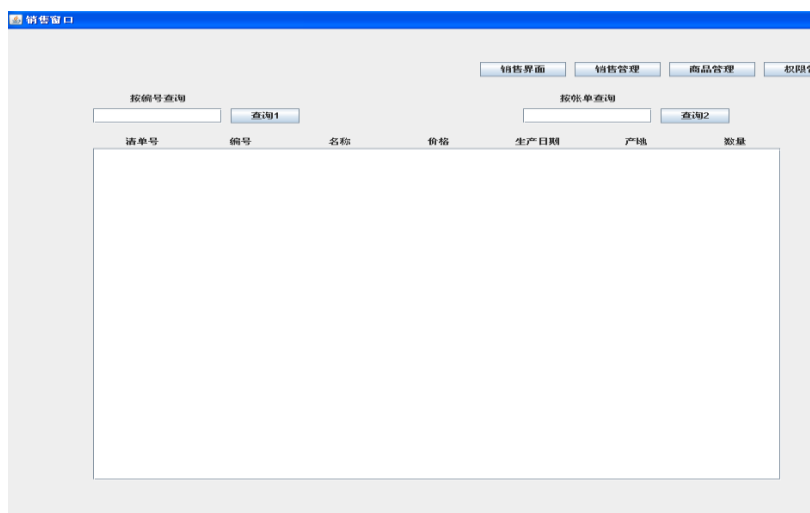


图 4-9 商品管理功能界面

4.4.1 编号查询功能

在已销售出的商品中，可以按照它的编号来进行查询，以显示售出商品的各种信息。其中最后的信息“数量”表示已售出商品的数量。如：查询 002 号商品的销售情况，该商品一共卖出 0 份。如图 4-10 所示。

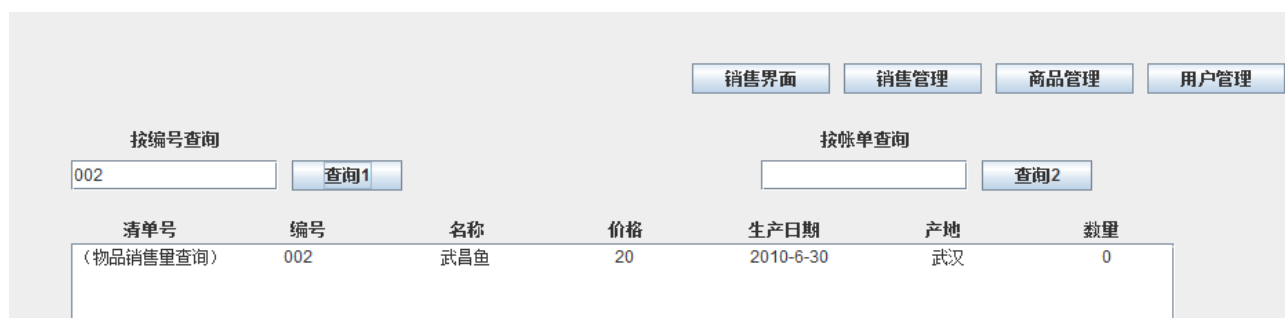


图 4-10 002 号商品的销售情况

4.4.2 账单查询功能

每一种商品在销售时，都有用户提交的一份账单，也就是流水号查询。消费者购买物品时，可以一次购买不同的商品，商品的编号不同，但是一次购买的账单号是一样的，账号查

询的功能就是查询消费者一次购买的商品。这个功能还可以防止消费者的结账出错时的查询，一次就可以查询出消费者的购买信息，解决错账的情况。

如查询流水号是 0 号的消费者的购买情况，在这一次的交易中销售如图 4-11 所示。

按编号查询				按帐单查询		
002		查询1		0		查询2
清单号	编号	名称	价格	生产日期	产地	数量
(物品销售量查询)	002	武昌鱼	20	2010-6-30	武汉	0
0	001	白菜	1	2010-7-1	沈阳	3
0	010	维达纸巾	26	2010-6-30	吉林	4
0	011	三笑牙刷	2	2010-6-30	大连	1

图 4-11 按账单查询情况

4.4.3 销售管理模块的核心代码

(1) 销售面板的实现

```
public sale2()
{
    label1=new JLabel("按编号查询",JLabel.CENTER);
    label2=new JLabel("按帐单查询",JLabel.CENTER);
    field1=new JTextField(30);field2=new JTextField(30);
    button1=new JButton("查询 1");button2=new JButton("查询 2");
    sM.setLayout(null);field1.setBounds(100,120,150,22);field2.setBounds(600,120,150,22);
    label1.setBounds(100,90,150,30); label2.setBounds(600,90,150,30);button1.setBounds(260,120,80,22);
    button2.setBounds(760,120,80,22); button1.addActionListener(this);button2.addActionListener(this);
    sM.add(label1);sM.add(label2);sM.add(field1);sM.add(field2);sM.add(button1);sM.add(button2);
    for(int i=0;i<4;i++)
    {
        pS[i].setBounds(550+i*110,50,100,22);
        sM.add(pS[i]);
        pS[i].addActionListener(this);
    }
    for(int i=0;i<7;i++) //6 个标签的绘制
    {
        CD[i].setBounds(100+i*115,160,115,20);
        sM.add(CD[i]);}
    jsp.setBounds(100,180,800,500);
    goodsShow.setLineWrap(true);
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
}
```

```

sM.add(jsp);
this.add(sM);
this.setTitle("销售窗口");
this.setResizable(false);
this.setBounds(0,0,1024,768);
this.setVisible(true); }

```

(2) 查询过程的处理

```

public void actionPerformed(ActionEvent p)
{
    Connection con,con1;
    Statement sql,sql1;
    ResultSet rs,rs1;
        if (p.getSource()==button1)
    {
        String num=field1.getText();
        try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); }
        catch (ClassNotFoundException f)
        {System.out.println(""+f);}
        try {
            con=DriverManager.getConnection("jdbc:odbc:q","", "");
            sql=con.createStatement();
            rs=sql.executeQuery("SELECT      count(goodsnum)      FROM      information      where
goodsnum='"+num+"'");
            while(rs.next())
            {
                countBG=rs.getInt(1);
                System.out.println(countBG);}
            rs=sql.executeQuery("SELECT * FROM goods where num='"+num+"'");
            while(rs.next())
            {name=rs.getString(2);
            price=rs.getInt(3);
            time=rs.getString(4);
            add=rs.getString(5);
            n1=1;}
            con.close();
            catch (SQLException g)
            {System.out.println(g);}
        if (n1==1)
            { goodsShow.append("      ( 物品销售量查询 )      "+num+"
"+name+"      "+price+"      "+time+"
"+add+"      "+countBG+"\n");
            n1=0; }
        else

```

```

        num=null;name=null;price=0;time=null;add=null;
        JOptionPane.showMessageDialog(this,"          输入错误\n          请
重新输入","提示",JOptionPane.WARNING_MESSAGE);
    }}
    if(p.getSource()==pS[0])
        { View2 frame=new View2();
          dispose();}
    if(p.getSource()==pS[2])
    {   Frm_Main frame=new Frm_Main();
        dispose();}
    if(p.getSource()==pS[3])
    {   View3 frame=new View3();
        dispose();}
    if (p.getSource()==button2)
    {   String numb=field2.getText();
        int num=Integer.parseInt(numb);
        try {   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); }
        catch (ClassNotFoundException f)
        {System.out.println(""+f);}
        try
        {con=DriverManager.getConnection("jdbc:odbc:q","","");
        sql=con.createStatement();
        rs=sql.executeQuery("SELECT   goodsnum,count(goodsnum)   FROM   information   where
listnum="+num+" group by goodsnum");
        while(rs.next())
        {   con1=DriverManager.getConnection("jdbc:odbc:q","","");
            sql1=con1.createStatement();
            String number=rs.getString(1);
            countBG=rs.getInt(2);
            System.out.println(countBG+number+name+time+price+add);
            rs1=sql1.executeQuery("SELECT * FROM goods where num="+number+"");
            while (rs1.next())
            {   name=rs1.getString(2);
                price=rs1.getInt(3);
                time=rs1.getString(4);
                add=rs1.getString(5);
                goodsShow.append("          "+num+"
"+number+"          "+name+"          "+price+"
"+time+"          "+add+"          "+countBG+"\n");
            }
        }
    }

```

```
        con1.close();
    }
    con.close();
}
catch (SQLException g)
{System.out.println(g);
}}}
```

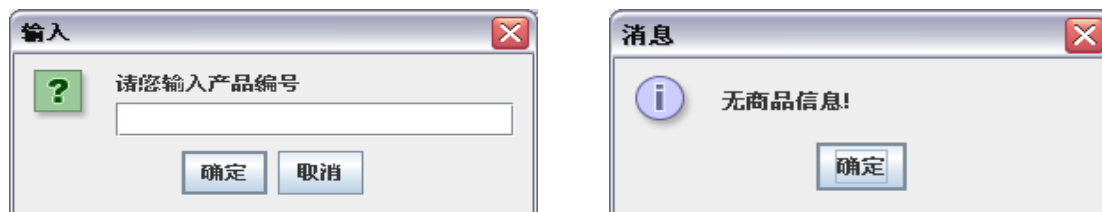
4.4 商品管理功能介绍

该模块主要实现对商品的信息管理功能，主要功能包括查询、添加、删除、修改商品信息，商品的主要信息包括商品编号、商品名称、商品价格、生产日期、生产地、库存量。商品管理界面如图 4-12 所示。

图 4-12 商品管理界面

4.4.1 查询信息

点击“查询信息”，出现如图 4-13(a)所示的对话框，填入所查询商品的编号，然后点击“确定”，若存在所查询的商品，将在左面的信息栏中显示商品信息；若不存在，则将出现如图 4-13(b)所示的对话框，提示无商品信息。



(a) 输入查询商品编号

(b) 无商品信息提示

图 4-13 查询的界面

4.4.2 添加信息及提交添加

点击“添加信息”后，会出现如图 4-14 所示的对话框，提示输入信息；当信息输入完毕后，点击提交添加，若输入正确，会出现如图 4-15(a)所示的对话框，提示添加成功，若输入不完全，有空项未添，则会出现如图 4-15(b)所示的对话框，提示添加错误。当添加成功后，信息栏会自动变为空。



图 4-14 提示输入信息



(a) 添加成功



(b) 添加错误

图 4-15 添加商品

4.4.3 删除信息及提交删除

点击“删除信息”后，会出现如图 4-16 所示的对话框，提示所要删除商品的名称，输入后点击“确定”，若输入的商品存在，在左面的信息栏中会显示所要删除的商品信息，再点击“提交删除”后，会出现如图 4-17(a)所示的对话框，提示删除成功；若输入的商品不存在，则会出现如图 4-17(b)所示的对话框，提示无商品信息。当删除成功后，信息栏会自动变为空。

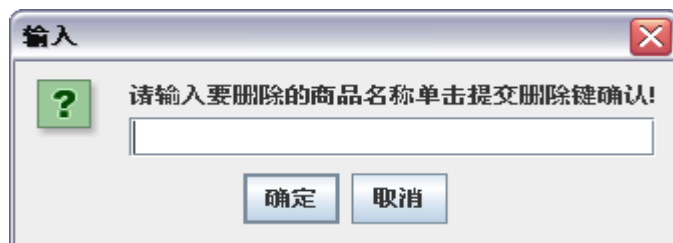


图 4-16 输入商品名称

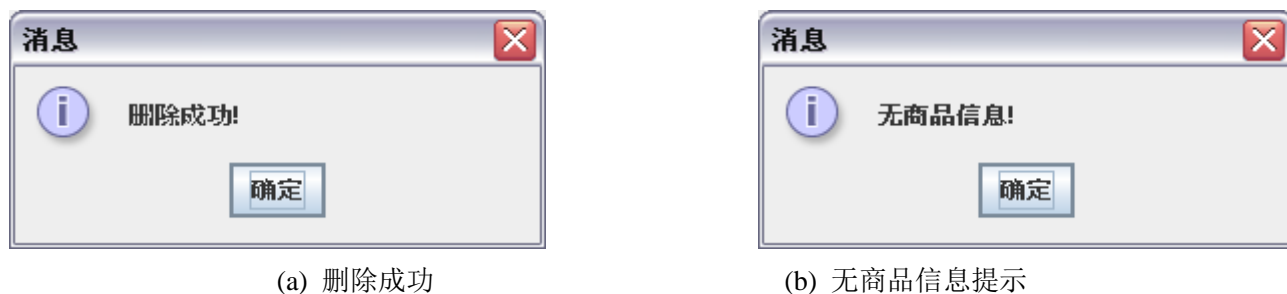


图 4.17 删除商品

4.4.4 修改信息及提交修改

点击“修改信息”后，会出现如图 4-18 所示的对话框，提示所要修改商品的名称，输入后点击“确定”，若输入的商品存在，在左面的信息栏中会显示所要修改的商品信息，修改完毕后，点击“提交修改”，会出现如图 4-19(a)所示的对话框，提示修改成功；若输入的商品不存在，则会出现如图 4-19(b)所示的对话框，提示无商品信息。修改成功后，信息栏会自动变为空。

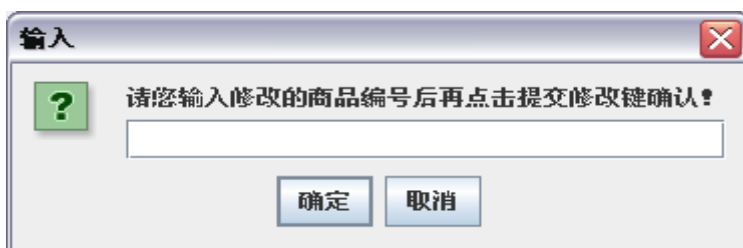


图 4-18 输入商品编号



图 4-19 修改商品

4.4.5 关于

当商品有问题或对超市有任何意见和建议时，可通过“关于”上的联系方式与主管部门取得联系。如图 4-20 所示。



图 4-20 联系电话提示

4.4.6 模块的核心代码

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
public class sale2 extends JFrame implements ActionListener //销售界面
{
    private JPanel sM=new JPanel(); //supermarket 面板
    JLabel label1,label2;
    JTextField field1,field2;
    JButton button1,button2;
    String num=null,name=null,time=null,add=null,t1=null,t2=null,t3=null;
    int price=0,countBG,n1;// 价格/商品的卖出数量
    private JTextArea goodsShow=new JTextArea(); //文本区
    private JScrollPane jsp=new JScrollPane(goodsShow); //滚动区域 设定内容为 商品显示的文本区

    private JButton[] pS={new JButton("销售界面"),
        new JButton("销售管理"),
        new JButton("商品管理"),
        new JButton("用户管理")}; //personty 按钮
    private JLabel[] CD={new JLabel("清单号",JLabel.CENTER),
        new JLabel("编号",JLabel.CENTER),
        new JLabel("名称",JLabel.CENTER),
        new JLabel("价格",JLabel.CENTER),
        new JLabel("生产日期",JLabel.CENTER),
        new JLabel("产地",JLabel.CENTER),
        new JLabel("数量",JLabel.CENTER)}; //帐单查询时的商品显示标签

    public sale2()
    {

        label1=new JLabel("按编号查询",JLabel.CENTER);
    }
}

```

```
label2=new JLabel("按帐单查询",JLabel.CENTER);

field1=new JTextField(30);
field2=new JTextField(30);

button1=new JButton("查询 1");
button2=new JButton("查询 2");
sM.setLayout(null);
field1.setBounds(100,120,150,22);
field2.setBounds(600,120,150,22);
label1.setBounds(100,90,150,30);
label2.setBounds(600,90,150,30);
button1.setBounds(260,120,80,22);
button2.setBounds(760,120,80,22);
button1.addActionListener(this);
button2.addActionListener(this);
sM.add(label1);
sM.add(label2);
sM.add(field1);
sM.add(field2);
sM.add(button1);
sM.add(button2);

for(int i=0;i<4;i++)
{
pS[i].setBounds(550+i*110,50,100,22);
sM.add(pS[i]);
pS[i].addActionListener(this);
}

for(int i=0;i<7;i++) //6 个标签的绘制
{
CD[i].setBounds(100+i*115,160,115,20);
sM.add(CD[i]);
}
jsp.setBounds(100,180,800,500);
goodsShow.setLineWrap(true);
addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
System.exit(0);
}
});
sM.add(jsp);
this.add(sM);
```

```
        this.setTitle("销售窗口");
        this.setResizable(false);
        this.setBounds(0,0,1024,768);
        this.setVisible(true);
    }
    public void actionPerformed(ActionEvent p)
    {
        Connection con,con1;
        Statement sql,sql1;
        String url = "jdbc:odbc:driver={Microsoft Access Driver (*.mdb)};DBQ=D:\\supermarket.mdb";
        ResultSet rs,rs1;
        if (p.getSource()==button1)
        {
            String num=field1.getText();
            try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            }
            catch (ClassNotFoundException f)
            {
                System.out.println(""+f);
            }
            try
            {
                con=DriverManager.getConnection(url);
                sql=con.createStatement();

                rs=sql.executeQuery("SELECT count(goodsnum) FROM information where
goodsnum='"+num+"'");

                while(rs.next())
                {
                    countBG=rs.getInt(1);
                    System.out.println(countBG);
                }
                rs=sql.executeQuery("SELECT * FROM goods where num='"+num+"'");
                while(rs.next())
                {
                    name=rs.getString(2);
                    price=rs.getInt(3);
                    time=rs.getString(4);
                    add=rs.getString(5);
                    n1=1;
                }

                con.close();
            }
            catch (SQLException g)
```

```

        {System.out.println(g);
        }
    if (n1==1)
        { goodsShow.append(" (物品销售量查询)           "+num+"
"+name+"           "+price+"           "+time+"
"+add+"           "+countBG+"\n");
        n1=0;
        }
    else
    {
        //goodsShow.setText(" ");
        num=null;name=null;price=0;time=null;add=null;
        JOptionPane.showMessageDialog(this,"           输入错误\n           请
重新输入","提示",JOptionPane.WARNING_MESSAGE);
    }
}

```

```

if(p.getSource()==pS[0])
{
    View2 frame=new View2();
    dispose();
}
if(p.getSource()==pS[2])
{
    Frm_Main frame=new Frm_Main();
    dispose();
}

```

```

if(p.getSource()==pS[3])
{
    View3 frame=new View3();
    dispose();
}

```

```

if (p.getSource()==button2)
{
    String numb=field2.getText();
    int num=Integer.parseInt(numb);

    try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    }
    catch (ClassNotFoundException f)
    {
        System.out.println(""+f);
    }
}

```

```

        try
        {
            con=DriverManager.getConnection(url);
            sql=con.createStatement();

            rs=sql.executeQuery("SELECT goodsnum,count(goodsnum) FROM information where
listnum="+num+" group by goodsnum");

            while(rs.next())
            {
                con1=DriverManager.getConnection(url);
                sql1=con1.createStatement();
                String number=rs.getString(1);
                countBG=rs.getInt(2);
                System.out.println(countBG+number+name+time+price+add);
                rs1=sql1.executeQuery("SELECT * FROM goods where num="+number+"");
                while (rs1.next())
                {
                    name=rs1.getString(2);
                    price=rs1.getint(3);
                    time=rs1.getString(4);
                    add=rs1.getString(5);
                    goodsShow.append("                "+num+"
"+number+"                "+name+"                "+price+"
"+time+"                "+add+"                "+countBG+"\n");
                }
                con1.close();
            }

            con.close();
        }
        catch (SQLException g)
        {System.out.println(g);
        }

        //{
        // goodsShow.setText(" ");
        // num=null;name=null;price=0;time=null;add=null;
        // JOptionPane.showMessageDialog(this,"                输入错误\n                请
重新输入","提示",JOptionPane.WARNING_MESSAGE);
        // }
    }
}
}
}

```

总 结

经过了两个星期的努力，在老师的大力支持和帮助下，我小组在规定时间内完成了此次课程设计的任务“小型超市管理系统”。

通过此次课程设计，使我们摆脱了单纯理论知识的学习状态，和实际的结合锻炼了我们解决实际设计中遇到问题的能力，也更加深了我们对所学的基础知识的掌握，深深地体会到了课程设计对理论知识促进与提高和对实践的具体应用。

选择设计题目很关键，小组人员共同研究讨论，最终确定了小型超市管理系统。之后在收集资料过程中，我们上网下载，并去了几家超市进行实地的考察，对超市的管理进行了需求分析，资料众多而且复杂，但我们克服了。

从最开始的不适应到将程序顺利的完成，是需要我们付出努力的，同时要有百折不挠的精神。这期间让我们懂得了许多在课堂上学不懂，学不透的知识，同时让我们感受到知识的重要性。开始时我们处于对很多相关知识不是很熟悉甚至不知道的状态，我们便开始研究和查看大量的相关资料和书籍，让自己头脑中模糊的概念逐渐清晰，使自己的作品一步步完善起来，每一次改进都是我们学习的收获。从中我们也充分认识到了面向对象 Java 程序设计给我们的学习带来了很大的帮助，也让我们学到了很多团结合作的经验，也锻炼了我们的整体思维的能力。有的时候把自己的新想法与他人分享，大家往往会提出置疑，但也就是在这种争论中才使我们的想法不断的提高、成熟。

通过对整体的设计，对局部的取舍，以及对细节的斟酌处理，都使我们的能力得到了锻炼，经验得到了丰富，并且意志品质力、抗压能力以及耐力也都得到了不同程度的提升，我想这就是我们课程设计要达到的目的。

经过长时间的艰辛努力最终换来了丰硕的成果，虽然有些不完善的地方，但毕竟是我们小组成员花费大量的时间设计出来的，也感到有一丝的欣慰。可以肯定地说现在的我们对程序设计这一块也有了一定的了解，但我们深知我们所了解的不过仅仅是皮毛而已，离真正的运用自如还相距甚远。

学无止境，学习生活无穷尽，只有掌握更多的、更新的知识才能使自己始终立于不败之地。如果只为现在所取得的成绩而沉浸于喜悦、骄傲之中的话，那么，我们就不会前进，只能停留在原来的地方踏步，甚至还会倒退。所以从毕业设计所暴露出的问题入手，不断努力完善自己，哪里欠缺，哪里补足。

致 谢

本次课程设计的选题、设计方法及报告的撰写均是在我们的指导教师的细心指导下进行的。设计中离不开老师细心的帮助，没有老师的细心指导，就没有我们的设计成果。老师在面向对象设计方面有很多的实践经验，使得我们在设计时遇到的困难都能够迎刃而解，在我们面对问题的时候对我们虚心的指导和帮助及其严谨的工作态度、创新的精神，都使我们受益匪浅，在此特别向老师表示深深的感谢和由衷的敬意。

在这次的设计中，我们小组遇到了一部分的困难，但是不放弃的精神使得我们并没有灰心，在经过大家的共同努力之后，我们的系统终于得到了完善。在此也要对帮助过我们的同学表示感谢，你们的帮助让我们的思路受到了启示，有了更大的发挥空间，让我们遇到困难的数量降到最低。

还要感谢的是我们小组内所有成员的团结协作的精神，正是因为有了团队意识，我们才能不畏困难，奋勇向前，进而圆满的完成了本次课程设计的任务。

另外，还要感谢学校给我们提供了此次课程设计的机会，不仅使我们所学的知识得以应用到实际中，也使我们开阔了视野，拓宽了思路，增长了学识，而且为我们今后的工作和学习打下了牢固的基础，也使增强我们对计算机的兴趣。

在此次课程设计中所学到的在以后的学习，工作，乃至生活上都将是我们宝贵的财富。我们以后会记住这两周的一点一滴的汗水和欢笑，会记住给予我们帮助的老师 and 同学们。最后，再一次衷心的感谢给予我们的帮助与支持的老师和同学们，谢谢你们！

参考文献

- [1] 耿祥义, 张跃平. Java2 实用教程(第三版). 北京: 清华大学出版社.. 2006.
- [2] 张小波, 曾锡山. Java 程序设计实训教程. 冶金工业出版社. 2006.
- [3] 孙一林, 彭波. Java 数据库编程实例. 北京: 清华大学出版社. 2003.
- [4] 严桂兰, 刘甲耀, 刘波. Java 编程模式与范例. 机械工业出版社. 2002.
- [5] 朱仲杰. Java 全方位学习. 机械工业出版社. 2006.
- [6] 张思民. Java 程序设计实践教程. 北京: 清华大学出版社. 2006.
- [7] 汤一平. Java 语言程序设计. 北京: 科学出版社. 2006.
- [8] 刘艺. Java 程序设计大学教程. 北京: 机械工业出版社. 2006.