组成

1.entity

1.Customers

```java
package entity;

public class Customers {
    private String customerNo;//会员编号
    private String customerName;//会员姓名
    private String phone;//会员电话
    private int score;//会员积分

    public Customers() {
        super();
    }

    public String getCustomerNo() {
        return customerNo;
    }
    public void setCustomerNo(String customerNo) {
        this.customerNo = customerNo;
    }
    public String getCustomerName() {
        return customerName;
    }
    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
    public int getScore() {
        return score;
    }
    public void setScore(int score) {
        this.score = score;
    }

}

```

2.ProductInfo

```java
package entity;

public class ProductInfo extends Products{
    private String categoryName;//商品种类名字

    public ProductInfo() {
        super();
    }

    public ProductInfo(String categoryName) {
        super();
        this.categoryName = categoryName;
    }

    public String getCategoryName() {
        return categoryName;
    }

    public void setCategoryName(String categoryName) {
        this.categoryName = categoryName;
    }

}

```

3.Products

```java
package entity;

public class Products {
    private String productNo;//商品编号
    private String categoryId;//种类编号
    private String productName;//商品名
    private String type;//规格
    private String unit;//单位
    private int pcount=-1;//库存数量
    private double price;//单价
    private double discount;//折扣
    public Products() {
        super();
    }

    public String getProductNo() {
        return productNo;
    }
    public void setProductNo(String productNo) {
        this.productNo = productNo;
    }
    public String getCategoryId() {
        return categoryId;
    }
    public void setCategoryId(String categoryId) {
        this.categoryId = categoryId;
    }
    public String getProductName() {
        return productName;
    }
    public void setProductName(String productName) {
        this.productName = productName;
    }
    public String getType() {
        return type;
    }
    public void setType(String type) {
        this.type = type;
    }
    public String getUnit() {
        return unit;
    }
    public void setUnit(String unit) {
        this.unit = unit;
    }
    public int getCount() {
        return pcount;
    }
    public void setCount(int pcount) {
        this.pcount = pcount;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public double getDiscount() {
        return discount;
    }
    public void setDiscount(double discount) {
        this.discount = discount;
    }

}

```

4.SaleHistory

```
package entity;

import javax.swing.JOptionPane;

public class SaleHistory {
    private int id;//记录编号
    private String productNo;//商品编号
    private int saleCount;//购买数量
    private double price;//商品单价
    private double discount;//商品折扣
    private int userId;//管理员编号
    private String saleTime;//售出时间
    private int customerNo;//消费者编号
    public SaleHistory() {
        super();
    }
    public SaleHistory(int id, String productNo, int saleCount, double price,
                       double discount, int userId, String saleTime, int customerNo) {
        super();
        this.id = id;
        this.productNo = productNo;
        this.saleCount = saleCount;
        this.price = price;
        this.discount = discount;
        this.userId = userId;
        this.saleTime = saleTime;
        this.customerNo = customerNo;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getProductNo() {
        return productNo;
    }
    public void setProductNo(String productNo) {
        this.productNo = productNo;
    }
    public int getSaleCount() {
        return saleCount;
    }
    public void setSaleCount(int saleCount) {
        this.saleCount = saleCount;
    }
    public double getPrice() {
        return price;
    }
    public int setPrice(double price) {
        int flag=0;
        if(price<0){
            flag=0;
        }
        else {
            this.price = price;
            flag=1;
        }
        return flag;
    }
    public double getDiscount() {
        return discount;
    }
    public void setDiscount(double discount) {
        this.discount = discount;
    }
    public int getUserId() {
        return userId;
    }
    public void setUserId(int userId) {
        this.userId = userId;}
    public String getSaleTime() {
        return saleTime;
    }
    public void setSaleTime(String saleTime) {
        this.saleTime = saleTime;
    }
    public int getCustomerNo() {
        return customerNo;
    }
    public void setCustomerNo(int customerNo) {
        this.customerNo = customerNo;
    }

}

```

5.SaleSummary

```
package entity;

public class SaleSummary extends SaleHistory{
    private double memberDiscount;//会员折扣
    private double totalAmount;//单行总金额
    public double getMemberDiscount() {
        return memberDiscount;
    }
    public void setMemberDiscount(double memberDiscount) {
        this.memberDiscount = memberDiscount;
    }
    public double getTotalAmount() {
        return totalAmount;
    }
    public void setTotalAmount(double totalAmount) {
        this.totalAmount = totalAmount;
    }
    public SaleSummary() {
        super();
    }

}

```

6.ScoreDiscount

```
package entity;

public class ScoreDiscount {
    private int minScore;//积分
    private double discount;//折扣
    public ScoreDiscount() {
        super();
    }
    public ScoreDiscount(int minScore, double discount) {
        super();
        this.minScore = minScore;
        this.discount = discount;
    }
    public int getMinScore() {
        return minScore;
    }
    public void setMinScore(int minScore) {
        this.minScore = minScore;
    }
    public double getDiscount() {
        return discount;
    }
    public void setDiscount(double discount) {
        this.discount = discount;
    }

}

```

7.Users

```
package entity;

public class Users {
    private int userId;//管理员编号
    private String userName;//管理员姓名
    private String password;//管理员密码
    private int roleId;//管理员级别
    public Users() {
        super();
    }
    public Users(int userId, String userName, String password, int roleId) {
        super();
        this.userId = userId;
        this.userName = userName;
        this.password = password;
        this.roleId = roleId;
    }
    public int getUserId() {
        return userId;
    }
    public void setUserId(int userId) {
        this.userId = userId;
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public int getRoleId() {
        return roleId;
    }
    public void setRoleId(int roleId) {
        this.roleId = roleId;
    }

}

```

2.dao

1.CategorieDao

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class CategorieDao {
    public String getCategoryNameDao(String categoryId){  //种类编号
```

```
        String categoryName = "";
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null; // PreparedStatement可以有效防止sql注入，
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from products where categoryid = ?");
            stat.setString(1, categoryId);
            rs=stat.executeQuery();
            rs.next();
            categoryName =rs.getString(2);
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return categoryName;
    }
}

```

2.CustomerDao

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JOptionPane;

import service.ScoreDiscountService;

import entity.Customers;
import entity.ScoreDiscount;
import entity.Users;

public class CustomerDao {
    public List<Customers> getAllCustomersDao(){
        List<Customers> list=new ArrayList<Customers>();
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from customers");
            rs=stat.executeQuery();
            while(rs.next()){
                Customers customer=new Customers();
                customer.setCustomerNo(rs.getString(1));
                customer.setCustomerName(rs.getString(2));
                customer.setPhone(rs.getString(3));
                customer.setScore(rs.getInt(4));
                list.add(customer);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return list;
    }
    public int newCustomersDao(Customers customer){
        boolean flag = false;
        String sql = "insert into customers (customerno,customername,phone,score) values(?,?,?,?)";
        Object[] objs = {customer.getCustomerNo(),customer.getCustomerName(),customer.getPhone(),0};
        flag = DBManager.executeUpdate(sql, objs);




        int customerNo = 0;
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select customerno from customers where customername= ? ");
            stat.setString(1,customer.getCustomerName());
            rs=stat.executeQuery();
            if(rs.next()){
                customerNo = rs.getInt(1);
```

```java
            }else{
                JOptionPane.showMessageDialog(null,"新增会员错误","错误提示",JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return customerNo;
    }
    public boolean deleteCustomersDao(String cstNo){
        boolean flag = false;
        String sql = "delete from customers where customerno =?";
        Object[] objs = {cstNo};
        flag = DBManager.executeUpdate(sql, objs);
        return flag;
    }
    public boolean checkCustomerDao(int customerNo){
        boolean flag = false;
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from customers where customerno = ?");
            stat.setInt(1, customerNo);
            rs=stat.executeQuery();
            if(rs.next()){
                flag = true;
            }else{
                JOptionPane.showMessageDialog(null,"该会员不存在","错误提示",JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return flag;
    }
    public boolean checkCustomerNoDao(String customerNo){
        boolean flag = false;
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from customers where customerno = ?");
            stat.setString(1, customerNo);
            rs=stat.executeQuery();
            if(rs.next()){
                JOptionPane.showMessageDialog(null,"该会员编号已经存在","错误提示",JOptionPane.ERROR_MESSAGE);
            }else{
                flag = true;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return flag;
    }
    public double getMemberDiscountByCustomerNoDao(int customerNo){
        double discount = 1;
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from customers where customerno = ?");
            stat.setInt(1, customerNo);
            rs=stat.executeQuery();
            if(rs.next()){
                int score = rs.getInt(4);
                ScoreDiscountService sds = new ScoreDiscountService();
                List<ScoreDiscount> list = sds.getAllScoreDiscountService();
                for (ScoreDiscount scoreDiscount : list) {
                    if(score>=scoreDiscount.getMinScore()){
                        discount = scoreDiscount.getDiscount();
                    }
                }
            }else{
```

```
        JOptionPane.showMessageDialog(null,"该会员不存在","错误提示",JOptionPane.ERROR_MESSAGE);
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }finally{
      DBManager.closeAll(conn, stat, rs);
    }
    return discount;
  }
  public boolean updateCustomerScoreDao(int customerNo,int score){
    boolean flag = false;
    String sql = "update customers set score=score+ ? where customerno = ?";
    Object[] objs = {score,customerNo};
    flag = DBManager.executeUpdate(sql, objs);
    return flag;
  }
}

```

3.DBManager

```
package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DBManager {
  static String driverName="com.microsoft.sqlserver.jdbc.SQLServerDriver";
  static String dbURL = "jdbc:sqlserver://127.0.0.1:1433;DatabaseName=heng_supermarket";
  static String userName="sa";
  static String userPwd="zhangheng0.0123";

  public static Connection getConnection(){
    Connection conn=null;
```

```java
        try {
            Class.forName(driverName);
            conn=DriverManager.getConnection(dbURL, userName,userPwd);
            System.out.println("succeed");
        } catch (Exception e) {
            System.out.println("failed");
            e.printStackTrace();
        }
        return conn;
    }
    public static boolean executeUpdate(String sql,Object[] objs){
        boolean flag=false;
        Connection conn=getConnection();
        PreparedStatement pst=null;
        try {
            pst=conn.prepareStatement(sql);
            for(int i=0;i<objs.length;i++){
                pst.setObject(i+1, objs[i]);
            }
            int num=pst.executeUpdate();
            if(num>0){
                flag=true;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            closeAll(conn, pst, null);
        }

        return flag;
    }
    public static void closeAll(Connection conn,Statement stat,ResultSet rs){
        try {
            if(rs!=null){
                rs.close();
            }
            if(stat!=null){
                stat.close();
            }
            if(conn!=null){
                conn.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

4.ProductsDao

```
package dao;

import java.awt.Component;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JOptionPane;

import service.CategorieService;
import UI.GoodsAdministrator;

import entity.Customers;
import entity.ProductInfo;
import entity.Products;

public class ProductsDao {
    public List<ProductInfo> getAllProductsDao(){
        List<ProductInfo> list=new ArrayList<ProductInfo>();
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from products");
            rs=stat.executeQuery();
            while(rs.next()){
                ProductInfo product = new ProductInfo();
                product.setProductNo(rs.getString(1));
                product.setCategoryId(rs.getString(2));
                product.setProductName(rs.getString(3));
                product.setType(rs.getString(4) );
                product.setUnit(rs.getString(5));
                product.setCount(rs.getInt(6));
                product.setPrice(rs.getDouble(7));
                product.setDiscount(rs.getDouble(8));
                CategorieService cs = new CategorieService();
                product.setCategoryName(cs.getCategoryNameService(product.getCategoryId()));
                list.add(product);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return list;
    }
    public ProductInfo getProductInfoByProductNo(String ProductNo){
        ProductInfo product = new ProductInfo();
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from products where productno = ?");
            stat.setString(1, ProductNo);
            rs=stat.executeQuery();
            if(rs.next()){
                product.setProductNo(rs.getString(1));
                product.setCategoryId(rs.getString(2));
                product.setProductName(rs.getString(3));
                product.setType(rs.getString(4) );
                product.setUnit(rs.getString(5));
                product.setCount(rs.getInt(6));
                product.setPrice(rs.getDouble(7));
                product.setDiscount(rs.getDouble(8));
                CategorieService cs = new CategorieService();
                product.setCategoryName(cs.getCategoryNameService(product.getCategoryId()));
            }else{
                JOptionPane.showMessageDialog(null,"查询的商品不存在","错误提示",JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return product;
    }
    public ProductInfo getProductInfoByproductName(String productName){
        ProductInfo product = new ProductInfo();
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from products where `name` =?");
            stat.setString(1, productName);
            rs=stat.executeQuery();
            if(rs.next()){
                product.setProductNo(rs.getString(1));
                product.setCategoryId(rs.getString(2));
                product.setProductName(rs.getString(3));
                product.setType(rs.getString(4) );
                product.setUnit(rs.getString(5));
                product.setCount(rs.getInt(6));
                product.setPrice(rs.getDouble(7));
                product.setDiscount(rs.getDouble(8));
                CategorieService cs = new CategorieService();
                product.setCategoryName(cs.getCategoryNameService(product.getCategoryId()));
            }else{
                JOptionPane.showMessageDialog(null,"查询的商品不存在","错误提示",JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return product;
    }
    public boolean newProductsDao(Products product){
        boolean flag = false;
```

```java
        String sql = "insert into products values(?,?,?,?,?,?,?,?)";
        Object[] objs = {product.getProductNo(),product.getCategoryId(),product.getProductName(),product.getType(),product.getUnit()
                ,product.getCount(),product.getPrice(),product.getDiscount()};
        flag = DBManager.executeUpdate(sql, objs);
        return flag;
    }
    public boolean deleteProductsDao(String productNo){
        boolean flag = false;
        String sql = "delete from products where productno =?";
        Object[] objs = {productNo};
        flag = DBManager.executeUpdate(sql, objs);
        return flag;
    }
    public boolean updateProductsDao(String productNo,int pcount,double discount){
        boolean flag = false;
        String sql = "update products set pcount = ?,`discount` = ? where productno = ?";
        Object[] objs = {pcount,discount,productNo};
        flag = DBManager.executeUpdate(sql, objs);
        return flag;
    }
    public boolean updateProductsCountDao(String productNo,int soldNumber){
        boolean flag = false;
        int pcount = 0;
        ProductsDao pd = new ProductsDao();
        pcount = pd.getProductInfoByProductNo(productNo).getCount()-soldNumber;
        String sql = "update products set pcount = ? where productno = ?";
        Object[] objs = {pcount,productNo};
        flag = DBManager.executeUpdate(sql, objs);
        return flag;
    }
    public boolean checkProductsNoDao(String ProductsNo){
        boolean flag = false;
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from products where productno = ?");
            stat.setString(1, ProductsNo);
            rs=stat.executeQuery();
            if(rs.next()){
                JOptionPane.showMessageDialog(null,"该商品编号已经存在","错误提示",JOptionPane.ERROR_MESSAGE);
            }else{
                flag = true;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return flag;
    }
}

```

5.SaleHistoryDao

```
package dao;

import entity.SaleHistory;
/*
 *
 *
 * */
public class SaleHistoryDao {
    public boolean newSaleHistoryDao(SaleHistory saleHistory){
        boolean flag = false;
        if (saleHistory.getCustomerNo()==0) {
            System.out.println("这个人的信息没必要插入");
            return flag;
        }
        else{
            String sql = "insert into salehistory(productno,salecount,price,discount,userID,saletime,customerno) values(?,?,?,?,?,?,?)";
            Object[] objs = {saleHistory.getProductNo(),saleHistory.getSaleCount(),saleHistory.getPrice(),saleHistory.getDiscount()
            ,saleHistory.getUserId(),saleHistory.getSaleTime(),saleHistory.getCustomerNo()};
            flag = DBManager.executeUpdate(sql, objs);
            System.out.println("插入在当前收银台的这个会员的消费信息");
            return flag;
        }
    }
}

```

6.ScoreDiscountDao

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import entity.ScoreDiscount;

public class ScoreDiscountDao {
    public List<ScoreDiscount> getAllScoreDiscountDao(){
        List<ScoreDiscount> list=new ArrayList<ScoreDiscount>();
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select distinct c.score,p.discount from customers c, products p");
            rs=stat.executeQuery();
            while(rs.next()){
                ScoreDiscount sc = new ScoreDiscount();
                sc.setMinScore(rs.getInt(1));
                sc.setDiscount(rs.getFloat(2));
                list.add(sc);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return list;
    }
}
```

7.UsersDao

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JOptionPane;

import service.CategorieService;

import entity.ProductInfo;
import entity.Users;
/*
 * UsersDao类
 * 连接数据库获取返回的对象
 * 将各个数据赋值给变量
 * */
public class UsersDao {
    public List<Users> getAllUsersDao(){
        List<Users> list=new ArrayList<Users>();//n个用户的数据
        Connection conn=DBManager.getConnection();//连接数据库
        PreparedStatement stat=null;//PreparedStatement 接口继承了Statement
        ResultSet rs=null;//返回的对象
        try {
            stat=conn.prepareStatement("select * from users");
            rs=stat.executeQuery();//用statement类的executeQuery()方法来下达select指令以查询数据库，
```

```java
            while(rs.next()){
                Users user=new Users();
                user.setUserId(rs.getInt(1));//返回的第一个数据
                user.setUserName(rs.getString(2));//返回的第二个数据
                user.setPassword(rs.getString(3));//返回的第三个数据
                user.setRoleId(rs.getInt(4));
                list.add(user);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return list;
    }
    /*
     * checkUsersDao判断用户信息是否正确
     * 判断name password 角色是否对应
     * */
    public boolean checkUsersDao(String UserName,String Pwd,int roleId){
        boolean flag = false;
        if(UserName.isEmpty()){
            return flag;
        }else if(Pwd.isEmpty()){
            return flag;
        }else{
            UsersDao user = new UsersDao();
            List<Users> l =user.getAllUsersDao();
            boolean tag = false;
            for (Users a : l) {
                if(UserName.equals(a.getUserName()) && Pwd.equals(a.getPassword()) && roleId == a.getRoleId()){
                    tag = true;
                    flag = true;
                }
            }
            if(!tag){
                return flag;
            }
        }
        return flag;
    }
    /*
     * 新建用户
     *
     * */


    public boolean newUsersDao(Users user){
        boolean flag = false;
        String sql = "insert into users values(?,?,?,?)";
        Object[] objs = {user.getUserId(),user.getUserName(),user.getPassword(),user.getRoleId()};
        flag = DBManager.executeUpdate(sql, objs);
        return flag;
    }
    public boolean deleteUsersDao(int userId){
        boolean flag = false;
        String sql = "delete from users where userId = ?";
        Object[] objs = {userId};
        flag = DBManager.executeUpdate(sql, objs);
        return flag;
    }
    public int getUserIdByUserName(String userName){
        int userId = 1000;
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select userId from users where username = ?");
            stat.setString(1, userName);
            rs=stat.executeQuery();
            if(rs.next()){
                userId = rs.getInt(1);
            }else{
                JOptionPane.showMessageDialog(null,"查询的管理员信息不存在","错误提示",JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return userId;
    }
    public Users getUsersByUserId(int userId){
        Users user = new Users();
        Connection conn=DBManager.getConnection();
        PreparedStatement stat=null;
        ResultSet rs=null;
        try {
            stat=conn.prepareStatement("select * from users where userId =?");
            stat.setInt(1, userId);
            rs=stat.executeQuery();
            if(rs.next()){
                user.setUserId(rs.getInt(1));
                user.setUserName(rs.getString(2));
                user.setPassword(rs.getString(3));
                user.setRoleId(rs.getInt(4));
            }else{
                JOptionPane.showMessageDialog(null,"查询的管理员信息不存在","错误提示",JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            DBManager.closeAll(conn, stat, rs);
        }
        return user;
    }
    public boolean updateUserPasswordDao(int userId,String password){
        boolean flag = false;
        String sql = "update users set password = ? where userId = ?";
        Object[] objs = {password,userId};
        flag = DBManager.executeUpdate(sql, objs);
        return flag;
    }
}

```

3.service

1.CategorieService

```java
package service;

import dao.CategorieDao;

public class CategorieService {
    public String getCategoryNameService(String categoryId){
        CategorieDao cd = new CategorieDao();
        return cd.getCategoryNameDao(categoryId);
    }
}

```

2.CustomerService

```java
package service;

import java.util.List;

import dao.CustomerDao;

import entity.Customers;

public class CustomerService {
    public List<Customers> getAllCustomersService(){
        CustomerDao cd = new CustomerDao();
        return cd.getAllCustomersDao();
    }
    public int newCustomersService(Customers customer){
        CustomerDao cd = new CustomerDao();
        return cd.newCustomersDao(customer);
    }
    public boolean deleteCustomersService(String cstNo){
        CustomerDao cd = new CustomerDao();
        return cd.deleteCustomersDao(cstNo);
    }
    public boolean checkCustomerDao(int customerNo){
        CustomerDao cd = new CustomerDao();
        return cd.checkCustomerDao(customerNo);
    }
    public double getMemberDiscountByCustomerNoService(int customerNo){
        CustomerDao cd = new CustomerDao();
        return cd.getMemberDiscountByCustomerNoDao(customerNo);
    }
    public boolean updateCustomerScoreDao(int customerNo,int score){
        CustomerDao cd = new CustomerDao();
        return cd.updateCustomerScoreDao(customerNo, score);
    }
    public boolean checkCustomerNoDao(String customerNo){
        CustomerDao cd = new CustomerDao();
        return cd.checkCustomerNoDao(customerNo);
    }
}

```

3.ProductsService

```
package service;

import java.util.List;

import dao.ProductsDao;

import entity.ProductInfo;
import entity.Products;

public class ProductsService {
    public List<ProductInfo> getAllProductsService(){
        ProductsDao pd = new ProductsDao();
        return pd.getAllProductsDao();
    }
    public ProductInfo getProductInfoByProductNoService(String ProductNo){
        ProductsDao pd = new ProductsDao();
        return pd.getProductInfoByProductNo(ProductNo);
    }

    public ProductInfo getProductInfoByproductNameService(String productName){
        ProductsDao pd = new ProductsDao();
        return pd.getProductInfoByproductName(productName);
    }
    public boolean newProductsService(Products product){
        ProductsDao pd = new ProductsDao();
        return pd.newProductsDao(product);
    }
    public boolean deleteProductsService(String productNo){
        ProductsDao pd = new ProductsDao();
        return pd.deleteProductsDao(productNo);
    }
    public boolean updateProductsService(String productNo,int count,double discount){
        ProductsDao pd = new ProductsDao();
        return pd.updateProductsDao(productNo, count, discount);
    }
    public boolean updateProductsCountService(String productNo,int soldNumber){
        ProductsDao pd = new ProductsDao();
        return pd.updateProductsCountDao(productNo, soldNumber);
    }
    public boolean checkProductsNoDao(String ProductsNo){
        ProductsDao pd = new ProductsDao();
        return pd.checkProductsNoDao(ProductsNo);
    }
}

```

4.SaleHistoryService

```
package service;

import dao.SaleHistoryDao;
import entity.SaleHistory;

public class SaleHistoryService {
    public boolean newSaleHistoryService(SaleHistory saleHistory){
        SaleHistoryDao shd = new SaleHistoryDao();
        return shd.newSaleHistoryDao(saleHistory);
    }
}

```

5.ScoreDiscountService

```java
package service;

import java.util.List;

import dao.ScoreDiscountDao;

import entity.ScoreDiscount;

public class ScoreDiscountService {
    public List<ScoreDiscount> getAllScoreDiscountService(){
        ScoreDiscountDao sdd = new ScoreDiscountDao();
        return sdd.getAllScoreDiscountDao();
    }
}

```

6.UsersService

```java
package service;

import java.util.List;

import dao.UsersDao;

import entity.Users;

public class UsersService {
    public List<Users> getAllUsersService(){
        UsersDao ud = new UsersDao();
        return ud.getAllUsersDao();
    }
    public boolean checkUsersService(String UserName,String Pwd,int roleId){
        UsersDao ud = new UsersDao();
        return ud.checkUsersDao(UserName, Pwd, roleId);
    }
    public boolean deleteUsersService(int userId){
        UsersDao ud = new UsersDao();
        return ud.deleteUsersDao(userId);
    }
    public boolean newUsersService(Users user){
        UsersDao ud = new UsersDao();
        return ud.newUsersDao(user);
    }
    public int getUserIdByUserNameService(String userName){
        UsersDao ud = new UsersDao();
        return ud.getUserIdByUserName(userName);
    }
    public Users getUsersByUserIdService(int userId){
        UsersDao ud = new UsersDao();
        return ud.getUsersByUserId(userId);
    }
    public boolean updateUserPasswordService(int userId,String password){
        UsersDao ud = new UsersDao();
        return ud.updateUserPasswordDao(userId, password);
    }
}

```

4.UI

1.AdministratorManagement

```java
package UI;

import java.util.List;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JButton;
import javax.swing.JTable;

import entity.Users;

import service.UsersService;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class AdministratorManagement extends JFrame {

    private JPanel contentPane;
    private JTable table;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        AdministratorManagement frame = new AdministratorManagement();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public AdministratorManagement() {
        this.setTitle("管理员信息管理");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(500, 200, 680, 390);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(40, 40, 40, 40));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(50, 85, 520, 192);
        contentPane.add(scrollPane);

        String[] cols={"员工编号","用户名","密码","管理员级别"};//建立一个数组
        final DefaultTableModel model = new DefaultTableModel(cols,0);   //建立一个有cols数组长度的表
        table = new JTable(model);
        UsersService us = new UsersService();
        List<Users> rows = us.getAllUsersService();
        for (Users users : rows) {
            Object[] row = {users.getUserId(),users.getUserName(),users.getPassword(),users.getRoleId()};
            model.addRow(row);
        }
        scrollPane.setViewportView(table);

        JButton btnNewButton = new JButton("新增");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                AdministratorManagement.this.dispose();
                NewAdministrator na = new NewAdministrator();
                na.setVisible(true);
            }
        });
        btnNewButton.setBounds(307, 23, 103, 33);
        contentPane.add(btnNewButton);

        JButton button_1 = new JButton("删除");
        button_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int rowIndex = table.getSelectedRow();
                if(rowIndex<0){
                    JOptionPane.showMessageDialog(AdministratorManagement.this, "请选择要删除的行","错误提示",JOptionPane.ERROR_MESSAGE);
                }else{
                    int userId = (Integer) model.getValueAt(rowIndex, 0);
                    if(JOptionPane.showConfirmDialog(AdministratorManagement.this, "确定删除第"+(rowIndex+1)+"行?")==0){
                        UsersService us = new UsersService();
                        if(us.deleteUsersService(userId)){
                            JOptionPane.showMessageDialog(AdministratorManagement.this, "删除成功!");
                            AdministratorManagement.this.dispose();
                            AdministratorManagement frame = new AdministratorManagement();
                            frame.setVisible(true);
                        }
                    }
                }
            }
        });
        button_1.setBounds(451, 23, 103, 33);
        contentPane.add(button_1);

        JButton btnNewButton_1 = new JButton("返回主管理员界面");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                AdministratorManagement.this.dispose();
                RootAdministrator ra = new RootAdministrator();
                ra.setVisible(true);
            }
        });

        btnNewButton_1.setBounds(469, 297, 155, 43);
        contentPane.add(btnNewButton_1);
    }
}

```

2.Cashier

```
package UI;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JLabel;

import service.CustomerService;
import service.UsersService;

import entity.Users;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;

public class Cashier extends JFrame {

    private JPanel contentPane;

```

```java
    private JTextField userName;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {

        Cashier frame = new Cashier(1003);
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public Cashier(final int userId) {
        this.setTitle("收银员界面");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 523, 370);

        JMenuBar menuBar = new JMenuBar();//菜单栏
        setJMenuBar(menuBar);

        JMenu mnNewMenu = new JMenu("收银员信息管理");//构造一个新 JMenu，用提供的字符串作为其文本。
//      mnNewMenu.setMnemonic('M');//键盘助记符
        mnNewMenu.setHorizontalAlignment(SwingConstants.LEFT);//设置对齐方式
        menuBar.add(mnNewMenu);

        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel cashierId = new JLabel("New label");
        cashierId.setText("当前操作收银员编号:"+userId);
        cashierId.setBounds(55, 267, 155, 15);
        contentPane.add(cashierId);


        final JButton button = new JButton("会员收银");
        button.setFont(new Font("微软雅黑 Light", Font.PLAIN,
21));
        button.setBounds(72, 91, 150, 88);
        contentPane.add(button);

        final JButton button_1 = new JButton("非会员收银");
        button_1.setFont(new Font("微软雅黑 Light",
Font.PLAIN, 21));
        button_1.setBounds(264, 91, 155, 88);
        contentPane.add(button_1);


        JMenuItem jmi1 = new JMenuItem("修改密码");
//      jmi1.setMnemonic('C');
        jmi1.setHorizontalAlignment(SwingConstants.CENTER);
        jmi1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                button.setVisible(false);
                button_1.setVisible(false);
                UsersService us = new UsersService();
                final Users changeUser =
 us.getUsersByUserIdService(userId);
                //显示修改密码界面
                {
                    JLabel labelun = new JLabel("用户名:");
                    labelun.setBounds(88, 40, 54, 15);
                    contentPane.add(labelun);

                    userName = new JTextField();

 userName.setText(changeUser.getUserName());
                    userName.setEditable(false);
                    userName.setBounds(206, 37, 134, 21);
                    contentPane.add(userName);
                    userName.setColumns(10);

                    JLabel label = new JLabel("请输入原密码:");
                    label.setBounds(88, 81, 94, 15);
                    contentPane.add(label);

                    textField = new JTextField();
                    textField.setColumns(10);
                    textField.setBounds(206, 78, 134, 21);
                    contentPane.add(textField);

                    JLabel label_1 = new JLabel("请输入新密码:");
                    label_1.setBounds(88, 122, 94, 15);
                    contentPane.add(label_1);

                    textField_1 = new JTextField();
                    textField_1.setColumns(10);
                    textField_1.setBounds(206, 119, 134, 21);
                    contentPane.add(textField_1);

                    JLabel label_2 = new JLabel("请确认新密码:");
                    label_2.setBounds(88, 169, 94, 15);
                    contentPane.add(label_2);

                    textField_2 = new JTextField();
                    textField_2.setColumns(10);
                    textField_2.setBounds(206, 166, 134, 21);
                    contentPane.add(textField_2);

                    JButton btnNewButton = new JButton("确认修改");
                    btnNewButton.addActionListener(new ActionListener() {
                        @Override
                        public void actionPerformed(ActionEvent e) {
                            String psd = textField.getText();
                            String newPsd = textField_1.getText();
                            String newPsdc = textField_2.getText();
                            if(changeUser.getPassword().equals(psd)){
                                if(newPsd.equals(newPsdc)){
                                    UsersService us1 = new UsersService();
                                    if(us1.updateUserPasswordService(userId, newPsdc)){
                                        JOptionPane.showMessageDialog(Cashier.this, "修改成功!");
                                        Cashier.this.dispose();
                                        Login frame = new Login();
                                        frame.setVisible(true);
                                    }else{
                                        JOptionPane.showMessageDialog(Cashier.this, "修改失败!");
                                    }
                                }else{
                                    JOptionPane.showMessageDialog(Cashier.this, "两次输入不一致!","错误提示",JOptionPane.ERROR_MESSAGE);
                                }
                            }else{
                                JOptionPane.showMessageDialog(Cashier.this, "原始密码错误!","错误提示",JOptionPane.ERROR_MESSAGE);
                            }
                        }
                    });
                    btnNewButton.setBounds(117, 216, 93, 23);
                    contentPane.add(btnNewButton);

                    JButton btnNewButton_1 = new JButton("取消");
                    btnNewButton_1.addActionListener(new ActionListener() {
                        @Override
                        public void actionPerformed(ActionEvent e) {
                            Cashier.this.dispose();
                            Cashier frame = new Cashier(userId);
                            frame.setVisible(true);

                        }
                    });
                    btnNewButton_1.setBounds(235, 216, 93, 23);
                    contentPane.add(btnNewButton_1);
                    //刷新页面
                    Cashier.this.repaint();
                }


```

```

            }
        });
        mnNewMenu.add(jmi1);
        JMenuItem Cancellation= new JMenuItem("注销");
//      Cancellation.setMnemonic('W');
        Cancellation.setHorizontalAlignment(SwingConstants.CENTER);
        Cancellation.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                Cashier.this.dispose();
                Login l = new Login();
                l.setVisible(true);
            }
        });
        mnNewMenu.add(Cancellation);


        JMenuItem exit = new JMenuItem("退出");
//      exit.setMnemonic('R');
        exit.setHorizontalAlignment(SwingConstants.CENTER);
        exit.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                Cashier.this.dispose();
            }
        });
        mnNewMenu.add(exit);


        //会员收银
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                button.setVisible(false);
                button_1.setVisible(false);
                JLabel lblNewLabel = new JLabel("请输入会员卡卡号");
                lblNewLabel.setBounds(101, 86, 109, 15);
                contentPane.add(lblNewLabel);

                textField_3 = new JTextField();
                textField_3.setBounds(220, 83, 155, 21);
                contentPane.add(textField_3);
                textField_3.setColumns(10);

                JButton btnNewButton_2 = new JButton("收银");
                btnNewButton_2.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        String customerNo1 = textField_3.getText();
                        int customerNo = 0;
                        try {
                            customerNo = Integer.parseInt(customerNo1);
                        } catch (NumberFormatException e1) {
                            e1.printStackTrace();
                        }
                        CustomerService cs = new CustomerService();
                        if(cs.checkCustomerDao(customerNo)){
                            Cashier.this.dispose();
                            CollectMoney cm = new CollectMoney(userId, customerNo);
                            //System.out.println("cm"+userId);
                            cm.setVisible(true);
                        }
                    }
                });
                btnNewButton_2.setBounds(127, 190, 93, 23);
                contentPane.add(btnNewButton_2);

                JButton button = new JButton("返回主窗口");
                button.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        Cashier.this.dispose();
                        Cashier c = new Cashier(userId);
                        c.setVisible(true);
                    }
                });
                button.setBounds(263, 190, 120, 23);
                contentPane.add(button);
                Cashier.this.repaint();

            }
```

```java
        });
        //非会员收银
        button_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Cashier.this.dispose();
                CollectMoney cm = new CollectMoney(userId,0);
                cm.setVisible(true);
            }
        });
        JButton btnNewButton_2 = new JButton("返回");
        btnNewButton_2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Cashier.this.dispose();
                Login am = new Login();
                am.setVisible(true);
            }
        });
        btnNewButton_2.setBounds(361, 254,113, 43);
        contentPane.add(btnNewButton_2);

    }
}

```

3.CollectMoney

```java
package UI;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.JButton;

import service.CustomerService;
import service.ProductsService;

import entity.ProductInfo;
import entity.SaleHistory;
import entity.SaleSummary;
import service.SaleHistoryService;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.text.DateFormat;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;

/*
* 结算界面
* */
public class CollectMoney extends JFrame {

    private JPanel contentPane;
    private double totalAmount = 0;
    private JTable table;
    private JTextField proNo;
    private JTextField number;
    private SaleSummary[] ss = new SaleSummary[20];

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        CollectMoney frame = new CollectMoney(1003, 0);
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public CollectMoney(final int userId, final int customerNo) {
        this.setTitle("收银界面");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 599, 383);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("当前操作收银员编号:" + userId);
        lblNewLabel.setBounds(27, 313, 159, 21);
        contentPane.add(lblNewLabel);

        final JLabel lblNewLabel_1 = new JLabel("购物总价:" + totalAmount);
        lblNewLabel_1.setBounds(406, 316, 106, 15);
        contentPane.add(lblNewLabel_1);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(10, 66, 563, 237);
        contentPane.add(scrollPane);

        String[] cols = { "产品编号", "数量", "单价", "商品折扣", "收银员编号", "销售时间", "会员编号",
                "会员折扣", "总价格" };
        final DefaultTableModel model = new DefaultTableModel(cols, 0);
        table = new JTable(model);
        scrollPane.setViewportView(table);

        JLabel lblNewLabel_2 = new JLabel("商品编号:");
        lblNewLabel_2.setBounds(10, 10, 70, 15);
        contentPane.add(lblNewLabel_2);

        proNo = new JTextField();
        proNo.setBounds(90, 7, 89, 21);
        contentPane.add(proNo);
        proNo.setColumns(10);

        JLabel lblNewLabel_3 = new JLabel("会员编号:" + customerNo);
        lblNewLabel_3.setBounds(242, 316, 106, 15);
        contentPane.add(lblNewLabel_3);

        JLabel lblNewLabel_4 = new JLabel("数量:");
        lblNewLabel_4.setBounds(10, 41, 54, 15);
        contentPane.add(lblNewLabel_4);

        number = new JTextField();
        number.setColumns(10);
        number.setBounds(90, 38, 89, 21);
        contentPane.add(number);

        JButton btnNewButton = new JButton("确认添加");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                if (proNo.getText().isEmpty()) {
                    JOptionPane.showMessageDialog(null, "请输入商品编号", "错误提示",
                            JOptionPane.ERROR_MESSAGE);
                } else if (number.getText().isEmpty()) {
                    JOptionPane.showMessageDialog(null, "请输入商品数量", "错误提示",
                            JOptionPane.ERROR_MESSAGE);
                } else {
                    ProductsService ps = new ProductsService();
                    String ProductNo = proNo.getText();
                    int num = Integer.valueOf(number.getText());
                    ProductInfo pi = new ProductInfo();
                    pi = ps.getProductInfoByProductNoService(ProductNo);
                    SaleSummary s = new SaleSummary();

                    //System.out.println(pi.getCount());
                    if (pi.getCount() >= num) {
                        s.setProductNo(ProductNo);
                        s.setSaleCount(num);
                        if(s.setPrice(pi.getPrice())==0) {
                            JOptionPane.showMessageDialog(null, "该商品价格小于0！请联系货物管理员", "错误提示", JOptionPane.ERROR_MESSAGE);}
                        else{
                            s.setDiscount(pi.getDiscount());
                            s.setUserId(userId);
                            Calendar c = Calendar.getInstance();
                            DateFormat format = new SimpleDateFormat(
                                    "yyyy-MM-dd HH:mm:ss");
                            String time = format.format(c.getTime());
                            s.setSaleTime(time);
                            s.setCustomerNo(customerNo);

                            CustomerService cs = new CustomerService();
                            //保留两位小数
                            DecimalFormat df = new DecimalFormat("#.00");
                            /*
                             * 以下两行是判断会员折扣的
                             * */
 s.setMemberDiscount(Double.valueOf(df.format(cs.getMemberDiscountByCustomerNoService(customerNo))));
                            double amount = pi.getPrice() * num * pi.getDiscount()*s.getMemberDiscount();
//                              System.out.println("amount: "+ amount);


                            amount = Double.valueOf(df.format(amount));
                            s.setTotalAmount(amount);
                            // 添加到数组中
                            int i = 0;
                            for (; ss[i] != null; i++) {
                            }
                            ss[i] = s;
                            // 增加总金额
                            totalAmount += amount;
                            totalAmount = Double.valueOf(df.format(totalAmount));
                            Object[] row = { s.getProductNo(), s.getSaleCount(),
                                    s.getPrice(), s.getDiscount(), s.getUserId(), time,
                                    customerNo, s.getMemberDiscount(),
                                    s.getTotalAmount() };
                            model.addRow(row);
                            proNo.setText("");
                            number.setText("");
                            lblNewLabel_1.setText("购物总价:" + totalAmount);
                        }} else if(pi.getCount()!=-1){
                        JOptionPane.showMessageDialog(null,
                                "库存不足!剩余库存为" + pi.getCount(), "错误提示",
                                JOptionPane.ERROR_MESSAGE);
                    }
                }
            }
        });
        btnNewButton.setBounds(223, 6, 93, 23);
        contentPane.add(btnNewButton);

        JButton button = new JButton("删除");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int rowIndex = table.getSelectedRow();
                if(rowIndex<0){
                    JOptionPane.showMessageDialog(CollectMoney.this, "请选择要删除的行","错误提示",JOptionPane.ERROR_MESSAGE);
                }else{
                    if(JOptionPane.showConfirmDialog(CollectMoney.this, "确定删除第"+(rowIndex+1)+"行?")==0){
                        int i = rowIndex;
                        model.removeRow(i);
                        double ramount = ss[i].getPrice()*ss[i].getSaleCount()*ss[i].getDiscount()*ss[i].getMemberDiscount();
                        totalAmount -= ramount;
                        lblNewLabel_1.setText("购物总价:" + totalAmount);
                        for(;ss[i]!=null;i++){
                            ss[i] = ss[i+1];
                        }
                        ss[i] = null;
                    }
                }
            }
        });
        button.setBounds(223, 33, 93, 23);
        contentPane.add(button);

        JButton btnNewButton_1 = new JButton("清空");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                model.setRowCount(0);
                ss = new SaleSummary[20];
        totalAmount = 0;
        lblNewLabel_1.setText("购物总价:" + totalAmount);
      }
    });
    btnNewButton_1.setBounds(338, 6, 93, 23);
    contentPane.add(btnNewButton_1);

    JButton btnNewButton_2 = new JButton("结算");
    btnNewButton_2.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        for(int i= 0;ss[i]!=null;i++){
          SaleHistory sh = new SaleHistory();
          sh.setProductNo(ss[i].getProductNo());
          sh.setSaleCount(ss[i].getSaleCount());
          sh.setPrice(ss[i].getPrice());
          sh.setDiscount(ss[i].getDiscount());
          sh.setUserId(ss[i].getUserId());
          sh.setSaleTime(ss[i].getSaleTime());
          sh.setCustomerNo(ss[i].getCustomerNo());
          SaleHistoryService shs = new SaleHistoryService();
          shs.newSaleHistoryService(sh);
          ProductsService ps = new ProductsService();
          ps.updateProductsCountService(ss[i].getProductNo(), ss[i].getSaleCount());
        }
        CollectMoney.this.dispose();

        if(customerNo!=0){
          System.out.println("新增积分");
          CustomerService cs = new CustomerService();
          cs.updateCustomerScoreDao(customerNo, (int)(totalAmount));
        }

        System.out.println("算账");
        GiveChange gc = new GiveChange(totalAmount,userId,customerNo);
        gc.setVisible(true);

```

```
            }
        });
        btnNewButton_2.setBounds(454, 7, 106, 49);
        contentPane.add(btnNewButton_2);

        JButton btnNewButton_3 = new JButton("返回");
        btnNewButton_3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                CollectMoney.this.dispose();
                Cashier c = new Cashier(userId);
                c.setVisible(true);
            }
        });
        btnNewButton_3.setBounds(338, 33, 93, 23);
        contentPane.add(btnNewButton_3);

    }
}

```

4.CustomerManagement

```
package UI;

import java.util.List;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JButton;

import entity.Customers;

import service.CustomerService;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class CustomerManagement extends JFrame {
```

```java

    private JPanel contentPane;
    private JTable table;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        CustomerManagement frame = new CustomerManagement();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public CustomerManagement() {
        this.setTitle("会员信息管理");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 450, 340);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(10, 58, 414, 165);
        contentPane.add(scrollPane);

        String[] cols={"会员编号","会员用户名","会员联系方式","会员积分"};
        final DefaultTableModel model = new DefaultTableModel(cols,0);
        table = new JTable(model);
        CustomerService cs = new CustomerService();
        List<Customers> rows = cs.getAllCustomersService();
        for (Customers customers : rows) {
            Object[] row = {customers.getCustomerNo(),customers.getCustomerName(),customers.getPhone(),customers.getScore()};
            model.addRow(row);
        }
        scrollPane.setViewportView(table);

        JButton button = new JButton("新增");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                CustomerManagement.this.dispose();
                NewCustomer nc = new NewCustomer();
                nc.setVisible(true);
            }
        });
        button.setBounds(214, 25, 93, 23);
        contentPane.add(button);

        JButton button_1 = new JButton("删除");
        button_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int rowIndex = table.getSelectedRow();
                if(rowIndex<0){
                    JOptionPane.showMessageDialog(CustomerManagement.this, "请选择要删除的行","错误提示",JOptionPane.ERROR_MESSAGE);
                }else{
                    String cstNo = String.valueOf(model.getValueAt(rowIndex, 0));
                    if(JOptionPane.showConfirmDialog(CustomerManagement.this, "确定删除第"+(rowIndex+1)+"行?")==0){
                        CustomerService cs = new CustomerService();
                        if(cs.deleteCustomersService(cstNo)){
                            JOptionPane.showMessageDialog(CustomerManagement.this, "删除成功!");
                            CustomerManagement.this.dispose();
                            CustomerManagement frame = new CustomerManagement();
                            frame.setVisible(true);
                        }
                    }
                }
            }
        });
        button_1.setBounds(331, 25, 93, 23);
        contentPane.add(button_1);

        JButton button_2 = new JButton("返回主界面");
        button_2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                CustomerManagement.this.dispose();
                RootAdministrator ra = new RootAdministrator();
                ra.setVisible(true);
            }
        });
        button_2.setBounds(279, 233, 145, 26);
        contentPane.add(button_2);

    }
}

```

5.GiveChange

```
package UI;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.text.DecimalFormat;

public class GiveChange extends JFrame {

    private JPanel contentPane;
    private JTextField textField;
    private double change = 0;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
//      EventQueue.invokeLater(new Runnable() {
```

```
//      public void run() {
//        try {
//          GiveChange frame = new GiveChange(0,1003,0);
//          frame.setVisible(true);
//        } catch (Exception e) {
//          e.printStackTrace();
//        }
//      }
//    });
    GiveChange frame = new GiveChange(0,1003,0);
    frame.setVisible(true);
  }

  /**
   * Create the frame.
   * @param customerNo
   */
  public GiveChange(final double totalAmount,final int userId, final int customerNo) {
    this.setTitle("找零界面");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(400, 200, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("商品总金额:"+totalAmount);
    lblNewLabel.setFont(new Font("微软雅黑", Font.PLAIN, 23));
    lblNewLabel.setBounds(56, 33, 294, 57);
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("支付金额:");
    lblNewLabel_1.setFont(new Font("微软雅黑", Font.PLAIN, 16));
    lblNewLabel_1.setBounds(56, 100, 87, 32);
    contentPane.add(lblNewLabel_1);

    textField = new JTextField();
    textField.setBounds(153, 107, 130, 21);
    contentPane.add(textField);
    textField.setColumns(10);
```

```

        final JLabel label = new JLabel("找零金额:"+change);
        label.setFont(new Font("微软雅黑", Font.PLAIN, 16));
        label.setBounds(56, 183, 186, 32);
        contentPane.add(label);

        JButton btnNewButton = new JButton("确认支付");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                change = Double.valueOf(textField.getText())-totalAmount;
                DecimalFormat df = new DecimalFormat("#.00");
                change = Double.valueOf(df.format(change));
                if(change<0){
                    JOptionPane.showMessageDialog(null,"付款不足，请支付"+totalAmount+"元","错误提示",JOptionPane.ERROR_MESSAGE);}
                else{
                    label.setText("找零金额:"+change);
                }


            }
        });
        btnNewButton.setBounds(50, 150, 93, 23);
        contentPane.add(btnNewButton);

        JButton button = new JButton("返回");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                GiveChange.this.dispose();
                Cashier frame = new Cashier(userId);
                frame.setVisible(true);
            }
        });
        button.setBounds(173, 150, 93, 23);
        contentPane.add(button);
    }
}

```

6.GoodsAdministrator

```java
package UI;

import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.util.List;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JTable;

import entity.ProductInfo;

import service.ProductsService;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class GoodsAdministrator extends JFrame {

    private JPanel contentPane;
    private JTextField ProductNo;
    private JTextField ProductName;
    private JTable table;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {

        GoodsAdministrator frame = new GoodsAdministrator();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public GoodsAdministrator() {
        this.setTitle("商品管理员界面");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 603, 377);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(10, 53, 570, 224);
        contentPane.add(scrollPane);

        String[] cols={"商品编号","商品类型","商品名称","规格","单位","数量","单价","折扣"};
        final DefaultTableModel model = new DefaultTableModel(cols,0);
        table = new JTable(model);
        scrollPane.setViewportView(table);


        JLabel lblNewLabel = new JLabel("商品编号:");
        lblNewLabel.setBounds(36, 290, 70, 15);
        contentPane.add(lblNewLabel);

        ProductNo = new JTextField();
        ProductNo.setBounds(116, 287, 102, 21);
        contentPane.add(ProductNo);
        ProductNo.setColumns(10);

        JLabel label = new JLabel("商品名称:");
        label.setBounds(241, 290, 68, 15);
        contentPane.add(label);

        ProductName = new JTextField();
        ProductName.setColumns(10);
        ProductName.setBounds(308, 287, 102, 21);
        contentPane.add(ProductName);

        JButton button = new JButton("查询指定商品");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String productNo = "";
                String productName = "";
                productNo = ProductNo.getText();
```

```java
                productName = ProductName.getText();
                ProductsService ps = new ProductsService();
                ProductInfo productInfo = new ProductInfo();
                if(!productNo.isEmpty()){
                    model.setRowCount(0);
                    productInfo = ps.getProductInfoByProductNoService(productNo);
                    Object[] row = {productInfo.getProductNo(),productInfo.getCategoryName(),productInfo.getProductName(),productInfo.getType()
                        ,productInfo.getUnit(),productInfo.getCount(),productInfo.getPrice(),productInfo.getDiscount()};
                    model.addRow(row);
                }else if(!productName.isEmpty()){
                    model.setRowCount(0);
                    productInfo = ps.getProductInfoByproductNameService(productName);
                    Object[] row = {productInfo.getProductNo(),productInfo.getCategoryName(),productInfo.getProductName(),productInfo.getType()
                        ,productInfo.getUnit(),productInfo.getCount(),productInfo.getPrice(),productInfo.getDiscount()};
                    model.addRow(row);
                }else{
                        JOptionPane.showMessageDialog(GoodsAdministrator.this, "请输入要查询的商品信息","错误提示",JOptionPane.ERROR_MESSAGE);
                }
            }
        });
        button.setBounds(451, 287, 117, 23);
        contentPane.add(button);

        JButton btnNewButton = new JButton("增加新商品");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                GoodsAdministrator.this.dispose();
                NewProducts np = new NewProducts();
                np.setVisible(true);
            }
        });
        btnNewButton.setBounds(308, 10, 117, 23);
        contentPane.add(btnNewButton);

        JButton btnNewButton_1 = new JButton("删除商品信息");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int rowIndex = table.getSelectedRow();
                if(rowIndex<0){
                    JOptionPane.showMessageDialog(GoodsAdministrator.this, "请选择要删除的行","错误提示",JOptionPane.ERROR_MESSAGE);
                }else{
                    String productNo =   (String) model.getValueAt(rowIndex, 0);
                    if(JOptionPane.showConfirmDialog(GoodsAdministrator.this, "确定删除第"+(rowIndex+1)+"行?")==0){
                        ProductsService ps = new ProductsService();
                        if(ps.deleteProductsService(productNo)){
                            JOptionPane.showMessageDialog(GoodsAdministrator.this, "删除成功!");
                            GoodsAdministrator.this.dispose();
                            GoodsAdministrator frame = new GoodsAdministrator();
                            frame.setVisible(true);
                        }
                    }
                }
            }
        });
        btnNewButton_1.setBounds(446, 10, 122, 23);
        contentPane.add(btnNewButton_1);

        JButton showAllProducts = new JButton("查询所有商品");
        showAllProducts.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                model.setRowCount(0);
                ProductsService ps = new ProductsService();
                List<ProductInfo> rows = ps.getAllProductsService();
                for (ProductInfo productInfo : rows) {
                    Object[] row = {productInfo.getProductNo(),productInfo.getCategoryName(),productInfo.getProductName(),productInfo.getType()
                    ,productInfo.getUnit(),productInfo.getCount(),productInfo.getPrice(),productInfo.getDiscount()};
                    model.addRow(row);
                }
            }
        });
        showAllProducts.setBounds(27, 10, 117, 23);
        contentPane.add(showAllProducts);

        JButton button_1 = new JButton("更新库存信息");
        button_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int rowIndex = table.getSelectedRow();
                if(rowIndex<0){
                    JOptionPane.showMessageDialog(GoodsAdministrator.this, "请选择要更新的行","错误提示",JOptionPane.ERROR_MESSAGE);
                }else{
                    Object[] row = {model.getValueAt(rowIndex, 0),model.getValueAt(rowIndex, 2),model.getValueAt(rowIndex, 5),model.getValueAt(rowIndex, 7)};
                    GoodsAdministrator.this.dispose();
                    UpdateProducts up = new UpdateProducts(row);
                    up.setVisible(true);
                }
            }
        });
        button_1.setBounds(172, 10, 126, 23);
        contentPane.add(button_1);
        JButton btnNewButton_6 = new JButton("返回");
        btnNewButton_6.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                GoodsAdministrator.this.dispose();
                RootAdministrator am = new RootAdministrator();
                am.setVisible(true);
            }
        });
        btnNewButton_6.setBounds(451, 317, 117, 23);
        contentPane.add(btnNewButton_6);
    }
}

```

7.Login

```
package UI;

import java.awt.*;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import javax.swing.ImageIcon;
import javax.swing.JButton;

import service.UsersService;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;

public class Login extends JFrame {

    private JPanel contentPane;//面板
    private JTextField userName;
    final JPasswordField pwd =new JPasswordField();//密码不可更改
    private int roleId = 0;//角色

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        Login frame = new Login();
        frame.setVisible(true);
```

```java

    }

    /**
     * Create the frame.
     */
    public Login() {

        this.setTitle("超市管理系统");//标题
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 470, 400);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));//控制面板边界
        setContentPane(contentPane);
        contentPane.setLayout(null);
        JLabel beijing =new JLabel();
        /*beijing.setBounds(400, 200, 450, 300);*/
        ImageIcon beijing1 = new ImageIcon("E:\\平时作业\\大二上\\JAVA\\算法个人作业\\Final project\\src\\UI\\背景.jpg");
        beijing.setBounds(0,0,520,436);
        beijing.setIcon(new ImageIcon(beijing1.getImage()));

        JLabel Title = new JLabel("超 市 管 理 系 统");
        Title.setBounds(125,15,400,30);
        Title.setFont(new Font("黑体",Font.PLAIN,26));
        contentPane.add(Title);

        JLabel Jusername = new JLabel("员工名:");//标签
        Jusername.setBounds(82, 98, 54, 15);
        Jusername.setFont(new Font("宋体",Font.BOLD,14));
        contentPane.add(Jusername);

        userName = new JTextField();
        userName.setBounds(186,95,141,21);
        contentPane.add(userName);    //输入文本框

        JLabel Jpwd = new JLabel("密码:");
        Jpwd.setBounds(82, 144, 79, 15);
        Jpwd.setFont(new Font("宋体",Font.BOLD,14));
        contentPane.add(Jpwd);

        pwd.setBounds(186,137,141,21);
        pwd.setColumns(10);    //限制长度
        contentPane.add(pwd);

        JLabel label_1 = new JLabel("员工类别:");
        label_1.setBounds(80, 186, 100, 15);
        label_1.setFont(new Font("宋体",Font.BOLD,14));
        contentPane.add(label_1);



        final JComboBox comboBox = new JComboBox();  //将按钮或可编辑字段与下拉列表组合的组件
        comboBox.setModel(new DefaultComboBoxModel(new String[] {"主管理员", "商品管理员", "收银员"}));
        //利用DefaultComboBoxModel类可以很方便地做到动态更改JComboBox的项目值
        comboBox.addItemListener(new ItemListener() {   //下拉列表框的选项监听，被选中的显示，未被选中的隐藏。
            @Override
            public void itemStateChanged(ItemEvent e) {
                if(ItemEvent.SELECTED==e.getStateChange()){
                    roleId = comboBox.getSelectedIndex();
                }
            }
        });
        comboBox.setBounds(186, 186, 79, 21);
        contentPane.add(comboBox);

        JButton button = new JButton("登录");
        button.addActionListener(new ActionListener() {  //设置一个监听
            public void actionPerformed(ActionEvent e) {
                String UserName = userName.getText();    //获取输入的名字
                String Pwd = pwd.getText();          //获取输入的密码
                UsersService user = new UsersService();            //用户相关服务
                if(user.checkUsersService(UserName, Pwd, roleId)){      //将输入的三个参数与数据库中的进行比对
                    Login.this.dispose();     //使用dispose()方法关闭的窗体可以使用pack 或 show 方法恢复，并且可以恢复到dispose前的状态
                    if(roleId == 0){//主管理员
                        RootAdministrator ra = new RootAdministrator();
                        ra.setVisible(true);
                    }else if(roleId == 1){//货物管理
                        GoodsAdministrator ga = new GoodsAdministrator();
                        ga.setVisible(true);
                    }else if(roleId == 2){//收银员
                        UsersService us = new UsersService();
                        int userId = us.getUserIdByUserNameService(UserName);
                        Cashier c = new Cashier(userId);
                        c.setVisible(true);
                    }
                }else{
                    JOptionPane.showMessageDialog(Login.this, "登陆失败!");
                }
            }
        });
        button.setBounds(100, 243, 93, 43);
        contentPane.add(button);

        JButton button_1 = new JButton("退出");
        button_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Login.this.dispose();
            }
        });
        button_1.setBounds(264, 243, 93, 43);
        contentPane.add(button_1);
        this.add(beijing);
    }

}

```

8.NewAdministrator

```
package UI;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
```

```java
import javax.swing.border.EmptyBorder;
import javax.swing.ButtonGroup;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JRadioButton;
import javax.swing.JButton;

import service.UsersService;

import entity.Users;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class NewAdministrator extends JFrame {

    private JPanel contentPane;
    private JTextField userName;
    private JTextField userId;
    final JPasswordField pwd =new JPasswordField();
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        NewAdministrator frame = new NewAdministrator();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public NewAdministrator() {
        this.setTitle("新增管理员");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 505, 319);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel labe0 = new JLabel("员工编号:");
        labe0.setBounds(85, 7, 104, 15);
        contentPane.add(labe0);
        userId = new JTextField();
        userId.setColumns(10);
        userId.setBounds(186, 7, 168, 21);
        contentPane.add(userId);

        JLabel label = new JLabel("用户名:");
        label.setBounds(85, 50, 54, 15);
        contentPane.add(label);

        userName = new JTextField();
        userName.setColumns(10);
        userName.setBounds(186, 50, 168, 21);
        contentPane.add(userName);

        JLabel label_1 = new JLabel("密码:");
        label_1.setBounds(85, 95, 54, 15);
        contentPane.add(label_1);

        pwd.setColumns(10);
        pwd.setBounds(186, 92, 168, 21);
        contentPane.add(pwd);



        JLabel label_2 = new JLabel("管理员级别:");
        label_2.setBounds(85, 144, 66, 15);
        contentPane.add(label_2);

        final JRadioButton rdbtnNewRadioButton = new JRadioButton("主管理员");
        rdbtnNewRadioButton.setBounds(173, 140, 93, 23);
        contentPane.add(rdbtnNewRadioButton);

        final JRadioButton radioButton = new JRadioButton("货物管理员");
        radioButton.setBounds(268, 140, 94, 23);
        contentPane.add(radioButton);

        final JRadioButton radioButton_1 = new JRadioButton("收银员");
        radioButton_1.setBounds(364, 140, 85, 23);
        contentPane.add(radioButton_1);
        ButtonGroup bg = new ButtonGroup();
        bg.add(rdbtnNewRadioButton);
        bg.add(radioButton);
        bg.add(radioButton_1);
        JButton btnNewButton = new JButton("提交");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Users user = new Users();               //连接User
                String s = userId.getText();
                int i = Integer.parseInt(s);  //Integer的默认值为null
                user.setUserId(i);
                user.setUserName(userName.getText());
                user.setPassword(pwd.getText());
                if(rdbtnNewRadioButton.isSelected()){
                    user.setRoleId(0);   //对应的相应地级别
                }else if(radioButton.isSelected()){
                    user.setRoleId(1);
                }else if(radioButton_1.isSelected()){
                    user.setRoleId(2);
                }else{
                    JOptionPane.showMessageDialog(NewAdministrator.this, "请选择管理员类别!","错误提示",JOptionPane.ERROR_MESSAGE);
                }
                UsersService us = new UsersService();
                if(us.newUsersService(user)){
                    JOptionPane.showMessageDialog(NewAdministrator.this, "添加管理员成功!");
                }else{
                    JOptionPane.showMessageDialog(NewAdministrator.this, "添加管理员失败!");
                }
            }
        });
        btnNewButton.setBounds(106, 194, 93, 23);
        contentPane.add(btnNewButton);

        JButton btnNewButton_1 = new JButton("返回");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                NewAdministrator.this.dispose();
                AdministratorManagement am = new AdministratorManagement();
                am.setVisible(true);
            }
        });
        btnNewButton_1.setBounds(241, 194, 93, 23);
        contentPane.add(btnNewButton_1);
    }
}

```

9.NewCustomer

```
package UI;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;

import service.CustomerService;

import entity.Customers;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class NewCustomer extends JFrame {


    private JPanel contentPane;
    private JTextField cstName;
    private JTextField telephone;
    private JTextField cstNo;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {

        NewCustomer frame = new NewCustomer();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public NewCustomer() {
        this.setTitle("新增会员");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("会员编号:");
        lblNewLabel.setBounds(80, 35, 76, 15);
        contentPane.add(lblNewLabel);

        cstNo = new JTextField();
        cstNo.setBounds(166, 32, 148, 21);
        contentPane.add(cstNo);
        cstNo.setColumns(10);

        JLabel label = new JLabel("会员名:");
        label.setBounds(89, 80, 54, 15);
        contentPane.add(label);

        cstName = new JTextField();
        cstName.setColumns(10);
        cstName.setBounds(166, 77, 148, 21);
        contentPane.add(cstName);

        JLabel label_1 = new JLabel("会员联系方式:");
        label_1.setBounds(58, 131, 85, 15);
        contentPane.add(label_1);

        telephone = new JTextField();
        telephone.setColumns(10);
        telephone.setBounds(166, 128, 148, 21);
        contentPane.add (telephone);

        JButton btnNewButton = new JButton("提交");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Customers customer = new Customers();
                customer.setCustomerNo(cstNo.getText());
                customer.setCustomerName(cstName.getText());
                customer.setPhone(telephone.getText());
                CustomerService cs = new CustomerService();
                //if(cs.checkCustomerNoDao(cstNo.getText())){
                int flag=cs.newCustomersService(customer);
                if(flag != 0){
                    JOptionPane.showMessageDialog(NewCustomer.this, "添加会员成功!会员号码是"+customer.getCustomerNo());
                }else{
                    JOptionPane.showMessageDialog(NewCustomer.this, "添加会员失败!");
                }
                //}
            }
        });
        btnNewButton.setBounds(99, 189, 93, 23);
        contentPane.add(btnNewButton);

        JButton button = new JButton("返回");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                NewCustomer.this.dispose();
                CustomerManagement cm = new CustomerManagement();
                cm.setVisible(true);
            }
        });
        button.setBounds(243, 189, 93, 23);
        contentPane.add(button);

    }
}

```

10.NewProducts

```java
package UI;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JButton;
import javax.swing.DefaultComboBoxModel;

import service.ProductsService;

import entity.Products;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;

public class NewProducts extends JFrame {

    private JPanel contentPane;
    private JTextField ProductNo;
    private JTextField ProductName;
    private JTextField Type;
    private JTextField Unit;
    private JTextField pCount;
    private JTextField Price;
    private JTextField Discount;
    private String categoryId = "000";

    /**
     * Launch the application.
     */
    public static void main(String[] args) {

        NewProducts frame = new NewProducts();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public NewProducts() {
        this.setTitle("新增商品信息");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 450, 407);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("商品编号:");
        lblNewLabel.setBounds(67, 22, 72, 15);
        contentPane.add(lblNewLabel);

        ProductNo = new JTextField();
        ProductNo.setBounds(164, 19, 144, 21);
        contentPane.add(ProductNo);
        ProductNo.setColumns(10);

        JLabel label = new JLabel("商品名称:");
        label.setBounds(67, 50, 72, 15);
        contentPane.add(label);

        ProductName = new JTextField();
        ProductName.setColumns(10);
        ProductName.setBounds(164, 47, 144, 21);
        contentPane.add(ProductName);

        JLabel label_1 = new JLabel("商品类别:");
        label_1.setBounds(67, 75, 72, 15);
        contentPane.add(label_1);

        JLabel label_2 = new JLabel("规格:");
        label_2.setBounds(67, 119, 54, 15);
        contentPane.add(label_2);

        Type = new JTextField();
        Type.setColumns(10);
        Type.setBounds(164, 116, 144, 21);
        contentPane.add(Type);

        JLabel label_3 = new JLabel("单位:");
        label_3.setBounds(67, 147, 54, 15);
        contentPane.add(label_3);

        Unit = new JTextField();
        Unit.setColumns(10);
        Unit.setBounds(164, 144, 144, 21);
        contentPane.add(Unit);

        JLabel label_4 = new JLabel("数量:");
        label_4.setBounds(67, 175, 54, 15);
        contentPane.add(label_4);

        pCount = new JTextField();
        pCount.setColumns(10);
        pCount.setBounds(164, 172, 144, 21);
        contentPane.add(pCount);

        JLabel label_5 = new JLabel("售价:");
        label_5.setBounds(67, 203, 54, 15);
        contentPane.add(label_5);

        Price = new JTextField();
        Price.setColumns(10);
        Price.setBounds(164, 200, 144, 21);
        contentPane.add(Price);

        JLabel label_6 = new JLabel("折扣:");
        label_6.setBounds(67, 231, 54, 15);
        contentPane.add(label_6);

        Discount = new JTextField();
        Discount.setColumns(10);
        Discount.setBounds(164, 228, 144, 21);
        contentPane.add(Discount);

        final JComboBox comboBox = new JComboBox();
        comboBox.setModel(new DefaultComboBoxModel(new String[] {" ","水果", "主食", "蔬菜", "文体办公", "家电五金", "洗浴用品"}));
        comboBox.addItemListener(new ItemListener() {
            @Override
            public void itemStateChanged(ItemEvent e) {
                if(ItemEvent.SELECTED==e.getStateChange()){
```

```
          categoryId = comboBox.getSelectedItem().toString();
          //categoryId = "00"+(comboBox.getSelectedIndex()+1);   //转换为数字
        }
      }
    });
    comboBox.setBounds(164, 78, 82, 21);
    contentPane.add(comboBox);

    JButton btnNewButton = new JButton("保存");
    btnNewButton.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        Products product = new Products();
        if(ProductNo.getText().isEmpty()){
          JOptionPane.showMessageDialog(null,"商品编号不能为空","错误提示",JOptionPane.ERROR_MESSAGE);
        }else{
          product.setProductNo(ProductNo.getText());
          product.setCategoryId(categoryId);
          if(ProductName.getText().isEmpty()){
            JOptionPane.showMessageDialog(null,"商品名称不能为空","错误提示",JOptionPane.ERROR_MESSAGE);
          }else{
            product.setProductName(ProductName.getText());
            product.setType(Type.getText());
            if(Unit.getText().isEmpty()){
              JOptionPane.showMessageDialog(null,"商品单位不能为空","错误提示",JOptionPane.ERROR_MESSAGE);
            }else{
              product.setUnit(Unit.getText());
              if(pCount.getText().isEmpty()){
                JOptionPane.showMessageDialog(null,"库存数量不能为空","错误提示",JOptionPane.ERROR_MESSAGE);
              }else{
                product.setCount(Integer.valueOf(pCount.getText()));
                String str1=Price.getText();
                int a = 0;
                try {
```

```
                                    a = Integer.parseInt(str1);
                                } catch (NumberFormatException e1) {
                                    e1.printStackTrace();
                                }
                                if(Price.getText().isEmpty()||a<=0){
                                    JOptionPane.showMessageDialog(null,"单价不能为空或负数","错误提示",JOptionPane.ERROR_MESSAGE);
                                }else{
                                    product.setPrice(Double.valueOf(Price.getText()));
                                    if(Discount.getText().isEmpty()){
                                        JOptionPane.showMessageDialog(null,"折扣不能为空","错误提示",JOptionPane.ERROR_MESSAGE);
                                    }else{
                                        product.setDiscount(Double.valueOf(Discount.getText()));
                                        ProductsService ps = new ProductsService();
                                        if(ps.checkProductsNoDao(ProductNo.getText())){
                                            if(ps.newProductsService(product)){
                                                JOptionPane.showMessageDialog(NewProducts.this, "添加商品成功!");
                                                NewProducts.this.dispose();
                                                NewProducts np = new NewProducts();
                                                np.setVisible(true);
                                            }else{
                                                JOptionPane.showMessageDialog(NewProducts.this, "添加商品失败!");
                                            }
                                        }
                                    }
                                }
```

```
                            }
                        }
                    }
                }
            }
        });
        btnNewButton.setBounds(115, 275, 175, 23);
        contentPane.add(btnNewButton);

        JButton button = new JButton("返回商品管理界面");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                NewProducts.this.dispose();
                GoodsAdministrator ga = new GoodsAdministrator();
                ga.setVisible(true);
            }
        });
        button.setBounds(115, 316, 175, 23);
        contentPane.add(button);
    }
}

```

11.RootAdministrator

```
package UI;

import java.awt.*;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class RootAdministrator extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
```

```java
     */
    public static void main(String[] args) {
        RootAdministrator frame = new RootAdministrator();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public RootAdministrator() {
        this.setTitle("主管理员界面");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 500, 500);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);   //清空布局管理器

        //修改字体大小
        JLabel lblNewLabel = new JLabel("管理员,请选择要执行的功能模块");
        lblNewLabel.setFont(new Font("微软雅黑", Font.PLAIN, 22));
        lblNewLabel.setSize(new Dimension(20,20));
        lblNewLabel.setBounds(78, 33, 318, 49);
        contentPane.add(lblNewLabel);

        JButton btnNewButton = new JButton("管理员工信息");

        btnNewButton.setFont(new Font("微软雅黑", Font.PLAIN, 17));

        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                RootAdministrator.this.dispose();
                AdministratorManagement am = new AdministratorManagement(); //打开员工信息管理界面
                am.setVisible(true);
            }
        });
        btnNewButton.setBounds(170, 120, 159, 41);
        contentPane.add(btnNewButton);

        JButton button = new JButton("管理顾客信息");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                RootAdministrator.this.dispose();
                CustomerManagement cm = new CustomerManagement();   // 打开会员信息管理界面
                cm.setVisible(true);
            }
        });
        button.setFont(new Font("微软雅黑", Font.PLAIN, 17));
        button.setBounds(170, 200, 159, 41);
        contentPane.add(button);

        JButton btnNewButton1 = new JButton("管理商品信息");
        btnNewButton1.setFont(new Font("微软雅黑", Font.PLAIN, 17));
        btnNewButton1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                RootAdministrator.this.dispose();
                GoodsAdministrator qh = new GoodsAdministrator();   // 打开商品信息管理界面
                qh.setVisible(true);
            }
        });
        btnNewButton1.setBounds(170, 280, 159, 41);
        contentPane.add(btnNewButton1);

        JButton btnNewButton_1 = new JButton("返回登录界面");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                RootAdministrator.this.dispose();
                Login lg = new Login();           // 返回登陆界面
                lg.setVisible(true);
            }
        });
        btnNewButton_1.setBounds(350, 360, 117, 35);
        contentPane.add(btnNewButton_1);
    }

}

```

12.UpdateProducts

```java
package UI;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;

import service.ProductsService;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class UpdateProducts extends JFrame {

    private JPanel contentPane;
    private JTextField productName;
    private JTextField addCount;
    private JTextField discount;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {

        Object[] row = {"dd","ww","ww","1"};
        UpdateProducts frame = new UpdateProducts(row);
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public UpdateProducts(final Object[] row) {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(400, 200, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("商品名称:");
        lblNewLabel.setBounds(98, 46, 68, 15);
        contentPane.add(lblNewLabel);

        productName = new JTextField();
        productName.setEditable(false);
        productName.setText((String) row[1]);
        productName.setBounds(192, 43, 151, 21);
        contentPane.add(productName);
        productName.setColumns(10);

        JLabel label = new JLabel("新增库存:");
        label.setBounds(98, 94, 68, 15);
        contentPane.add(label);

        addCount = new JTextField();
        addCount.setColumns(10);
        addCount.setBounds(192, 91, 151, 21);
        contentPane.add(addCount);

        JLabel label_1 = new JLabel("折扣:");
        label_1.setBounds(98, 139, 54, 15);
        contentPane.add(label_1);

        discount = new JTextField();
        discount.setText(String.valueOf(row[3]));
        discount.setColumns(10);
        discount.setBounds(192, 136, 151, 21);
        contentPane.add(discount);

        JButton btnNewButton = new JButton("更新");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int pcount = (Integer)(row[2]) +
Integer.valueOf(addCount.getText());
                double Discount =
Double.valueOf(discount.getText());
                String productNo = (String) row[0];
                ProductsService ps = new ProductsService();
                if(ps.updateProductsService(productNo,
pcount, Discount)){
```

```
                    JOptionPane.showMessageDialog(UpdateProducts.this, "更新库存信息成功!当前库存:"+pcount+"当前折扣:"+Discount);
                }else{
                    JOptionPane.showMessageDialog(UpdateProducts.this, "更新库存信息失败!");
                }

            }
        });
        btnNewButton.setBounds(112, 190, 93, 23);
        contentPane.add(btnNewButton);

        JButton button = new JButton("返回");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                UpdateProducts.this.dispose();
                GoodsAdministrator ga = new GoodsAdministrator();
                ga.setVisible(true);
            }
        });
        button.setBounds(231, 190, 93, 23);
        contentPane.add(button);
    }
}


```

5.main

1.main

```
package main;
import UI.Login;
public class main {
    public static void main(String[] args) {
        Login logooin = new Login();
        logooin.setVisible(true);

    }
}
```

6.iml

```
.iml文件


<?xml version="1.0" encoding="UTF-8"?>
<module type="JAVA_MODULE" version="4">
  <component name="NewModuleRootManager" LANGUAGE_LEVEL="JDK_1_8">
    <output url="file://$MODULE_DIR$/out/production/Supermarket" />
    <output-test url="file://$MODULE_DIR$/out/test/Supermarket" />
    <exclude-output />
    <content url="file://$MODULE_DIR$">
      <sourceFolder url="file://$MODULE_DIR$/src" isTestSource="false" />
    </content>
    <orderEntry type="inheritedJdk" />
    <orderEntry type="sourceFolder" forTests="false" />
    <orderEntry type="library" name="mssql-jdbc-6.4.0.jre8" level="project" />
  </component>
</module>
```