**小型超市商品管理系统的设计与实现**

目 录

[1 引言 1](#_Toc106479006)

[1.1课题背景和意义 1](#_Toc106479007)

[1.2 课题内容 1](#_Toc106479008)

[2 系统需求分析及相关技术介绍 1](#_Toc106479009)

[2.1 功能需求分析 1](#_Toc106479010)

[2.2 可行性分析 1](#_Toc106479011)

[2.3 系统运行环境 2](#_Toc106479012)

[2.4 相关技术介绍 2](#_Toc106479013)

[3 系统总体设计 2](#_Toc106479014)

[3.1 系统功能结构设计 2](#_Toc106479015)

[3.2 系统功能流程设计 3](#_Toc106479016)

[3.2.1 登录功能流程设计 3](#_Toc106479017)

[3.2.2商品售出流程设计 4](#_Toc106479018)

[3.2.3 个人信息管理流程设计 5](#_Toc106479019)

[3.2.4流水信息管理流程设计 5](#_Toc106479020)

[3.2.5商品信息管理流程设计 6](#_Toc106479021)

[3.2.6人员信息管理流程设计 7](#_Toc106479022)

[3.3 数据库设计 8](#_Toc106479023)

[3.3.1概念结构设计 8](#_Toc106479024)

[3.3.2逻辑结构设计 10](#_Toc106479025)

[4系统详细设计 11](#_Toc106479026)

[4.1 登录模块详细设计 12](#_Toc106479027)

[4.2个人信息管理模块模块详细设计 12](#_Toc106479028)

[4.3商品售出模块详细设计 13](#_Toc106479029)

[4.4商品信息管理模块详细设计 13](#_Toc106479030)

[4.5流水信息管理模块详细设计 13](#_Toc106479031)

[4.6人员信息管理模块详细设计 13](#_Toc106479032)

[5 系统实现 14](#_Toc106479033)

[5.1 登录模块实现 14](#_Toc106479034)

[5.2 个人信息管理模块实现 16](#_Toc106479035)

[5.3 商品售出管理模块实现 18](#_Toc106479036)

[5.4商品信息管理模块详细设计 20](#_Toc106479037)

[5.5流水信息管理模块详细设计 27](#_Toc106479038)

[5.6人员信息管理模块详细设计 30](#_Toc106479039)

[6 总结 36](#_Toc106479040)

小型超市商品管理系统

许意杨 202083290379 刘辉 202083290374

南京信息工程大学计算机与软件学院，江苏 南京 210044

# 1 引言

## 1.1课题背景和意义

21世纪,超市的竞争也进入到 了一个全新的领域，竞争已不再是规模的竞争而是技术的竞争、管理的竞争、人才的竞争。技术的提升和管理的升级是超市业的竞争核心。管理超市各类数据是一项繁重的工作，通过人工来核对数据，费时费力，容易出现差错。一个合适、高效的数据库管理系统能够很好的解决小心超市商品管理的需求。

## 1.2 课题内容

针对小型超市的特点，我们开发了对应的数据管理系统来更好的实现超市数据的管理以及使用。

# 2 系统需求分析及相关技术介绍

## 2.1 功能需求分析

小型超市商品管理系统主要研究商品库存信息的管理，员工以及客户信息的管理，商品售出报表的管理。超市管理系统的使用者主要有两类人，管理员即经理还有员工，两者分别拥有不同的权限。员工可以修改个人信息以及负责商品的售出，产生流水单号。经理拥有所有的权限，可以对商品信息，流水统计，员工以及客户的信息进行增删改查等。

## 2.2 可行性分析

本套超市管理系统主要针对着小型超市而开发，不仅可以提高超市的工作效率，减少工作人员数量，减少劳动力资本的投入，还可以保证超市内部数据的精确性和实时性，进而帮助超市的正常运营。

在技术层面上，本套超市管理系统，使用了最为基本的C/S技术配置,  这样以便于大多数的超市都可以使用本套管理系统，不需要特别配置一些高难度的软硬件附件，基本的计算机都可以保证本套超市管理系统的正常使用和运行。 减少因为软硬件配置不搭配而产生的运行错误，或者是无法使用。

在经济层面上，本套系统的开发没有使用昂贵的系统配置，但是一套软件从开发到正式使用需要经过很长一段时间的技术研究和程序开发，所以还是有一定的基本开销。而且日后的软件系统的升级维护对于应用主体也没有很高的要求，只需要满足基本的经济保障即可。

在操作层面上，就目前管理者对超市管理系统的需求而言，本套超市管理系统是可以完全满足这些要求。不论是从技术上，还是从系统自身的操作性而言,  都采用了人机交互型的友好性配置。操作界面简洁易懂，容易操作，没有复杂的管理程序。操作人员只要是具备计算机的基本操作技能就都可以顺利使用此系统.  不会带来其他的不必要的麻烦和困扰。所以本套系统在技术、经济，以及操作上都是可行的，可以进行开发。

## 2.3 系统运行环境

Win10，Microsoft Visual Studio，SQL Server2019

## 2.4 相关技术介绍

本文使用C#和WinForm应用程序来进行系统的开发。C#是专门为.NET平台设计的程序语言，它由C++和Java语言发展而来，集成C++和Java的优秀特点，并使用事件驱动、完全面向对象的编程模式。WinForm应用程序又称窗体应用程序。WinForm应用程序具有事件驱动、能响应复杂的操作、可产生丰富的回馈信息等优点。

# 3 系统总体设计

## 3.1 系统功能结构设计

本系统的功能主要给予使用系统的两类使用者，员工以及经理。首先是登录功能，完成两者的登录。员工登录后可以实现商品的销售以及个人信息的修改。管理员登录后可以对流水信息进行统计查询，商品的信息进行管理，员工以及顾客的信息进行管理，还有管理员也可以对自身信息进行管理。系统功能模块划分如图3-1所示。

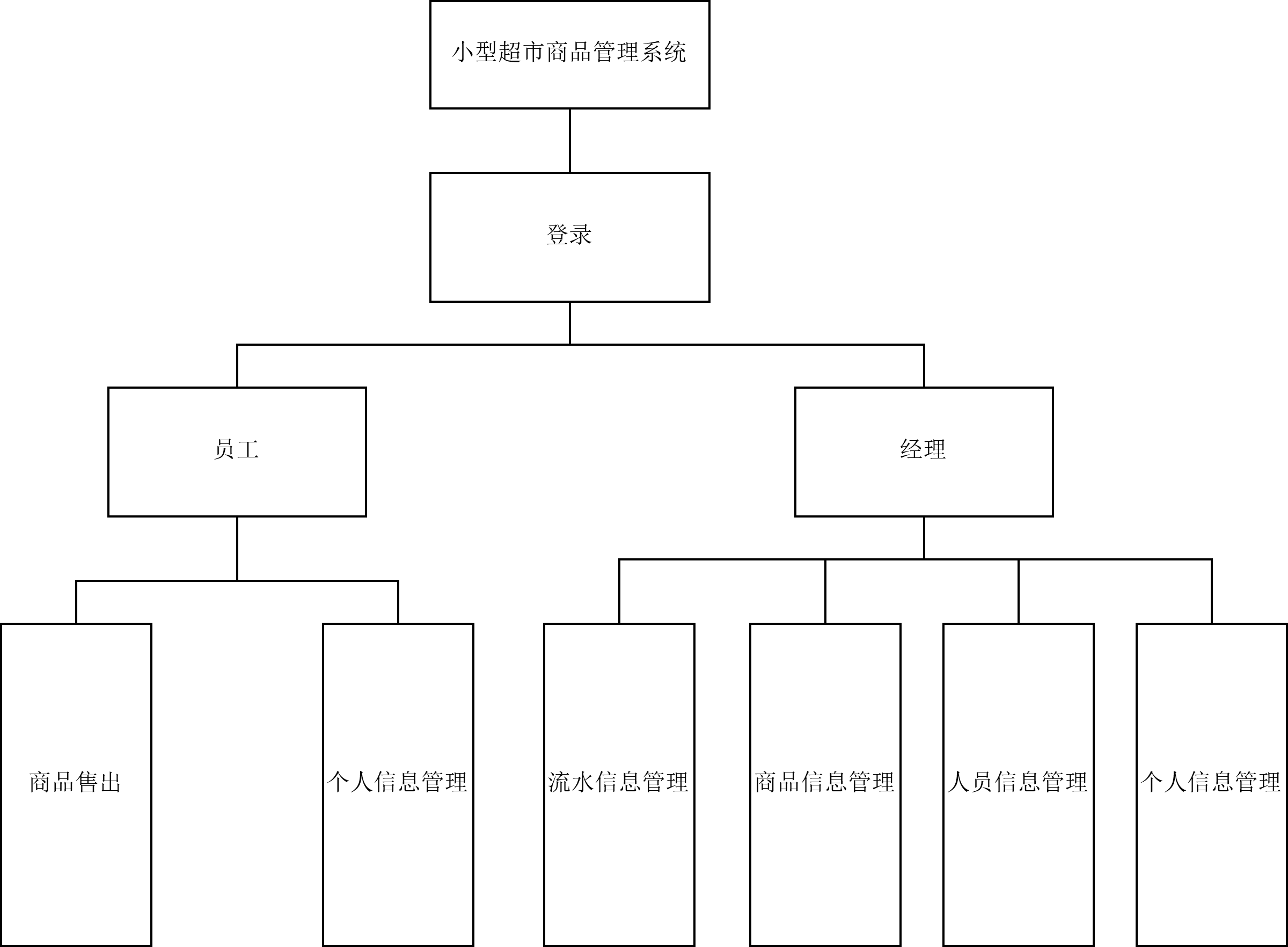


图3-1 系统功能结构图

## 3.2 系统功能流程设计

### 3.2.1 登录功能流程设计

登录过程可以直接退出系统，也可以在主界面选择员工或管理员身份，输入账号密码，进入数据库，通过查找匹配信息来完成登录。登录流程如图3-2-1所示。

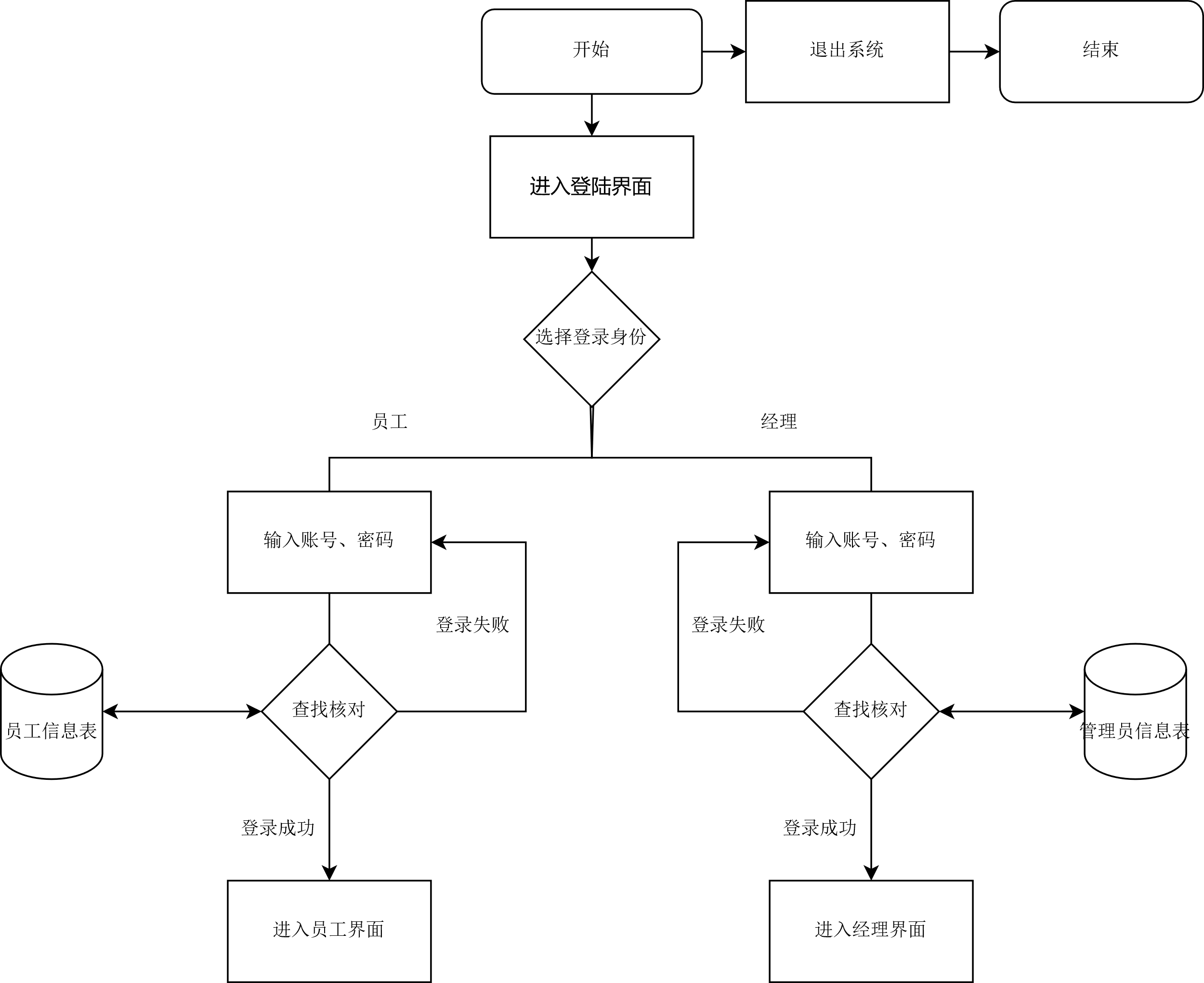


图3-2-1登录流程图

### 3.2.2商品售出流程设计

进入员工界面后可以进入商品售出界面。员工需要输入商品信息，以及购买的个数来完成商品售出。商品售出流程如图3-2-2所示。

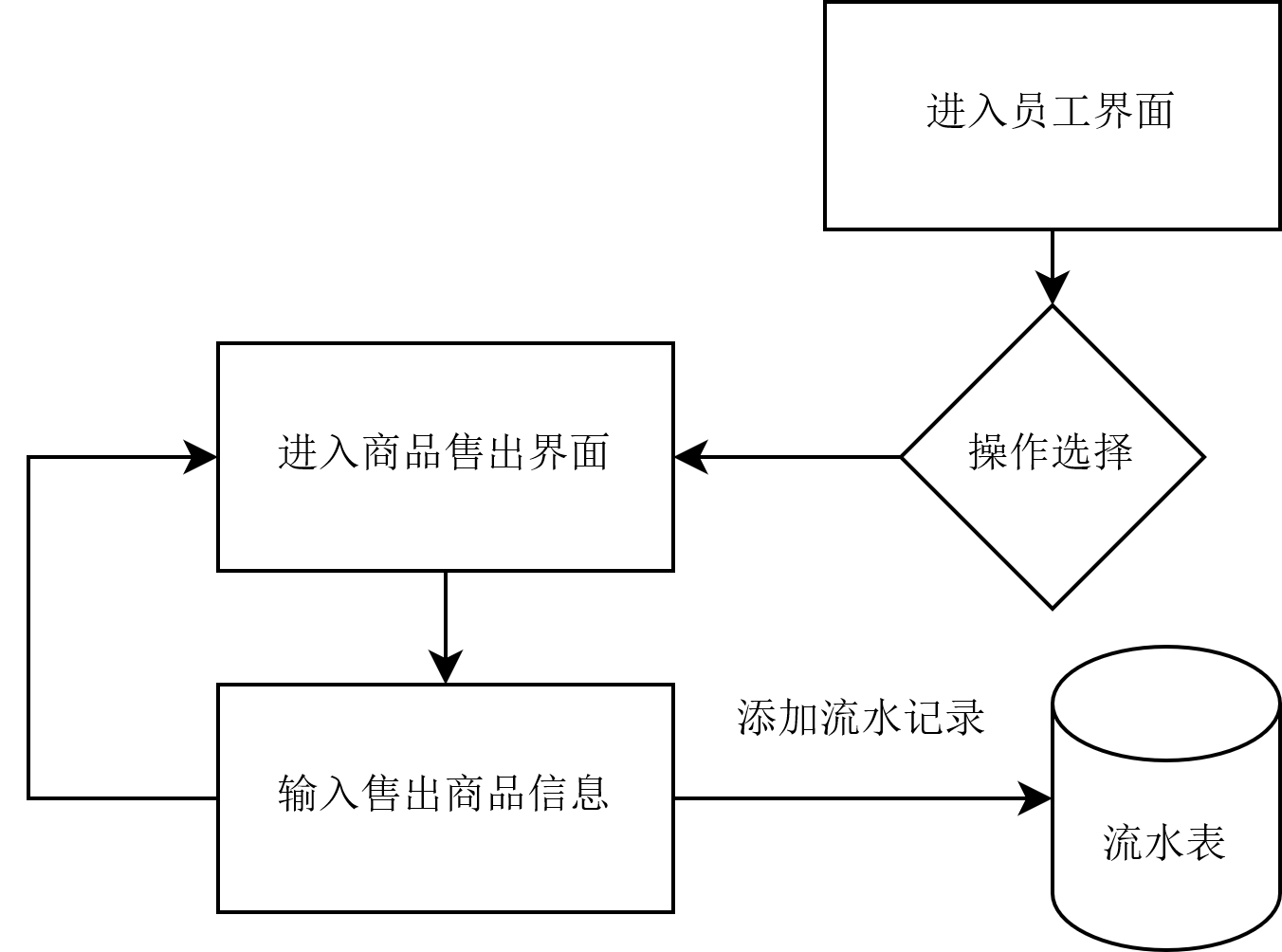


图3-2-2商品售出流程图

### 3.2.3 个人信息管理流程设计

在进入了员工或经理的主界面后都可以进入个人信息管理界面。进入后便是通过数据库查看到的个人信息界面，还可以选择进行个人信息的修改，密码修改以及返回等功能。个人信息流程如图3-2-3所示。

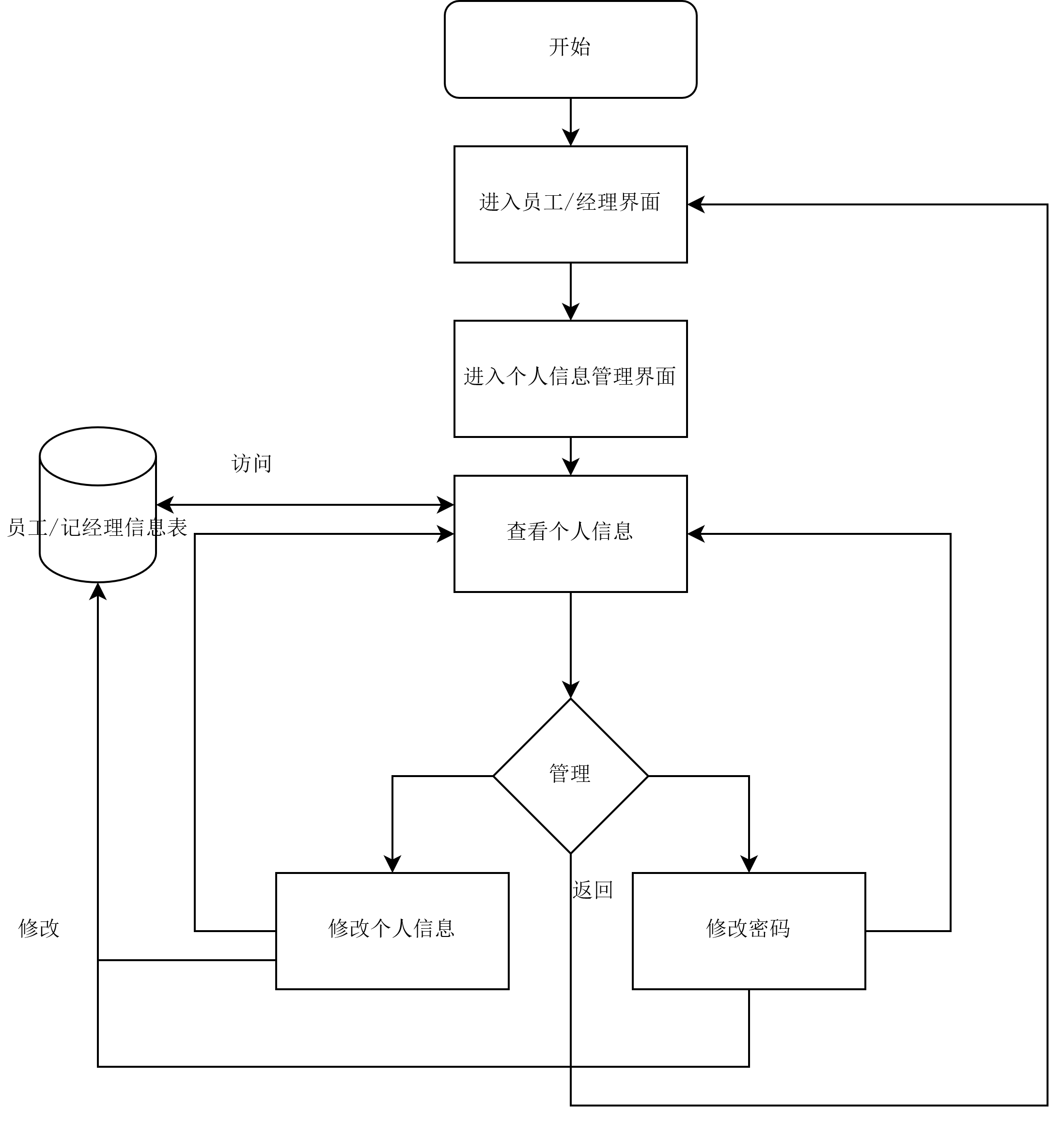


图3-2-3个人信息流程图

### 3.2.4流水信息管理流程设计

在经理进入主界面后可以选择进入流水信息管理界面。流水管理主要为流水单的查看和修改。进入流水管理界面后可选择根据单号查看，以及选择流水单修改。流水信息流程如图3-2-4所示。

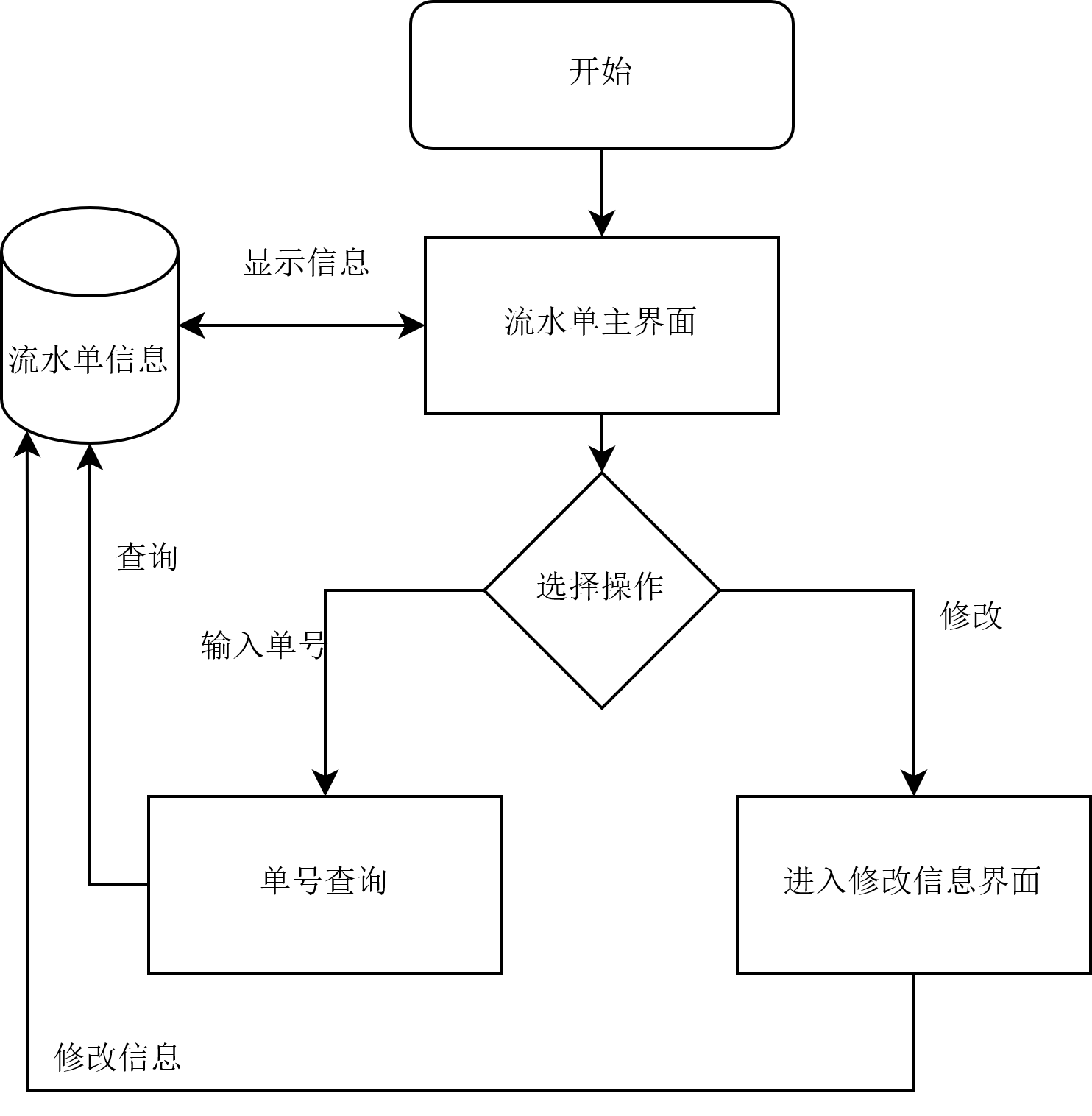


图3-2-4流水信息流程图

### 3.2.5商品信息管理流程设计

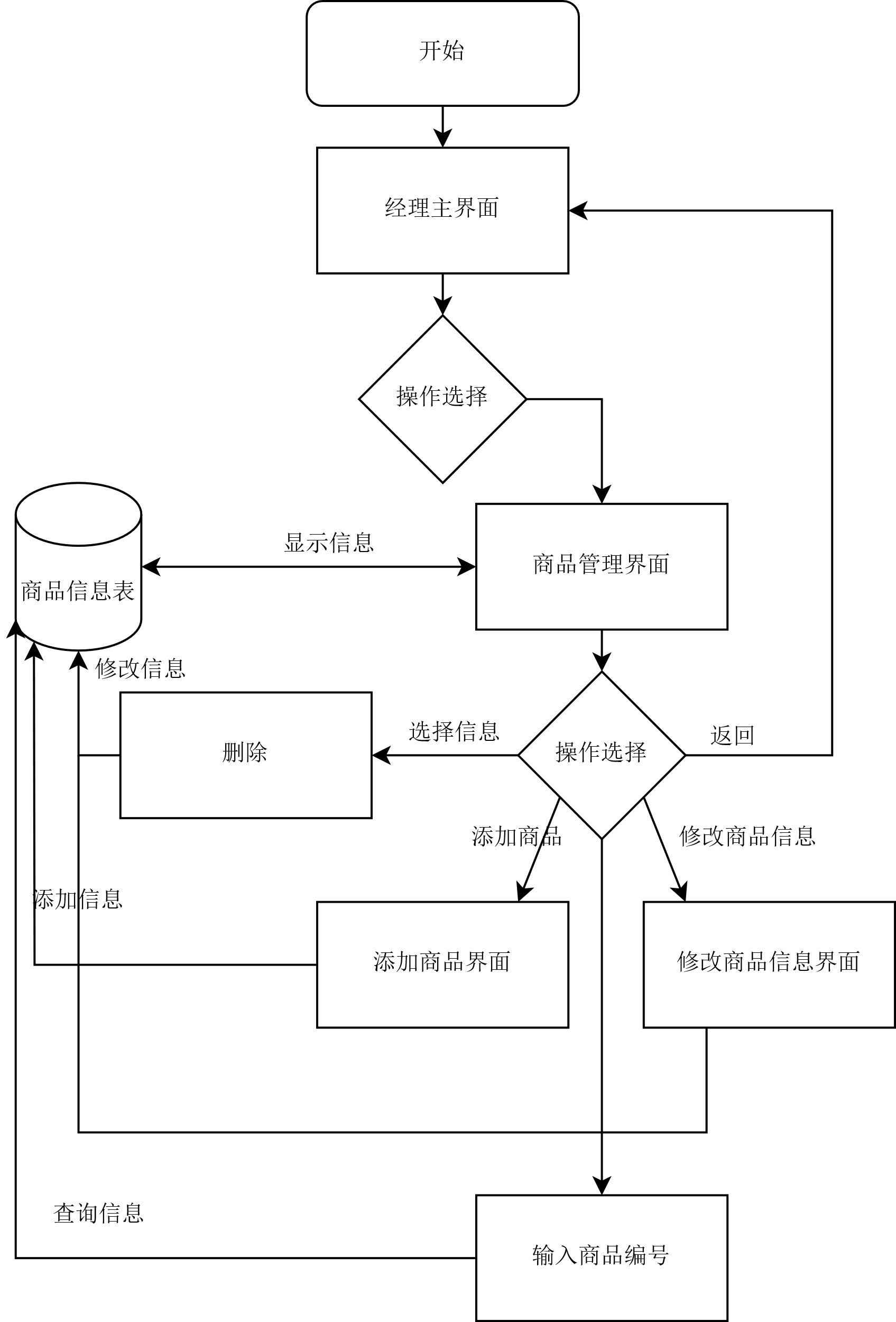
在经理进入主界面后可以选择进入商品管理界面。商品管理主要为商品信息的增删查改，在商品管理主界面可以实现商品信息的删除和查询。增加信息时，会进入增加商品界面，需要输入商品信息。修改商品信息时也会进入相应界面，对信息进行修改。商品信息管理流程如图3-2-5所示。

图3-2-5商品信息管理流程图

### 3.2.6人员信息管理流程设计

在经理进入主界面后可以选择进行人员管理。人员管理分为员工信息管理以及顾客信息管理。两类人员管理类似于商品信息管理，均实现了对该人员信息的增删查改。人员信息管理流程如图3-2-6所示。

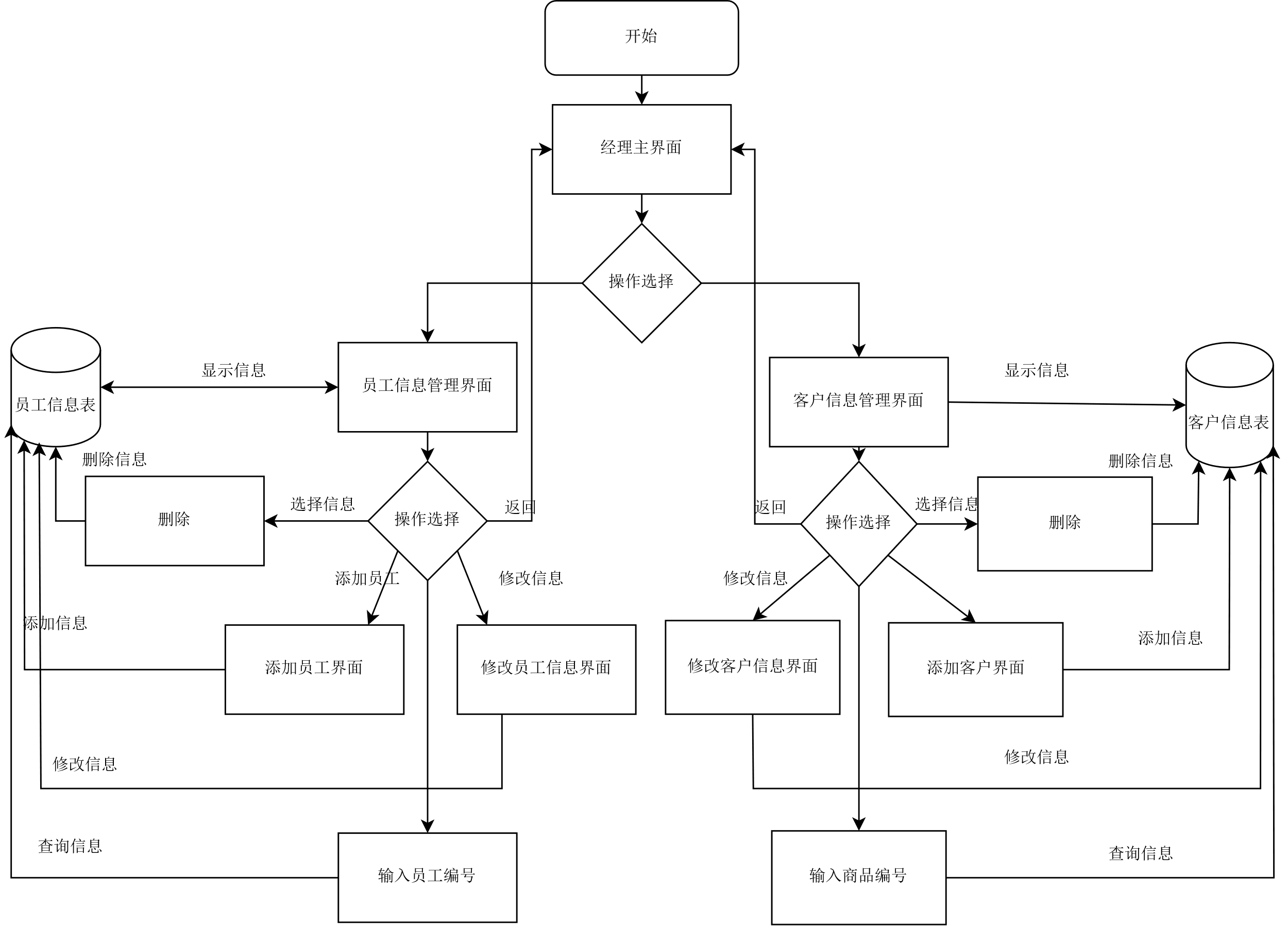


图3-2-6人员信息管理流程图

## 3.3 数据库设计

### 3.3.1概念结构设计

（1）经理信息

经理对应的实体为管理员，具有管理员编号，管理员姓名，性别，年龄，电话，住址，密码等属性，经理实体如图3-3-1 所示。

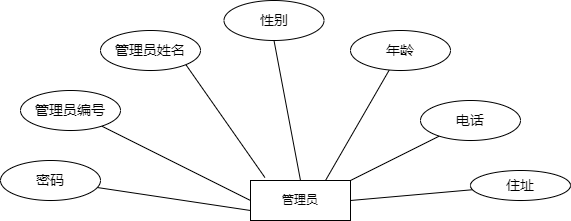


图3-3-1 经理实体属性图

（2）员工信息

员工实体，具有员工编号，员工姓名，性别，年龄，电话，住址，密码等属性，员工实体如图3-3-2 所示。

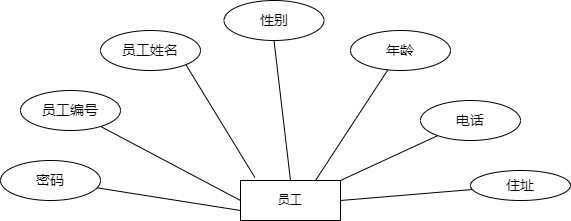


图3-3-2 员工实体属性图

（3）客户信息

客户实体，具有客户编号，客户姓名，性别，电话，VIP，备注等属性，客户实体如图3-3-3-3 所示。

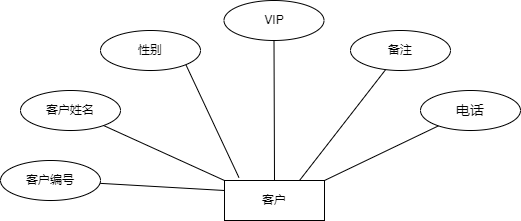


图3-3-3 客户实体属性图

（3）商品信息

商品实体，具有商品编号，商品属性，商品名称，商品单价，商品库存量，生产厂商，保质期，备注等属性，商品实体如图3-3-3-4 所示。

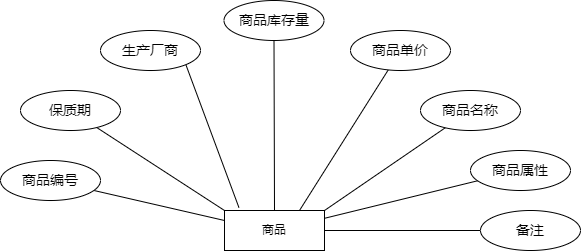


图3-3-4 客户实体属性图

（3）实体联系图

总体实体联系图如图3-3-5 总体实体联系图。

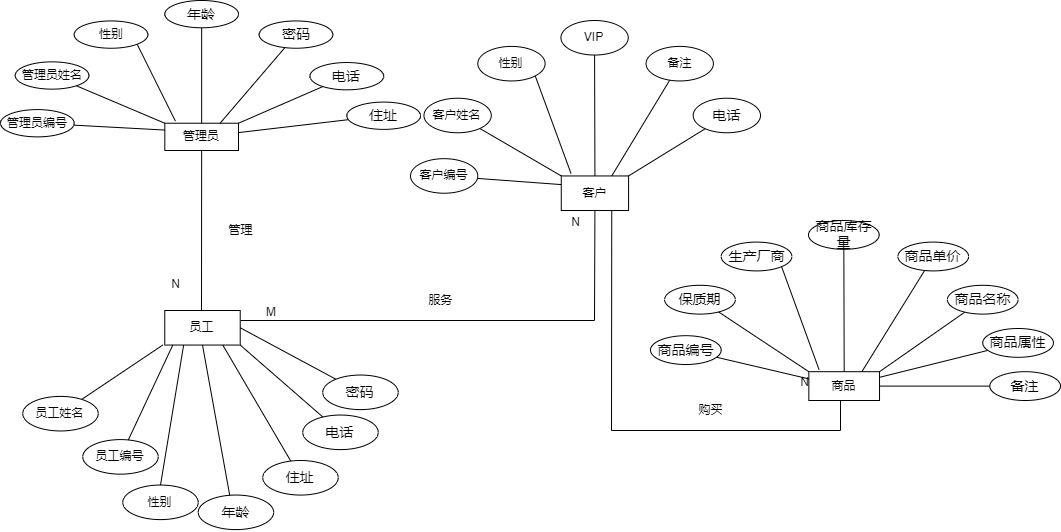


图3-3-5 总体实体联系图

### 3.3.2逻辑结构设计

（1）Admin

Admin表如表3-3-6所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| 管理员编号 | varchar(12) | YES | YES | 编号 |
| 姓名 | varchar(12) | YES | NO | 管理员姓名 |
| 性别 | varchar(4) | YES | NO | 管理员性别 |
| 年龄 | int | YES | NO | 管理员年龄 |
| 住址 | varchar(24) | YES | NO | 管理员住址 |
| 密码 | varchar(20) | YES | NO | 管理员密码 |
| 电话 | varchar(18) | YES | NO | 管理员电话 |

表3-3-6 Admin表结构说明

（2）Customers

Customers表如表3-3-7所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| 客户编号 | varchar(12) | YES | YES | 编号 |
| 姓名 | varchar(12) | YES | NO | 管理员姓名 |
| 性别 | varchar(4) | YES | NO | 管理员性别 |
| VIP | bit | NO | NO | 是否VIP |
| 备注 | Test | NO | NO | 客户信息 |

表3-3-7 Customers表结构说明

（3）Employees

Employees表如表3-3-8所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| 员工编号 | varchar(12) | YES | YES | 编号 |
| 姓名 | varchar(12) | YES | NO | 员工姓名 |
| 性别 | varchar(4) | YES | NO | 员工性别 |
| 年龄 | int | YES | NO | 员工年龄 |
| 住址 | varchar(24) | YES | NO | 员工住址 |
| 密码 | varchar(20) | YES | NO | 员工密码 |
| 电话 | varchar(18) | YES | NO | 员工电话 |
| 职位 | varchar(12) | YES | NO | 员工职位 |

表3-3-8 Employees表结构说明

（3）Goods

Goods表如表3-3-9所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| 商品编号 | varchar(12) | YES | YES | 编号 |
| 商品属性 | varchar(12) | YES | NO | 类别 |
| 商品名称 | varchar(12) | YES | NO | 名称 |
| 商品单价 | int | YES | NO | 价格 |
| 库存量 | int | YES | NO | 存量 |
| 生产厂商 | varchar(24) | YES | NO | 产商 |
| 保质期 | date | YES | NO | 可售卖的日期 |
| 备注 | text | NO | NO | 补充说明 |

表3-3-9 Goods表结构说明

（5）Trade

Trade表如表3-3-10所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 非空 | 主键 | 含义 |
| 流水单号 | varchar(12) | YES | YES | 编号 |
| 客户编号 | varchar(12) | YES | NO | 购买的客户 |
| 商品编号 | varchar(12) | YES | NO | 购买的商品 |
| 商品数量 | int | YES | NO | 单次购买数量 |
| 购买时间 | data | YES | NO | 购买时间 |
| 金额 | int | YES | NO | 单次购买金额 |

表3-3-10 Trade表结构说明

# 4系统详细设计

## 4.1 登录模块详细设计

在WinForm中完成对登录界面的设计。设置2个输入窗口，完成对账号，密码的输入，1个身份选择按钮，1个退出系统的按钮。在点击确认后鉴于身份建立一个与数据库的连接，通过输入的账号比对账号的密码是否正确，正确的话关闭登录界面，进入相应身份的主界面，同时传入账号信息。如果登陆失败则提醒账号或密码错误，同时仅保留账号。

## 4.2个人信息管理模块模块详细设计

在员工和经理的主界面都设计一个个人界面的按钮，可以通过点击进入个人信息管理主界面同时通过主界面传入对应使用者的账号。在个人主界面上显示查找到的员工或经理的个人信息，界面上方还有个人信息修改，密码修改以及返回按钮。点击个人信息修改将原来的标签窗口隐藏显示文本输入窗口，密码修改时进入新的界面同时传入账号。

## 4.3商品售出模块详细设计

该模块相对简单，通过时间函数，给输入的信息自动添加时间，同时通过查找读取到流水单号的下一个单号并自动填入。其他流水单的信息设计出输入窗口，以及最后的一个确认按钮，将信息录入数据库中的流水单表中。

## 4.4商品信息管理模块详细设计

该模块简单实现了对商品信息的增删查改功能。删除功能在商品主界面便可以完成，通过C#中的表单组件，可以返回鼠标点中的那一行的商品信息，在对其进行删除操作。选中的信息也可以传入修改界面，会直接显示在商品的各属性后的输入文本框内。在进入商品主界面时新建一个table函数，加载商品信息表，该函数同样可以用于刷新表单。在主界面内的查询框内可以输入对应商品的编号以及商品的名称，名称实现模糊查询。

## 4.5流水信息管理模块详细设计

流水信息管理界面相对更简单一些，实现流水单号的修改以及查询功能。通过一个函数来实现对流水信息表的加载以及显示，同样也便实现了刷新功能。对流水信息可以通过流水单号进行查询。还有点击修改页面后，会传入选中的流水单的信息，同时显示在修改页面内。

## 4.6人员信息管理模块详细设计

该模块简单实现了对人员信息的增删查改功能。包含两类人员员工以及客户。均实现对人员信息的增删查改，只有一些细小的差别。删除功能在商品主界面便可以完成，通过C#中的表单组件，可以返回鼠标点中的那一行的商品信息，可以显示在界面上方，再对其进行删除操作时会有一定的提示，避免误删的情况。选中的信息也可以传入修改界面，会直接显示在商品的各属性后的输入文本框内。在增加人员信息时都会通过对人员编号的查询获取下一位人员应有的编号并直接输入进文本框内，对于员工账号会有多余的密码项，在有提示的同时，也会直接输入默认密码。对于员工的查询可以通过账号即员工编号，以及员工的姓名，对于客户则通过电话以及姓名查询，姓名查询均支持模糊查询。

# 5 系统实现

## 5.1 登录模块实现

进入系统后首先会显示的便是登录界面。输入账号以及密码，选择身份以登录。

登录实现界面如图4-1所示。



图4-1 登录功能实现界面

登录功能的核心代码如下：

public partial class login1 : Form

{

public login1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "")

{

login();

}

else

{

MessageBox.Show("账号密码均不能为空！");

}

}

public void login()

{

if (radioButtonUser.Checked == true)

{

Link da= new Link();

string sql = "select \* from Employees where 员工编号='" + textBox1.Text+"' and 密码='"+textBox2.Text+"'";

IDataReader dc = da.read(sql);

if (dc.Read())

{

data.UID = dc["员工编号"].ToString();

data.UNAME = dc["姓名"].ToString();

employee1 em=new employee1(textBox1.Text);

this.Hide();

em.ShowDialog();

this.textBox2.Clear();

this.Show();

}

else

{

MessageBox.Show("账号或密码错误，登录失败！");

}

}

if(radioButtonAdmin.Checked == true)

{

Link da = new Link();

string sql = "select \* from Admin where 管理员编号='" + textBox1.Text + "' and 密码='" + textBox2.Text + "'";

IDataReader dc = da.read(sql);

if (dc.Read())

{

admin em = new admin(textBox1.Text);

this.Hide();

em.ShowDialog();

this.textBox2.Clear();

this.Show();

}

else

{

MessageBox.Show("账号或密码错误，登录失败");

}

da.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

## 5.2 个人信息管理模块实现

员工的个人信息界面如图4-1-1所示。



图4-2-1 个人信息功能实现界面

管理员的个人信息界面如图4-1-1所示。



图4-2-1 管理员信息功能实现界面

个人信息管理功能的核心代码如下：

public partial class employee1 : Form

{

string ID;

public employee1()

{

InitializeComponent();

}

public employee1(string id)

{

InitializeComponent();

ID = id;

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

admin客主 admin = new admin客主();

this.Hide();

admin.ShowDialog();

this.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

employee11 em = new employee11();

this.Hide();

em.ShowDialog();

this.Show();

}

private void employee1\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

employee个人 em = new employee个人(ID);

this.Hide();

em.ShowDialog();

this.Show();

}

}

## 5.3 商品售出管理模块实现

商品售出界面如图4-3所示。

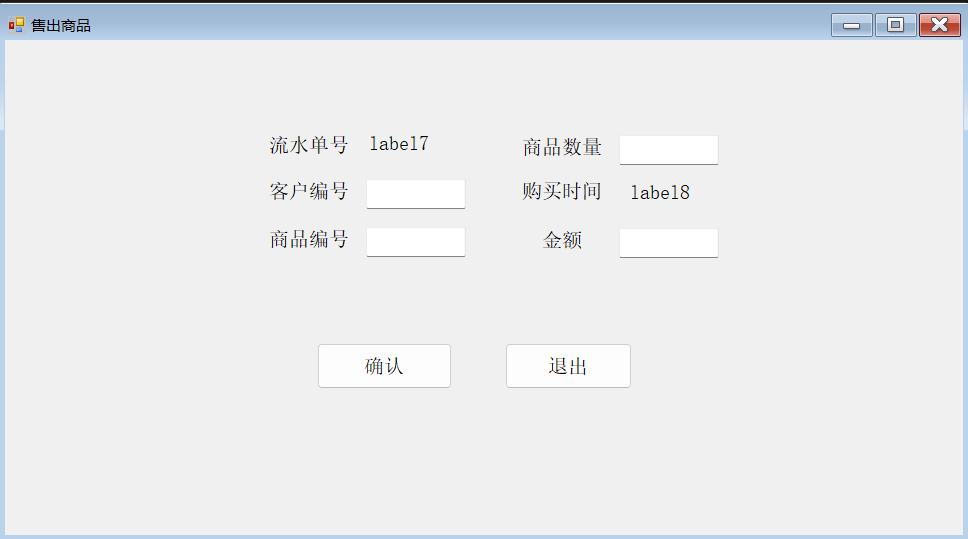


图4-3 商品售出功能实现界面

商品售出功能的核心代码如下：

public partial class employee11 : Form

{

public employee11()

{

InitializeComponent();

count();

}

private void count()

{

Link da = new Link();

string sql = $"select max(流水单号)+1 as x from Trade";

IDataReader dc = da.read(sql);

if (dc.Read())

{

label7.Text = dc["x"].ToString();

}

label8.Text=DateTime.Now.ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

if (label7.Text != "" && textBox2.Text != "" && textBox3.Text != "" && textBox4.Text != "" && textBox6.Text != "")

{

Link da = new Link();

string sql = $"insert into Trade values('{label7.Text}','{textBox2.Text}','{textBox3.Text}'," +

$"{textBox4.Text},'{label8.Text}',{textBox6.Text})";

int n = da.Excute(sql);

if (n > 0)

{

MessageBox.Show("添加成功");

Link da2 = new Link();

string sql2 = $"update Goods set 库存量=库存量-'{textBox4.Text}' where 商品编号='{textBox3.Text}'";

int n2 = da2.Excute(sql2);

count();

textBox2.Text = textBox3.Text = textBox4.Text = textBox6.Text = "";

}

else

{

MessageBox.Show("添加失败");

}

}

else

{

MessageBox.Show("有非必要空值");

}

}

private void employee11\_Load(object sender, EventArgs e)

{

count();

}

## 5.4商品信息管理模块详细设计

商品信息管理界面如图4-4所示。

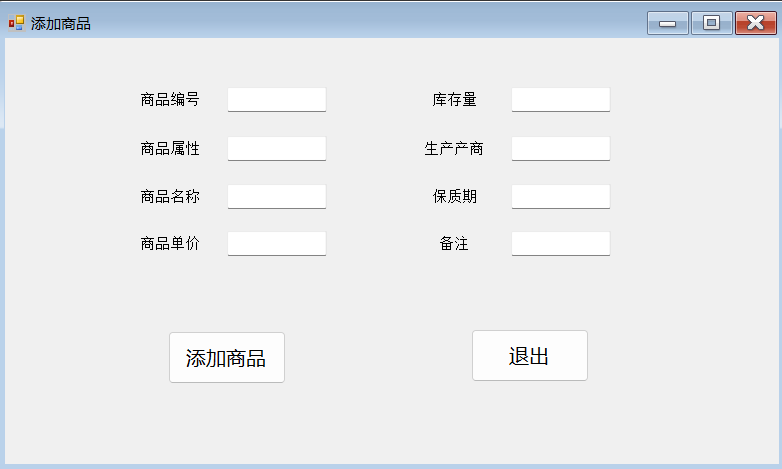




图4-4 商品信息管理功能实现界面

商品信息管理功能的核心代码如下：

public partial class admin商主 : Form

{

public admin商主()

{

InitializeComponent();

Table();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void admin2\_Load(object sender, EventArgs e)

{

string id = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

label2.Text = id + dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

}

public void Table()//读取数据

{

dataGridView1.Rows.Clear();

Link da=new Link();

string sql = "select \* from Goods";

IDataReader dc = da.read(sql);

string a0, a1, a2, a3, a4, a5, a6, a7;

while (dc.Read())

{

a0 = dc[0].ToString();

a1 = dc[1].ToString();

a2 = dc[2].ToString();

a3 = dc[3].ToString();

a4 = dc[4].ToString();

a5 = dc[5].ToString();

a6 = dc[6].ToString();

a6=a6.Remove(9);

a7 = dc[7].ToString();

string[] table = { a0, a1, a2, a3, a4, a5, a6, a7 };

dataGridView1.Rows.Add(table);

}

dc.Close();

da.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

admin商添 admin = new admin商添();

this.Hide();

admin.ShowDialog();

this.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

try

{

string id = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

label2.Text = id + dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

DialogResult dr = MessageBox.Show("确认删除？", "信息提示", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);

if (dr == DialogResult.OK)

{

string sql = $"delete from Goods where 商品编号='{id}'";

Link da = new Link();

if (da.Excute(sql)>0)

{

MessageBox.Show("删除成功");

Table();

}

else

{

MessageBox.Show("删除失败"+sql);

}

da.Close();

}

}

catch

{

MessageBox.Show("选择要删除的商品", "信息提示", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void dataGridView1\_Click(object sender, EventArgs e)

{

string id = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

label2.Text = id + dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

}

private void button7\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

string a1 = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

string a2 = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

string a3 = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

string a4 = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

string a5 = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

string a6 = dataGridView1.SelectedRows[0].Cells[5].Value.ToString();

string a7 = dataGridView1.SelectedRows[0].Cells[6].Value.ToString();

string a8 = dataGridView1.SelectedRows[0].Cells[7].Value.ToString();

admin商改 admin=new admin商改(a1,a2,a3,a4,a5,a6,a7,a8);

this.Hide();

admin.ShowDialog();

this.Show();

Table();

}

catch

{

MessageBox.Show("Error");

}

}

private void button7\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

Table();

textBox1.Text = "";

textBox2.Text = "";

}

private void button5\_Click(object sender, EventArgs e)

{

TableId();

}

public void TableId()

{

dataGridView1.Rows.Clear();

Link da = new Link();

string sql = $"select \* from Goods where 商品编号 ='{textBox1.Text}'";

IDataReader dc = da.read(sql);

string a0, a1, a2, a3, a4, a5, a6, a7;

while (dc.Read())

{

a0 = dc[0].ToString();

a1 = dc[1].ToString();

a2 = dc[2].ToString();

a3 = dc[3].ToString();

a4 = dc[4].ToString();

a5 = dc[5].ToString();

a6 = dc[6].ToString();

a6 = a6.Remove(9);

a7 = dc[7].ToString();

string[] table = { a0, a1, a2, a3, a4, a5, a6, a7 };

dataGridView1.Rows.Add(table);

}

dc.Close();

da.Close();

}

public void TableNa()

{

dataGridView1.Rows.Clear();

Link da = new Link();

string sql = $"select \* from Goods where 商品名称 like '%{textBox2.Text}%'";

IDataReader dc = da.read(sql);

string a0, a1, a2, a3, a4, a5, a6, a7;

while (dc.Read())

{

a0 = dc[0].ToString();

a1 = dc[1].ToString();

a2 = dc[2].ToString();

a3 = dc[3].ToString();

a4 = dc[4].ToString();

a5 = dc[5].ToString();

a6 = dc[6].ToString();

a6 = a6.Remove(9);

a7 = dc[7].ToString();

string[] table = { a0, a1, a2, a3, a4, a5, a6, a7 };

dataGridView1.Rows.Add(table);

}

dc.Close();

da.Close();

}

private void button6\_Click\_1(object sender, EventArgs e)

{

TableNa();

}

## 5.5流水信息管理模块详细设计

流水信息管理界面如图4-5所示。

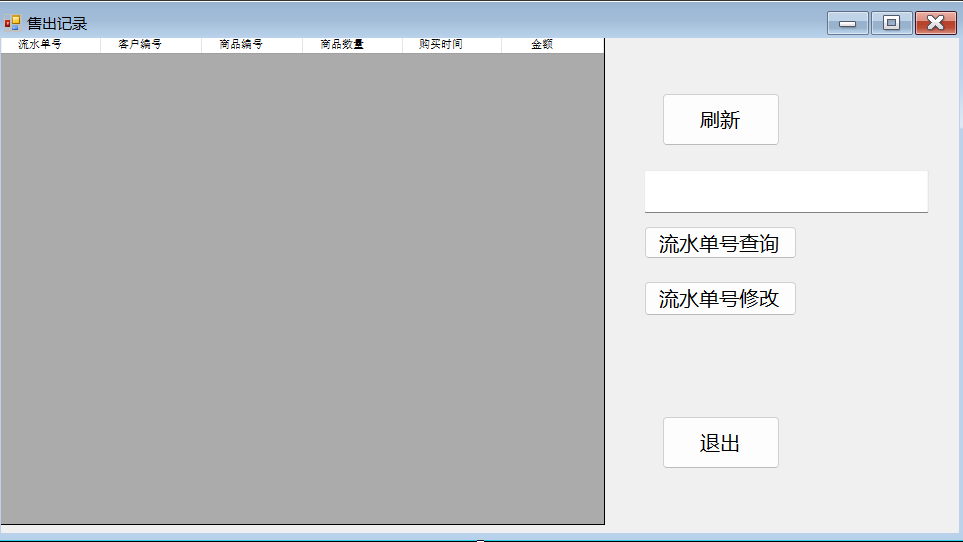




图4-5 流水信息管理功能实现界面

流水信息管理功能的核心代码如下：

public partial class admin单主 : Form

{

public admin单主()

{

InitializeComponent();

Table();

}

public void Table()//读取数据

{

dataGridView1.Rows.Clear();

Link da = new Link();

string sql = $"select \* from Trade";

IDataReader dc = da.read(sql);

string a0, a1, a2, a3, a4, a5;

while (dc.Read())

{

a0 = dc[0].ToString();

a1 = dc[1].ToString();

a2 = dc[2].ToString();

a3 = dc[3].ToString();

a4 = dc[4].ToString();

a4 = a4.Remove(9);

a5 = dc[5].ToString();

string[] table = { a0, a1, a2, a3, a4, a5};

dataGridView1.Rows.Add(table);

}

dc.Close();

da.Close();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button7\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

Table();

}

private void button5\_Click(object sender, EventArgs e)

{

TableId();

}

public void TableId()

{

dataGridView1.Rows.Clear();

Link da = new Link();

string sql = $"select \* from Trade where 流水单号 ='{textBox1.Text}'";

IDataReader dc = da.read(sql);

string a0, a1, a2, a3, a4, a5;

while (dc.Read())

{

a0 = dc[0].ToString();

a1 = dc[1].ToString();

a2 = dc[2].ToString();

a3 = dc[3].ToString();

a4 = dc[4].ToString();

a4 = a4.Remove(9);

a5 = dc[5].ToString();

string[] table = { a0, a1, a2, a3, a4, a5 };

dataGridView1.Rows.Add(table);

}

dc.Close();

da.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

string a1 = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

string a2 = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

string a3 = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

string a4 = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

string a5 = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

string a6 = dataGridView1.SelectedRows[0].Cells[5].Value.ToString();

admin单改 admin = new admin单改(a1, a2, a3, a4, a5, a6);

this.Hide();

admin.ShowDialog();

this.Show();

Table();

}

catch

{

MessageBox.Show("Error");

}

}

private void admin单主\_Load(object sender, EventArgs e)

{

}

## 5.6人员信息管理模块详细设计

人员信息管理界面如图4-6所示。



图4-6 人员信息管理功能实现界面

人员信息管理功能的核心代码如下：

public partial class admin员主 : Form

{

public admin员主()

{

InitializeComponent();

Table();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void admin3\_Load(object sender, EventArgs e)

{

string id = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

label2.Text = id + dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

}

public void Table()//读取数据

{

dataGridView1.Rows.Clear();

Link da = new Link();

string sql = "select \* from Employees";

IDataReader dc = da.read(sql);

string a0, a1, a2, a3, a4, a5, a6, a7;

while (dc.Read())

{

a0 = dc[0].ToString();

a1 = dc[1].ToString();

a2 = dc[2].ToString();

a3 = dc[3].ToString();

a4 = dc[4].ToString();

a5 = dc[5].ToString();

a6 = dc[6].ToString();

a7 = dc[7].ToString();

string[] table = { a0, a1, a2, a3, a4, a5, a6, a7 };

dataGridView1.Rows.Add(table);

}

dc.Close();

da.Close();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void dataGridView1\_Click(object sender, EventArgs e)

{

string id = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

label2.Text = id + dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

}

private void button7\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

string a1 = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

string a2 = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

string a3 = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

string a4 = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

string a5 = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

string a6 = dataGridView1.SelectedRows[0].Cells[5].Value.ToString();

string a7 = dataGridView1.SelectedRows[0].Cells[6].Value.ToString();

string a8 = dataGridView1.SelectedRows[0].Cells[7].Value.ToString();

admin商改 admin = new admin商改(a1, a2, a3, a4, a5, a6, a7, a8);

this.Hide();

admin.ShowDialog();

this.Show();

Table();

}

catch

{

MessageBox.Show("Error");

}

}

private void button7\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

public void TableId()

{

dataGridView1.Rows.Clear();

Link da = new Link();

string sql = $"select \* from Employees where 员工编号 ='{textBox1.Text}'";

IDataReader dc = da.read(sql);

string a0, a1, a2, a3, a4, a5, a6, a7;

while (dc.Read())

{

a0 = dc[0].ToString();

a1 = dc[1].ToString();

a2 = dc[2].ToString();

a3 = dc[3].ToString();

a4 = dc[4].ToString();

a5 = dc[5].ToString();

a6 = dc[6].ToString();

a7 = dc[7].ToString();

string[] table = { a0, a1, a2, a3, a4, a5, a6, a7 };

dataGridView1.Rows.Add(table);

}

dc.Close();

da.Close();

}

public void TableNa()

{

dataGridView1.Rows.Clear();

Link da = new Link();

string sql = $"select \* from Employees where 姓名 like '%{textBox2.Text}%'";

IDataReader dc = da.read(sql);

string a0, a1, a2, a3, a4, a5, a6, a7;

while (dc.Read())

{

a0 = dc[0].ToString();

a1 = dc[1].ToString();

a2 = dc[2].ToString();

a3 = dc[3].ToString();

a4 = dc[4].ToString();

a5 = dc[5].ToString();

a6 = dc[6].ToString();

a7 = dc[7].ToString();

string[] table = { a0, a1, a2, a3, a4, a5, a6, a7 };

dataGridView1.Rows.Add(table);

}

dc.Close();

da.Close();

}

private void button6\_Click(object sender, EventArgs e)

{

TableNa();

}

private void button5\_Click\_1(object sender, EventArgs e)

{

TableId();

}

private void button3\_Click(object sender, EventArgs e)

{

try

{

string id = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

label2.Text = id + dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

DialogResult dr = MessageBox.Show("确认删除？", "信息提示", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);

if (dr == DialogResult.OK)

{

string sql = $"delete from Employees where 员工编号='{id}'";

Link da = new Link();

if (da.Excute(sql) > 0)

{

MessageBox.Show("删除成功");

Table();

}

else

{

MessageBox.Show("删除失败" );

}

da.Close();

}

}

catch

{

MessageBox.Show("选择要删除的员工", "信息提示", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button7\_Click\_2(object sender, EventArgs e)

{

this.Close();

}

private void button4\_Click\_1(object sender, EventArgs e)

{

Table();

textBox1.Text = "";

textBox2.Text = "";

}

private void button1\_Click\_1(object sender, EventArgs e)

{

admin员添 admin=new admin员添();

this.Hide();

admin.ShowDialog();

this.Show();

Table();

}

private void button2\_Click\_1(object sender, EventArgs e)

{

try

{

string a1 = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

string a2 = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

string a3 = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

string a4 = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();

string a5 = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();

string a6 = dataGridView1.SelectedRows[0].Cells[5].Value.ToString();

string a7 = dataGridView1.SelectedRows[0].Cells[6].Value.ToString();

string a8 = dataGridView1.SelectedRows[0].Cells[7].Value.ToString();

admin员改 admin = new admin员改(a1, a2, a3, a4, a5, a6, a7, a8);

this.Hide();

admin.ShowDialog();

this.Show();

Table();

}

catch

{

MessageBox.Show("Error");

}

}

private void label2\_Click\_1(object sender, EventArgs e)

{

}

private void dataGridView1\_CellContentClick\_1(object sender, DataGridViewCellEventArgs e)

{

string id = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

label2.Text = id + dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

}

# 6 总结

在这次课程设计中，我们第一次使用C#平台来数据库管理系统的设计和实现，锻炼了我们对类的使用，以及编程的能力；通过对数据关系的设计，我们对于数据库有了更深的理解，对数据库的操作更加熟练。

本次数据库的课程设计我们设计并实现了小型超市商品管理系统。我们主要实现了对员工、客户的信息管理，对商品、流水单信息的管理，以及经理，员工的登录以及其个人信息的管理。

对于这次的课程设计，我们自认为界面还不够简洁，美观，代码的设计也不够简明、精炼。还有部分关系不够契合实际，还可以进一步进行优化，对商品的流水进行进一步的统计。对于商品信息的统计还不够详细以及对流水处理功能还可以进一步完善。