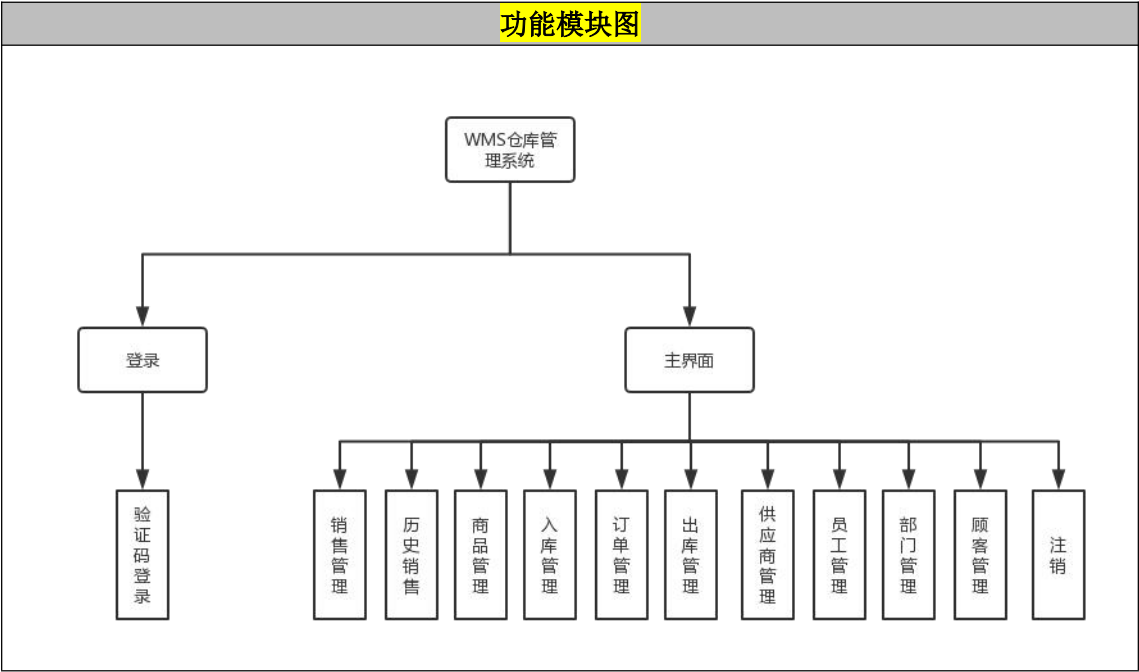


目录

一、 功能说明及图示	1
二、 界面图示及输入，处理和输出的说明	1
三、 自编的代码和说明	8
1.项目结构	8
2. 关键代码	9
2.1 wms\Common\NotifyBase.cs	9
2.2 wms\Common\CommandBase.cs	9
2.3 wms\Converter\BoolToIntConverter.cs	10
2.4 wms\DataAccess\ServerDataAccess.cs	11
2.5 wms\Model\	35
2.6 wms\View\	47
2.7 wms\ViewModel\	108
四、 包含数据的界面截图以及调试过程	143

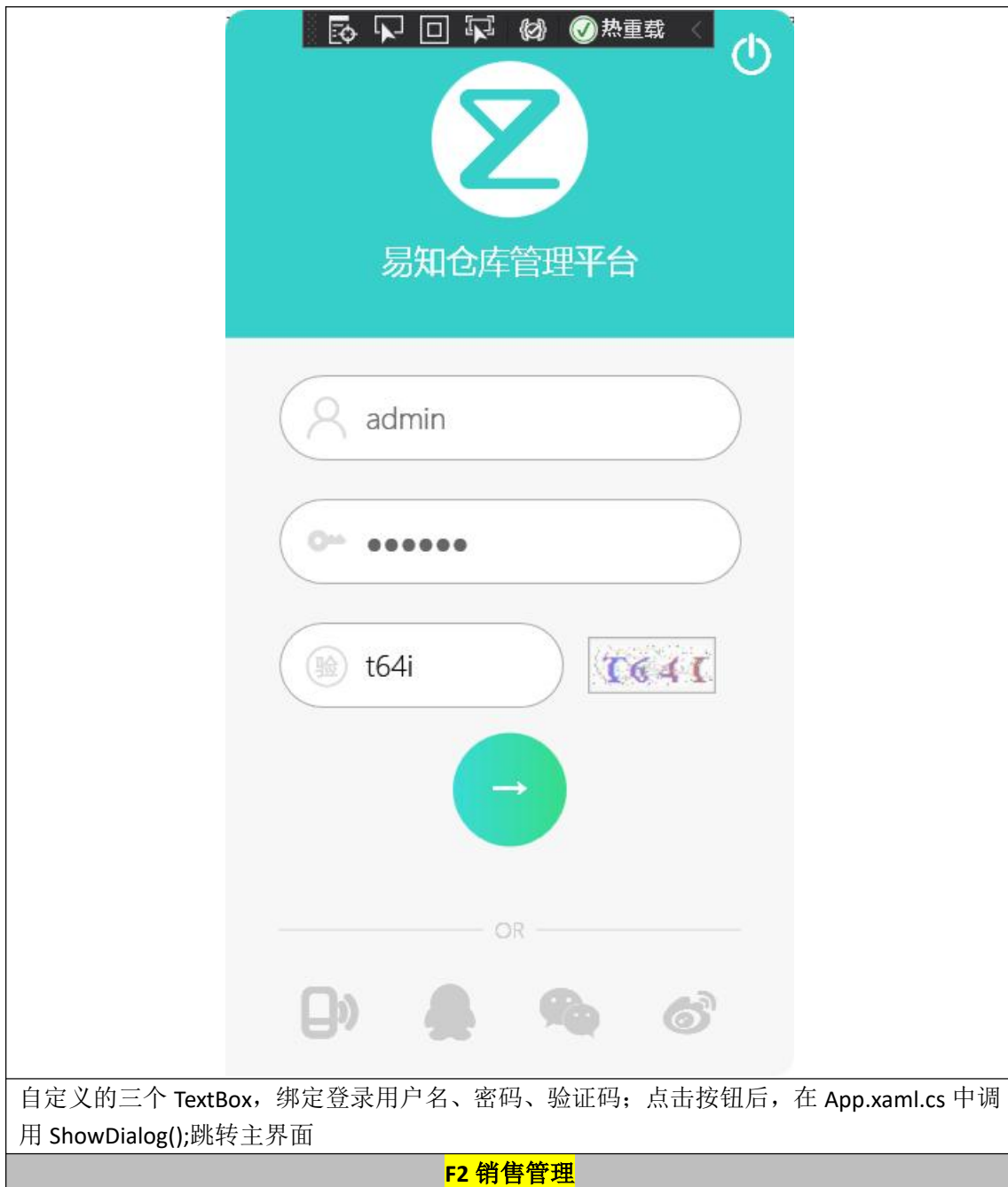
一、功能说明及图示

功能序号	功能	描述
F1	登录	管理员身份与非管理员身份进行登录；图片验证码；
F2	销售管理	查询所有销售记录
F3	历史销售	按月按年查询历史销售
F4	商品管理	货物分级代码存储；存量的多角度查询；
F5	入库管理	货物入库信息的添加、修改和删除；入库信息的多角度查询；
F6	订单管理	货物订单信息的添加、修改和删除；订单信息的多角度查询；
F7	出库管理	货物出库信息的添加、修改和删除； 出库信息的多角度查询；
F8	供应商管理	供应商改名问题（方案三）
F9	顾客管理	客户信息查询，客户地址更改（类供应商改名问题方案三）
F10	员工管理	员工信息的管理
F11	部门管理	部门信息的管理
F12	注销	注销当前账户，返回登录界面
...



二、界面图示及输入，处理和输出的说明

F1 登录



易知仓库管理平台							
admin							
销售管理	历史销售	商品管理	入库管理	订单管理	出库管理	供应商管理	顾客管理
员工管理	部门管理						
单据号	日期	用户Id	商品	商品名	价格	数量	总计
OR1000000001	4/10/2021 12:15:07 AM	C00001	4	火腿肠	34.0000	20	680.0000
OR1000000001	4/10/2021 12:15:07 AM	C00001	5	面包	48.0000	190	9120.0000
OR1000000001	4/10/2021 12:15:07 AM	C00001	7	薯片	41.0000	150	6150.0000
OR1000000001	4/10/2021 12:15:07 AM	C00001	8	罐头	37.0000	150	5550.0000
OR1000000002	3/10/2021 12:15:07 AM	C00002	4	火腿肠	34.0000	30	1020.0000

将数据库获取的数据，转化为 List<Model>，通过 textblock 展示，查询销售详情，全局悬浮按钮绑定刷新功能，即重新载入顶部 NavRadioButton 对应的子用户控件

F3 历史销售

易知仓库管理平台							
admin							
销售管理	历史销售	商品管理	入库管理	订单管理	出库管理	供应商管理	顾客管理
员工管理	部门管理						
年份	月份	商品	商品名	月销售量	月销售额	年销售量	年销售额
2021	3	4	火腿肠	30	1020.0000	30	1020.0000
2021	4	4	火腿肠	20	680.0000	50	1700.0000
2021	4	5	面包	190	9120.0000	190	9120.0000
2021	4	7	薯片	150	6150.0000	150	6150.0000
2021	4	8	罐头	150	5550.0000	150	5550.0000

将数据库获取的数据，转化为 List<Model>，通过 textblock 展示，查询历史销售详情（月销售量和年销售量），全局悬浮按钮绑定刷新功能，即重新载入顶部 NavRadioButton 对应的子用户控件

F4 商品管理



商品分级代码的左侧是自定义的 `datagrid`，右侧是数据库数据转化为 `List<Model>` 绑定到 `ScrollView` 里的 `TextBlock`，保存按钮会将 `dataset.tables[0]` 里的内容更新到数据库

F5 入库管理



入库管理是两个 `datatable` 用于展示，左边的 `in` 表各列绑定到各个控件，`combobox` 用到了 `DisplayMemberPath` 和 `DisplayMemberPath` 来通过供应商 `id` 显示供应商名称，`datetimepicker` 引入了 `github` 上开源的控件，状态里的 `RadioButton` 用到了 `BoolToIntConverter` 类在 `style` 定义使用，将 `int` 里的 `0` 和 `1` 与 `bool` 进行转换；右边的 `document_info` 里的各列绑定到 `datagrid` 下排的按钮可实现翻页和增删等操作；

F6 订单管理

易知仓库管理平台

admin
➔

销售管理
历史销售
商品管理
入库管理
订单管理
出库管理
供应商管理
顾客管理
员工管理
部门管理

订单信息

订单详情

订单ID

顾客ID

顾客 刘博文

顾客信息版本号

订单时间

状态 ☒ 已完成 ☐ 未完成

商品id	单号	数量	价格	备注
4	OR1000000001	20	34.0000	
5	OR1000000001	190	48.0000	
7	OR1000000001	150	41.0000	
8	OR1000000001	150	37.0000	

|<
<
>
>|
新增
删除
保存

订单管理是两个 `datatable` 用于展示，左边的 `order` 表各列绑定到各个控件，`combobox` 用到了 `DisplayMemberPath` 和 `DisplayMemberPath` 来通过顾客 id 显示顾客名称，`datetimepicker` 引入了 `github` 上开源的控件，状态里的 `RadioButton` 用到了 `BoolToIntConverter` 类在 `style` 定义使用，将 `int` 里的 0 和 1 与 `bool` 进行转换；右边的 `document_info` 里的各列绑定到 `datagrid` 下排的按钮可实现翻页和增删等操作；

F7 出库管理

易知仓库管理平台

admin
➔

销售管理
历史销售
商品管理
入库管理
订单管理
出库管理
供应商管理
顾客管理
员工管理
部门管理

出库单信息

出库单详情

出库单ID

顾客ID

顾客 刘博文

顾客信息版本号

出库单时间

状态 ☒ 已完成 ☐ 未完成

商品id	单号	数量	价格	备注
4	OU1000000001	20	34.0000	
5	OU1000000001	190	48.0000	
7	OU1000000001	150	41.0000	
8	OU1000000001	150	37.0000	

|<
<
>
>|
新增
删除
保存

出库管理是两个 `datatable` 用于展示，左边的 `out` 表各列绑定到各个控件，`combobox` 用到了

DisplayMemberPath 和 DisplayMemberPath 来通过顾客 id 显示顾客名称, datatimepicker 引入了 github 上开源的控件, 状态里的 RadioButton 用到了 BoolToIntConverter 类在 style 定义使用, 将 int 里的 0 和 1 与 bool 进行转换; 右边的 document_info 里的各列绑定到 datagrid 下排的按钮可实现翻页和增删等操作;

F8 供应商管理

易知仓库管理平台

销售管理 历史销售 商品管理 入库管理 订单管理 出库管理 供应商管理 顾客管理 员工管理 部门管理

供应商信息

供应商ID	供应商信息版本号	信息变更时间	名称	地址	联系电话
S00001	01	5/7/2019 10:08:25 PM	保洁日用	哈尔滨道里湘江路国庆大厦36号	18948566268
S00001	02	5/8/2020 10:08:25 PM	保洁日化	哈尔滨道里湘江路国庆大厦40号	18948566268
S00001	03	5/9/2021 10:08:25 PM	保洁	哈尔滨道里湘江路国庆大厦41号	18948566200
S00002	01	5/9/2019 10:08:25 PM	捷成食品	黑龙江伊春	13674583486
S00003	01	5/9/2019 10:08:25 PM	洗发香水	广州的新路36号	18904588666
S00004	01	5/9/2019 10:08:25 PM	广州佳美	山东省烟台市新桥大厦6号	13674589874
S00005	01	5/9/2019 10:08:25 PM	姐妹洗发	上海	18804588666
S00006	01	5/9/2019 10:08:25 PM	宝洁日化	广州	14785236921
S00007	01	5/9/2019 10:08:25 PM	季吉乐	深圳市建设镇568号	18475569555

保存

用一个 datatable 来进行供应商信息的单表查询与操作

F9 顾客管理

易知仓库管理平台

销售管理 历史销售 商品管理 入库管理 订单管理 出库管理 供应商管理 顾客管理 员工管理 部门管理

顾客信息

顾客ID	顾客信息版本号	信息变更时间	名称	地址	联系电话
C00001	01	5/7/2019 8:16:38 PM	刘岩	上海市浦东新区金海路2360号	18018594399
C00001	02	5/8/2020 8:16:39 PM	刘岩	上海市浦东新区金海路10号	18018594399
C00001	03	5/9/2021 8:16:40 PM	刘博文	上海市浦东新区金海路2360号	18018593333
C00002	01	5/9/2019 8:16:40 PM	韩黄	河南省鹤壁市淇滨区而能物路想辰春社区14号	13718195424
C00003	01	5/9/2019 8:16:40 PM	冯昇院	河南省许昌市魏都区知而路形植社区92号	18273294375
C00004	01	5/9/2019 8:16:40 PM	张级	河南省周口市鹿邑县所者身路恒流社区40号	15699075013
C00005	01	5/9/2019 8:16:40 PM	郑角调	河南省驻马店市上蔡县其在路远舒社区52号	17272152878
C00006	01	5/9/2019 8:16:40 PM	郑始通	河南省开封市禹王台区而后路而归社区8号	17516062617
C00007	01	5/9/2019 8:16:40 PM	尤演	河南省安阳市林州市之知路其出社区87号	13492128223

保存

用一个 datatable 来进行顾客信息的单表查询与操作

F10 员工管理

易知仓库管理平台

admin

→

销售管理

历史销售

商品管理

入库管理

订单管理

出库管理

供应商管理

顾客管理

员工管理

部门管理

员工信息

员工ID

E00001

部门名称

总经理室

员工姓名

刘博文

员工电话

(010)-11111

员工性别

☒ 男
 ☐ 女

员工学历

研究生

员工地址

北京市海淀区

员工邮箱

15881464660@qq.com

员工Id

登录用户名

密码

头像

E00001	admin	51A70A1E25F9E6A8954F54D6DF935B0D	https://z3.ax1x.com/2021/05/21/...

<

>

<

>

新增

删除

保存

员工管理是两个 `datatable` 用于展示，左边的 `employee` 表各列绑定到各个控件，`combobox` 用到了 `DisplayMemberPath` 和 `DisplayMemberPath` 来通过部门 id 显示部门名称，`datepicker` 引入了 `github` 上开源的控件，性别里的 `RadioButton` 用到了 `BoolToIntConverter` 类在 `style` 定义使用，将 `int` 里的 0 和 1 与 `bool` 进行转换；右边的 `login` 里的各列绑定到 `datagrid` 下排的按钮可实现翻页和增删等操作；

F11 部门管理

易知仓库管理平台

admin

→

销售管理

历史销售

商品管理

入库管理

订单管理

出库管理

供应商管理

顾客管理

员工管理

部门管理

部门信息

departid	departname
D00001	总经理室
D00002	监督室
D00003	管理部
D00004	会计科
D00005	业务部
D00006	生管科
D00007	品保科
D00008	制造部
D00009	压合科
D00010	仓库科

保存

用一个 `datatable` 来进行部门信息的单表查询与操作

7

F12 注销	
	
点击注销按钮通过绑定命令的方式使得 DialogResult 值得改变，并在 App.xaml.cs 的 OnStart up 函数中通过 DialogResult 值的判断，来决定打开登录界面或者主界面	

三、自编的代码和说明

1.项目结构

项目结构	
	
项目分为 wms 项目和 wms.Controls 用户控件子项目，其中 wms 中 Assets 用来存放资源文件，Common 存放工具类，Converter 存放数据转换类，DataAccess 里定义数据库操作的各种方法，Model、View、ViewModel 是组成 MVVM 的部分，App.config 存放配置（数据库连接字符串等），App.xaml 定义启动页和跳转、关闭逻辑等，此外 wms.Controls 中用来定义一些第三方控件并在 wms 中引入使用	

2. 关键代码

2.1 wms\Common\NotifyBase.cs

把 Model 需要 INotifyPropertyChanged 接口和 PropertyChanged 属性写成一个 Base 类

```
using System.ComponentModel;
using System.Runtime.CompilerServices;

namespace wms.Common
{
    /*把 Model 需要 INotifyPropertyChanged 接口和 PropertyChanged 属性写成一个 Base 类*/
    public class NotifyBase : INotifyPropertyChanged
    {
        /// <summary>
        /// 属性改变时，调用该方法发出通知
        /// </summary>
        public event PropertyChangedEventHandler PropertyChanged;

        public void DoNotify([CallerMemberName] string propName = "")
        {
            //NULL 检查运算符(?.) 判断一下这个委托是不是为 null；如果是则不执行委托，如
            果不是则执行该委托；
            PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propName));
        }
    }
}
```

2.2 wms\Common\CommandBase.cs

委托命令类

```
using System;
using System.Windows.Input;

namespace wms.Common
{
    public class CommandBase : ICommand
    {
        public Action<object> DoExecute { get; set; }
        //参数为 object 无返回的委托，对应接口中的 Execute
        public Func<object, bool> DoCanExecute { get; set; }
    }
}
```

```

//参数为 object，返回 bool 型的委托，对应接口中 CanExecute
//上面两个委托在实例化对象时根据需要实例化（属性赋值）
public event EventHandler CanExecuteChanged; //接口中定义的事件
//下面用实例化时传来的上面两个委托实现接口中定义的方法
//判断是否实例化后直接调用

/// <summary>
/// 命令是否可执行
/// </summary>
/// <param name="parameter"></param>
/// <returns></returns>
public bool CanExecute(object parameter)
{
    //实例化了该委托（该属性赋值）
    //NULL 检查运算符(?.) 判断一下这个委托是不是为 null；如果是则不执行委托，如
    果不是则执行该委托：
    return DoCanExecute?. Invoke(parameter) == true;
}
/// <summary>
/// 执行命令
/// </summary>
/// <param name="parameter"></param>
public void Execute(object parameter)
{
    DoExecute?. Invoke(parameter);
}
/// <summary>
/// 引起允许执行变化事件
/// </summary>
public void RaiseCanExecuteChanged()
{
    //NULL 检查运算符(?.) 判断一下这个委托是不是为 null；如果是则不执行委托，如
    果不是则执行该委托：
    CanExecuteChanged?. Invoke(this, new EventArgs());
}
}
}

```

2.3 wms\Converter\BoolToIntConverter.cs

数据类型转换类

```

using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Data;

namespace wms.Converter
{
    public class BoolToIntConverter : IValueConverter
    {
        public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
        {
            if (value == null || parameter == null)
                return false;
            return value.ToString() == parameter.ToString();
        }

        public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
        {
            return parameter;
        }
    }
}

```

2.4 wms\DataAccess\ServerDataAccess.cs

数据库操作类

```

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.Common;
using System.Data.SqlClient;
using wms.Common;
using wms.DataAccess.DataEntity;
using wms.Model;

```

```

namespace wms.DataAccess
{
    public class ServerDataAccess
    {
        private static ServerDataAccess instance;
        private ServerDataAccess() { }
        public static ServerDataAccess GetInstance()
        {
            //instance 为空就返回新的实例化
            return instance ?? (instance = new ServerDataAccess());
        }

        SqlConnection conn;
        SqlCommand comm;
        SqlDataAdapter adapter;
        SqlCommandBuilder builder;

        private void Dispose()
        {
            if (adapter != null)
            {
                adapter.Dispose();
                adapter = null;
            }
            if (comm != null)
            {
                comm.Dispose();
                comm = null;
            }
            if (conn != null)
            {
                conn.Close();
                conn.Dispose();
                conn = null;
            }
        }

        private bool DBConnection()
        {
            string connStr =
ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
            if (conn == null)
                conn = new SqlConnection(connStr);
            try

```

```

        {
            conn.Open();
            return true;
        }
        catch
        {
            return false;
        }
    }

    public UserEntity CheckUserInfo(string loginName, string loginPassword)
    {
        try
        {
            if (DBConnection())
            {
                string userSql = "select
a.login_name,b.employee_name,a.login_password,a.logo,b.employee_sex,a.is_can_login from
login a,employee b where a.login_name = @login_name and
a.login_password=@login_password and a.is_valid=1 and a.employee_id=b.employee_id";
                adapter = new SqlDataAdapter(userSql, conn);
                adapter.SelectCommand.Parameters.Add(new
SqlParameter("@login_name", SqlDbType.VarChar) { Value = loginName });
                adapter.SelectCommand.Parameters.Add(new
SqlParameter("@login_password", SqlDbType.VarChar) { Value =
MD5Provider.GetMD5String(loginPassword + "@" + loginName) });

                DataTable table = new DataTable();
                int count = adapter.Fill(table);

                if (count <= 0)
                    throw new Exception("用户名或密码不正确！");

                DataRow dr = table.Rows[0];
                if (dr.Field<int>("is_can_login") == 0)
                    throw new Exception("当前用户没有权限使用此平台！");

                UserEntity userInfo = new UserEntity();
                userInfo.LoginName = dr.Field<string>("login_name");
                userInfo.EmployeeName = dr.Field<string>("employee_name");
                userInfo.LoginPassword = dr.Field<string>("login_password");
                userInfo.Logo = dr.Field<string>("logo");
                userInfo.EmployeeSex = dr.Field<int>("employee_sex");
                return userInfo;
            }
        }
    }
}

```

```

        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }

    return null;
}

public List<SalesModel> GetSales()
{
    try
    {
        List<SalesModel> result = new List<SalesModel>();
        if (this.DBConnection())
        {
            string sql = @"select * from sales";
            adapter = new SqlDataAdapter(sql, conn);

            DataTable table = new DataTable();
            int count = adapter.Fill(table);
            if (count > 0)
            {
                SalesModel model = null;
                foreach (DataRow dr in table.AsEnumerable())
                {
                    model = new SalesModel();
                    model.DocumentId = dr.Field<string>("document_id");
                    model.DocumentDate =
dr.Field<DateTime>("document_date");
                    model.CustomerId = dr.Field<string>("customer_id");
                    model.Id = dr.Field<int>("id");
                    model.ProductName = dr.Field<string>("product_name");
                    model.Price = dr.Field<decimal>("price");
                    model.Quantity = dr.Field<int>("quantity");
                    model.Total = dr.Field<decimal>("total");

                    result.Add(model);
                }
            }
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }

    return result;
}

```



```

        }
    }
}
return result;
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    this.Dispose();
}
}
public List<SalesHistoryModel> GetSalesHistory()
{
    try
    {
        List<SalesHistoryModel> result = new List<SalesHistoryModel>();
        if (this.DBConnection())
        {
            string sql = @"if object_id('tempdb..#temp') is not null
drop table #temp;
create table #temp
(
    hyear          int,
    hmon           int,
    id             int,
    product_name   varchar(20),
    monqty         int,
    monsaleroom    decimal(19, 4),
    yearqty        int,
    yearsaleroom   decimal(19, 4)
);
insert into #temp exec SelectYearSales;
select *
from #temp;";

            adapter = new SqlDataAdapter(sql, conn);

            DataTable table = new DataTable();
            int count = adapter.Fill(table);
            if (count > 0)
            {
                SalesHistoryModel model = null;

```

```

        foreach (DataRow dr in table.AsEnumerable())
        {
            model = new SalesHistoryModel();
            model.Hyear = dr.Field<int>("hyear");
            model.Hmon = dr.Field<int>("hmon");
            model.Id = dr.Field<int>("id");
            model.ProductName = dr.Field<string>("product_name");
            model.Monqty = dr.Field<int>("monqty");
            model.Monsalesroom = dr.Field<decimal>("monsalerroom");
            model.Yearqty = dr.Field<int>("yearqty");
            model.Yearsalesroom = dr.Field<decimal>("yearsalerroom");

            result.Add(model);
        }
    }
    return result;
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    this.Dispose();
}
}

public DataTable GetProduct()
{
    try
    {
        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select * from product";
            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

```

        finally
        {
            this.Dispose();
        }
    }
    public bool UpdateProduct(DataTable dataTable)
    {
        try
        {
            if (this.DBConnection())
            {
                string sql = @"select * from product";
                adapter = new SqlDataAdapter(sql, conn);
                builder = new SqlCommandBuilder(adapter);
                adapter.Update(dataTable);
                return true;
            }
            return false;
        }
        catch (Exception ex)
        {
            throw ex;
        }
        finally
        {
            this.Dispose();
        }
    }
    public List<ProductModel> GetStore()
    {
        try
        {
            List<ProductModel> result = new List<ProductModel>();
            if (this.DBConnection())
            {
                string sql = @"if object_id('tempdb..#temp') is not null
drop table #temp;
create table #temp
(
    id            int,
    code          char(2),
    pid           int,
    product_name  varchar(20),
    amt           int

```

```

);
insert into #temp exec SelectStore;
select id,product_name,isnull(amt,0) as amt
from #temp;";

        adapter = new SqlDataAdapter(sql, conn);
        DataTable table = new DataTable();
        int count = adapter.Fill(table);
        if (count > 0)
        {
            ProductModel model = null;
            foreach (DataRow dr in table.AsEnumerable())
            {
                model = new ProductModel();
                model.Id = dr.Field<int>("id");
                model.ProductName = dr.Field<string>("product_name");
                model.Amt = dr.Field<int>("amt");

                result.Add(model);
            }
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public DataTable GetEmployee()
{
    try
    {
        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select * from employee";
            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
    }
}

```

```

        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public bool UpdateEmployee(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {
            string sql = @"select * from employee";
            adapter = new SqlDataAdapter(sql, conn);
            builder = new SqlCommandBuilder(adapter);
            adapter.Update(dataTable);
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public DataTable GetLogin()
{
    try
    {
        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select employee_id as '员工 id', login_name as '登录用户名', login_password as '密码', logo as '头像', is_admin as '管理员权限', is_valid as '生效', is_can_login as '允许登录' from login";

```

```

        adapter = new SqlDataAdapter(sql, conn);
        adapter.Fill(result);
    }
    return result;
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    this.Dispose();
}
}
public bool UpdateLogin(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {
            string sql = @"select employee_id as '员工 id', login_name as '登录用户名', login_password as '密码', logo as '头像', is_admin as '管理员权限', is_valid as '生效', is_can_login as '允许登录' from login";
            adapter = new SqlDataAdapter(sql, conn);
            builder = new SqlCommandBuilder(adapter);
            adapter.Update(dataTable);
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}
public DataTable GetDepart()
{
    try
    {
        DataTable result = new DataTable();
    }
}

```

```

        if (this.DBConnection())
        {
            string sql = @"select * from depart";
            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public bool UpdateDepart(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {
            string sql = @"select * from depart";
            adapter = new SqlDataAdapter(sql, conn);
            builder = new SqlCommandBuilder(adapter);
            adapter.Update(dataTable);
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public DataTable GetIn()
{
    try
    {

```



```

        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select * from [in] order by document_id";
            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public bool UpdateIn(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {
            string sql = @"select * from [in] order by document_id";
            adapter = new SqlDataAdapter(sql, conn);
            builder = new SqlCommandBuilder(adapter);
            adapter.Update(dataTable);
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public DataTable GetInInfo()
{

```

```

        try
        {
            DataTable result = new DataTable();
            if (this.DBConnection())
            {
                string sql = @"select id as '商品 id', document_id as '单号', quantity as '
数量', price as '价格', remark as '备注'
from document_info
where document_id like 'IN%'";
                adapter = new SqlDataAdapter(sql, conn);
                adapter.Fill(result);
            }
            return result;
        }
        catch (Exception ex)
        {
            throw ex;
        }
        finally
        {
            this.Dispose();
        }
    }

    public bool UpdateInInfo(DataTable dataTable)
    {
        try
        {
            if (this.DBConnection())
            {
                string sql = @"select id as '商品 id', document_id as '单号', quantity as '
数量', price as '价格', remark as '备注'
from document_info
where document_id like 'IN%'";
                adapter = new SqlDataAdapter(sql, conn);
                builder = new SqlCommandBuilder(adapter);
                adapter.Update(dataTable);
                return true;
            }
            return false;
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

```

```

        finally
        {
            this.Dispose();
        }
    }

    public DataTable GetCurrentSupplier()
    {
        try
        {
            DataTable result = new DataTable();
            if (this.DBConnection())
            {
                string sql = @"select *
from (select * from supplier where supplier_set_date < GETDATE()) a
where supplier_set_date >= all
(
    select b.supplier_set_date
    from (select * from supplier where supplier_set_date < GETDATE()) b
    where b.supplier_id = a.supplier_id
)";

                adapter = new SqlDataAdapter(sql, conn);
                adapter.Fill(result);
            }
            return result;
        }
        catch (Exception ex)
        {
            throw ex;
        }
        finally
        {
            this.Dispose();
        }
    }

    public DataTable GetSupplier()
    {
        try
        {
            DataTable result = new DataTable();
            if (this.DBConnection())
            {
                string sql = @"select supplier_id          as '供应商 ID',

```

```

        supplier_info_id as '供应商信息版本号',
        supplier_set_date as '信息变更时间',
        supplier_name      as '名称',
        supplier_address   as '地址',
        supplier_phone     as '联系电话'
from supplier";

        adapter = new SqlDataAdapter(sql, conn);
        adapter.Fill(result);
    }
    return result;
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    this.Dispose();
}
}

public bool UpdateSupplier(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {
            string sql = @"select supplier_id      as '供应商 ID',
supplier_info_id as '供应商信息版本号',
supplier_set_date as '信息变更时间',
supplier_name     as '名称',
supplier_address  as '地址',
supplier_phone    as '联系电话'
from supplier";

            adapter = new SqlDataAdapter(sql, conn);
            builder = new SqlCommandBuilder(adapter);
            adapter.Update(dataTable);
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

```

        finally
        {
            this.Dispose();
        }
    }

    public DataTable GetCustomer()
    {
        try
        {
            DataTable result = new DataTable();
            if (this.DBConnection())
            {
                string sql = @"select customer_id          as '顾客 ID',
customer_info_id  as '顾客信息版本号',
customer_set_date as '信息变更时间',
customer_name     as '名称',
customer_address  as '地址',
customer_phone_number  as '联系电话'
from customer";

                adapter = new SqlDataAdapter(sql, conn);
                adapter.Fill(result);
            }
            return result;
        }
        catch (Exception ex)
        {
            throw ex;
        }
        finally
        {
            this.Dispose();
        }
    }

    public bool UpdateCustomer(DataTable dataTable)
    {
        try
        {
            if (this.DBConnection())
            {
                string sql = @"select customer_id          as '顾客 ID',
customer_info_id  as '顾客信息版本号',
customer_set_date as '信息变更时间',
customer_name     as '名称',

```

```

        customer_address as '地址',
        customer_phone_number as '联系电话'
from customer";

        adapter = new SqlDataAdapter(sql, conn);
        builder = new SqlCommandBuilder(adapter);
        adapter.Update(dataTable);
        return true;
    }
    return false;
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    this.Dispose();
}
}

public DataTable GetCurrentCustomer()
{
    try
    {
        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select *
from (select * from customer where customer_set_date < GETDATE()) a
where customer_set_date >= all
(
    select b.customer_set_date
from (select * from customer where customer_set_date < GETDATE()) b
where b.customer_id = a.customer_id
)";

            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

```

        finally
        {
            this.Dispose();
        }
    }

    public DataTable GetOrder()
    {
        try
        {
            DataTable result = new DataTable();
            if (this.DBConnection())
            {
                string sql = @"select * from [order] order by document_id";
                adapter = new SqlDataAdapter(sql, conn);
                adapter.Fill(result);
            }
            return result;
        }
        catch (Exception ex)
        {
            throw ex;
        }
        finally
        {
            this.Dispose();
        }
    }

    public bool UpdateOrder(DataTable dataTable)
    {
        try
        {
            if (this.DBConnection())
            {
                string sql = @"select * from [order] order by document_id";
                adapter = new SqlDataAdapter(sql, conn);
                builder = new SqlCommandBuilder(adapter);
                adapter.Update(dataTable);
                return true;
            }
            return false;
        }
        catch (Exception ex)
        {

```



```

        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public DataTable GetOrderInfo()
{
    try
    {
        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select id as '商品 id', document_id as '单号', quantity as '
数量', price as '价格', remark as '备注'
from document_info
where document_id like 'OR%'";
            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public bool UpdateOrderInfo(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {
            string sql = @"select id as '商品 id', document_id as '单号', quantity as '
数量', price as '价格', remark as '备注'
from document_info
where document_id like 'OR%'";
            adapter = new SqlDataAdapter(sql, conn);

```

```

        builder = new SqlCommandBuilder(adapter);
        adapter.Update(dataTable);
        return true;
    }
    return false;
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    this.Dispose();
}
}

public DataTable GetOut()
{
    try
    {
        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select * from out order by document_id";
            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public bool UpdateOut(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {

```

```

        string sql = @"select * from out order by document_id";
        adapter = new SqlDataAdapter(sql, conn);
        builder = new SqlCommandBuilder(adapter);
        adapter.Update(dataTable);
        return true;
    }
    return false;
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    this.Dispose();
}
}

public DataTable GetOutInfo()
{
    try
    {
        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select id as '商品 id', document_id as '单号', quantity as '
数量', price as '价格', remark as '备注'
from document_info
where document_id like 'OU%'";
            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public bool UpdateOutInfo(DataTable dataTable)

```

```

        {
            try
            {
                if (this.DBConnection())
                {
                    string sql = @"select id as '商品 id', document_id as '单号', quantity as '
数量', price as '价格', remark as '备注'
from document_info
where document_id like 'OU%'";

                    adapter = new SqlDataAdapter(sql, conn);
                    builder = new SqlCommandBuilder(adapter);
                    adapter.Update(dataTable);
                    return true;
                }
                return false;
            }
            catch (Exception ex)
            {
                throw ex;
            }
            finally
            {
                this.Dispose();
            }
        }

        public DataTable GetDocumentIn()
        {
            try
            {
                DataTable result = new DataTable();
                if (this.DBConnection())
                {
                    string sql = @"select * from document where document_id like 'IN%'";
                    adapter = new SqlDataAdapter(sql, conn);
                    adapter.Fill(result);
                }
                return result;
            }
            catch (Exception ex)
            {
                throw ex;
            }
            finally
            {

```

```

        this.Dispose();
    }
}

public bool UpdateDocumentIn(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {
            string sql = @"select * from document where document_id like 'IN%';";
            adapter = new SqlDataAdapter(sql, conn);
            builder = new SqlCommandBuilder(adapter);
            adapter.Update(dataTable);
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}

public DataTable GetDocumentOut()
{
    try
    {
        DataTable result = new DataTable();
        if (this.DBConnection())
        {
            string sql = @"select * from document where document_id like 'OU%';";
            adapter = new SqlDataAdapter(sql, conn);
            adapter.Fill(result);
        }
        return result;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

```

        finally
        {
            this.Dispose();
        }
    }

    public bool UpdateDocumentOut(DataTable dataTable)
    {
        try
        {
            if (this.DBConnection())
            {
                string sql = @"select * from document where document_id like 'OU%';";
                adapter = new SqlDataAdapter(sql, conn);
                builder = new SqlCommandBuilder(adapter);
                adapter.Update(dataTable);
                return true;
            }
            return false;
        }
        catch (Exception ex)
        {
            throw ex;
        }
        finally
        {
            this.Dispose();
        }
    }

    public DataTable GetDocumentOrder()
    {
        try
        {
            DataTable result = new DataTable();
            if (this.DBConnection())
            {
                string sql = @"select * from document where document_id like 'OR%';";
                adapter = new SqlDataAdapter(sql, conn);
                adapter.Fill(result);
            }
            return result;
        }
        catch (Exception ex)
        {

```

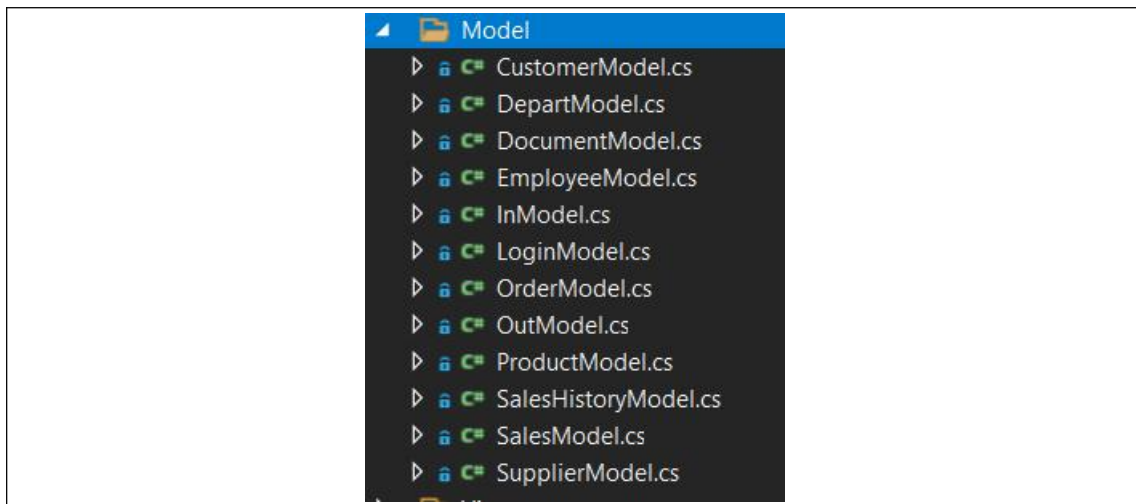
```

        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}
public bool UpdateDocumentOrder(DataTable dataTable)
{
    try
    {
        if (this.DBConnection())
        {
            string sql = @"select * from document where document_id like 'OR%'";
            adapter = new SqlDataAdapter(sql, conn);
            builder = new SqlCommandBuilder(adapter);
            adapter.Update(dataTable);
            return true;
        }
        return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        this.Dispose();
    }
}
}
}

```

2.5 wms\Model\

存放实体类



2.5.1 CustomerModel.cs

```
using wms.Common;

namespace wms.Model
{
    public class CustomerModel : NotifyBase
    {
        private string _customerId;

        public string CustomerId
        {
            get { return _customerId; }
            set { _customerId = value; this.DoNotify(); }
        }

        private string _customerInfoId;

        public string CustomerInfoId
        {
            get { return _customerInfoId; }
            set { _customerInfoId = value; this.DoNotify(); }
        }

        private string _customerName;

        public string CustomerName
        {
            get { return _customerName; }
            set { _customerName = value; this.DoNotify(); }
        }
    }
}
```

```
}  
}
```

2.5.2 DepartModel.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using wms.Common;  
  
namespace wms.Model  
{  
    public class DepartModel : NotifyBase  
    {  
        private string _departName;  
  
        public string DepartName  
        {  
            get { return _departName; }  
            set { _departName = value; this.DoNotify(); }  
        }  
  
        private string _departId;  
  
        public string DepartId  
        {  
            get { return _departId; }  
            set { _departId = value; this.DoNotify(); }  
        }  
    }  
}
```

2.5.3 DocumentModel.cs

```
using System;
```

```

using wms.Common;

namespace wms.Model
{
    public class DocumentModel : NotifyBase
    {
        private string _documentId;

        public string DocumentId
        {
            get { return _documentId; }
            set { _documentId = value; this.DoNotify(); }
        }

        private DateTime _documentDate;
        public DateTime DocumentDate
        {
            get { return _documentDate; }
            set { _documentDate = value; this.DoNotify(); }
        }
    }
}

```

2.5.4 EmployeeModel.cs

```

using wms.Common;

namespace wms.Model
{
    public class EmployeeModel : NotifyBase
    {
        private string _employeeId;

        public string EmployeeId
        {
            get { return _employeeId; }
            set { _employeeId = value; this.DoNotify(); }
        }

        private string _employeeName;

        public string EmployeeName

```

```

{
    get { return _employeeName; }
    set { _employeeName = value; this.DoNotify(); }
}

private string _employeePhone;

public string EmployeePhone
{
    get { return _employeePhone; }
    set { _employeePhone = value; this.DoNotify(); }
}

private int _employeeSex;

public int EmployeeSex
{
    get { return _employeeSex; }
    set { _employeeSex = value; this.DoNotify(); }
}

private string _employeeEducation;
public string EmployeeEducation
{
    get { return _employeeEducation; }
    set { _employeeEducation = value; this.DoNotify(); }
}

private string _employeeAddress;
public string EmployeeAddress
{
    get { return _employeeAddress; }
    set { _employeeAddress = value; this.DoNotify(); }
}

private string _employeeEmail;
public string EmployeeEmail
{
    get { return _employeeEmail; }
    set { _employeeEmail = value; this.DoNotify(); }
}

private string _employeeRemark;
public string EmployeeRemark

```

```

        {
            get { return _employeeRemark; }
            set { _employeeRemark = value; this.DoNotify(); }
        }
    }
}

```

2.5.5 InModel.cs

```

using System;
using wms.Common;

namespace wms.Model
{
    public class InModel : NotifyBase
    {
        private int _inStatus;

        public int InStatus
        {
            get { return _inStatus; }
            set { _inStatus = value; this.DoNotify(); }
        }
    }
}

```

2.5.6 LoginModel.cs

```

using wms.Common;

namespace wms.Model
{
    public class LoginModel : NotifyBase
    {
        private string _loginName;
        public string LoginName
        {

```

```

        get { return _loginName; }
        set
        {
            _loginName = value;
            this.DoNotify();
        }
    }

    private string _loginPassword;

    public string LoginPassword
    {
        get { return _loginPassword; }
        set
        {
            _loginPassword = value;
            this.DoNotify();
        }
    }

    private string _logo;

    public string Logo
    {
        get { return _logo; }
        set { _logo = value; this.DoNotify(); }
    }
}

```

2.5.7 OrderModel.cs

```

using System;
using wms.Common;

namespace wms.Model
{
    public class OrderModel : NotifyBase
    {
        private string _customerId;
        public string CustomerId
        {

```

```

        get { return _customerId; }
        set
        {
            _customerId = value;
            this.DoNotify();
        }
    }

    private string _Id;
    public string Id
    {
        get { return _Id; }
        set
        {
            _Id = value;
            this.DoNotify();
        }
    }

    private string _productName;
    public string ProductName
    {
        get { return _productName; }
        set
        {
            _productName = value;
            this.DoNotify();
        }
    }

    private string _price;
    public string Price
    {
        get { return _price; }
        set
        {
            _price = value;
            this.DoNotify();
        }
    }

    private string _quantity;
    public string Quantity
    {
        get { return _quantity; }
        set
        {
            _quantity = value;

```

```

        this.DoNotify();
    }
}

private string _total;
public string Total
{
    get { return _total; }
    set
    {
        _total = value;
        this.DoNotify();
    }
}

private int _orderStatus;

public int OrderStatus
{
    get { return _orderStatus; }
    set { _orderStatus = value; this.DoNotify(); }
}
}
}

```

2.5.8 OutModel.cs

```

using System;
using wms.Common;

namespace wms.Model
{
    public class OutModel : NotifyBase
    {
        private int _outStatus;

        public int OutStatus
        {
            get { return _outStatus; }
            set { _outStatus = value; this.DoNotify(); }
        }
    }
}

```



```
}
```

2.5.9 ProductModel.cs

```
using System;
using wms.Common;

namespace wms.Model
{
    public class ProductModel : NotifyBase
    {
        private int _id;
        public int Id
        {
            get { return _id; }
            set
            {
                _id = value;
                this.DoNotify();
            }
        }

        private string _code;
        public string Code
        {
            get { return _code; }
            set
            {
                _code = value;
                this.DoNotify();
            }
        }

        private int _pid;
        public int Pid
        {
            get { return _pid; }
            set
            {
                _pid = value;
                this.DoNotify();
            }
        }
    }
}
```

```

        private string _productName;
        public string ProductName
        {
            get { return _productName; }
            set
            {
                _productName = value;
                this.DoNotify();
            }
        }
        public int Amt { get; set; }
    }
}

```

2.5.10 SalesHistoryModel.cs

```

using System;
using wms.Common;

namespace wms.Model
{
    public class SalesHistoryModel
    {
        public int Hyear { get; set; }
        public int Hmon { get; set; }
        public int Id { get; set; }
        public string ProductName { get; set; }
        public int Monqty { get; set; }
        public decimal Monsalesroom { get; set; }
        public int Yearqty { get; set; }
        public decimal Yearsalesroom { get; set; }
    }
}

```

2.5.11 SalesModel.cs

```

using System;
using wms.Common;

```

```

namespace wms.Model
{
    public class SalesModel
    {
        public string DocumentId { get; set; }
        public DateTime DocumentDate { get; set; }
        public string CustomerId { get; set; }
        public int Id { get; set; }
        public string ProductName { get; set; }
        public decimal Price { get; set; }
        public int Quantity { get; set; }
        public decimal Total { get; set; }

    }
}

```

2.5.12 SupplierModel.cs

```

using wms.Common;

namespace wms.Model
{
    public class SupplierModel : NotifyBase
    {
        private string _supplierId;

        public string SupplierId
        {
            get { return _supplierId; }
            set { _supplierId = value; this.DoNotify(); }
        }

        private string _supplierInfoId;

        public string SupplierInfoId
        {
            get { return _supplierInfoId; }
            set { _supplierInfoId = value; this.DoNotify(); }
        }

        private string _supplierName;
    }
}

```

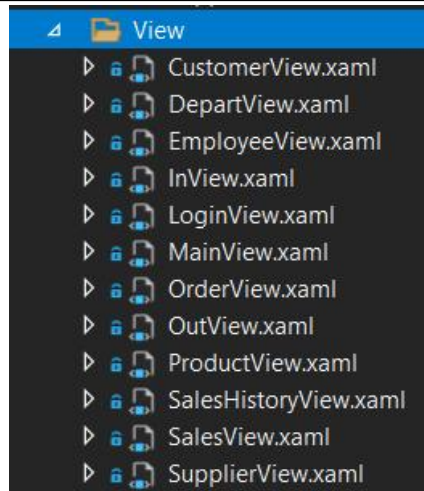
```

        public string SupplierName
        {
            get { return _supplierName; }
            set { _supplierName = value; this.DoNotify(); }
        }
    }
}

```

2.6 wms\View\

存放登录界面、主界面、以及用户控件子界面



2.6.1 CustomerView.xaml

```

<UserControl
    x:Class="wms.View.CustomerView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">
    <UserControl.DataContext>
        <local:CustomerViewModel />
    </UserControl.DataContext>
    <UserControl.Resources>

```

```

        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />

    </UserControl.Resources>
    <Grid Margin="24">
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>

        <Grid Grid.Row="0" Grid.Column="0">
            <Grid.RowDefinitions>
                <RowDefinition Height="36" />
                <RowDefinition />
            </Grid.RowDefinitions>
            <Border
                Grid.Row="0"
                Background="#37DC8A"
                CornerRadius="12, 12, 0, 0">
                <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="顾客信息"
            />
            </Border>
            <Border
                Grid.Row="1"
                Background="#37DC8A"
                CornerRadius="0, 0, 12, 12">
                <DataGrid
                    Margin="12, 0, 12, 12"
                    Background="#37dc8a"
                    BorderThickness="0"
                    CellStyle="{StaticResource CellStyle}"
                    ColumnHeaderStyle="{StaticResource ColumnHeaderStyle}"
                    FontSize="14"
                    ItemsSource="{Binding CustomerDataSet.Tables[0]}"
                    RowHeaderWidth="0"
                    RowStyle="{StaticResource RowStyle}" />
                </DataGrid>
            </Border>
        </Grid>

        <Grid Grid.Row="1" Margin="0, 24, 0, 0">
            <StackPanel
                HorizontalAlignment="Center"
                VerticalAlignment="Center"

```

```

        Orientation="Horizontal">
        <Button
            Command="{Binding SaveCommand}"
            Content="保存"
            Style="{StaticResource OperateButtonStyle}" />
    </StackPanel>

</Grid>
</Grid>
</UserControl>

```

2.6.2 DepartView.xaml

```

<UserControl
    x:Class="wms.View.DepartView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">
    <UserControl.DataContext>
        <local:DepartViewModel />
    </UserControl.DataContext>
    <UserControl.Resources>
        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
    </UserControl.Resources>
    <Grid Margin="24">
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>

        <Grid Grid.Row="0" Grid.Column="0">
            <Grid.RowDefinitions>
                <RowDefinition Height="36" />
                <RowDefinition />
            </Grid.RowDefinitions>

```

```

        </Grid.RowDefinitions>
        <Border
            Grid.Row="0"
            Background="#37DC8A"
            CornerRadius="12, 12, 0, 0">
            <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="部门信息"
        />

    </Border>
    <Border
        Grid.Row="1"
        Background="#37DC8A"
        CornerRadius="0, 0, 12, 12">
        <DataGrid
            Margin="12, 0, 12, 12"
            Background="#37dc8a"
            BorderThickness="0"
            CellStyle="{StaticResource CellStyle}"
            ColumnHeaderStyle="{StaticResource ColumnHeaderStyle}"
            FontSize="14"
            ItemsSource="{Binding DepartDataSet.Tables[0]}"
            RowHeaderWidth="0"
            RowStyle="{StaticResource RowStyle}" />
        </Border>

    </Grid>

    <Grid Grid.Row="1" Margin="0, 24, 0, 0">
        <StackPanel
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            Orientation="Horizontal">
            <Button
                Command="{Binding SaveCommand}"
                Content="保存"
                Style="{StaticResource OperateButtonStyle}" />
        </StackPanel>

    </Grid>
</Grid>
</UserControl>

```

2.6.3 EmployeeView.xaml

```
<UserControl
    x:Class="wms.View.EmployeeView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">
    <UserControl.DataContext>
        <local:EmployeeViewModel />
    </UserControl.DataContext>
    <UserControl.Resources>
        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
    </UserControl.Resources>
    <Grid Margin="12">
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="1*" />
                <ColumnDefinition Width="2*" />
            </Grid.ColumnDefinitions>

            <Grid Grid.Column="0">
                <Grid.RowDefinitions>
                    <RowDefinition Height="36" />
                    <RowDefinition />
                </Grid.RowDefinitions>
                <Border
                    Grid.Row="0"
                    Background="#37DC8A"
                    CornerRadius="12,12,0,0">
                    <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="员工信
息" />
                </Border>
                <Border
                    Grid.Row="1"

```



```

Background="#fff"
CornerRadius="0,0,12,12">
<ScrollView VerticalScrollBarVisibility="Hidden">
    <Grid Margin="18">
        <Grid.RowDefinitions>
            <RowDefinition Height="48" />
            <RowDefinition Height="48" />
            <RowDefinition Height="48" />
            <RowDefinition Height="48" />
            <RowDefinition Height="48" />
            <RowDefinition Height="48" />
            <RowDefinition Height="48" />
            <RowDefinition Height="48" />
            <RowDefinition Height="48" />
        </Grid.RowDefinitions>
        <StackPanel Grid.Row="0" VerticalAlignment="Center">
            <Grid>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition />
                    <ColumnDefinition Width="150" />
                </Grid.ColumnDefinitions>
                <TextBlock
RowTextBlockStyle}" Text="员工 ID" />
                    Style="{StaticResource
/>
                <TextBox
                    Grid.Column="2"
                    Width="150"
                    HorizontalAlignment="Right"
                    Template="{StaticResource TextBoxTemplate}"
                    Text="{Binding EmployeeModel.EmployeeId}"
                />
            </Grid>
        </StackPanel>
        <StackPanel Grid.Row="1" VerticalAlignment="Center">
            <Grid>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition />
                    <ColumnDefinition Width="150" />
                </Grid.ColumnDefinitions>
                <TextBlock
RowTextBlockStyle}" Text="部门名称" />
                    Style="{StaticResource
                <ComboBox
                    Grid.Column="2"
                    Width="150"

```

```

EmployeeDataSet.Tables[2]]"
HorizontalAlignment="Right"
DisplayMemberPath="depart_name"
ItemsSource="{Binding

DepartModel.DepartId}"
SelectedIndex="0"
SelectedValue="{Binding

SelectedValuePath="depart_id" />
</Grid>
</StackPanel>
<StackPanel Grid.Row="2" VerticalAlignment="Center">
<Grid>
<Grid.ColumnDefinitions>
<ColumnDefinition />

<ColumnDefinition Width="150" />
</Grid.ColumnDefinitions>
<TextBlock
RowTextBlockStyle}" Text="员工姓名" />
Style="{StaticResource

<TextBox
Grid.Column="2"
Width="150"
HorizontalAlignment="Right"
Template="{StaticResource TextBoxTemplate}"
Text="{Binding
EmployeeModel.EmployeeName}" />
</Grid>
</StackPanel>
<StackPanel Grid.Row="3" VerticalAlignment="Center">
<Grid>
<Grid.ColumnDefinitions>
<ColumnDefinition />

<ColumnDefinition Width="150" />
</Grid.ColumnDefinitions>
<TextBlock
RowTextBlockStyle}" Text="员工电话" />
Style="{StaticResource

<TextBox
Grid.Column="2"
Width="150"
HorizontalAlignment="Right"
Template="{StaticResource TextBoxTemplate}"
Text="{Binding
EmployeeModel.EmployeePhone}" />

```

```

        </Grid>
    </StackPanel>
    <StackPanel Grid.Row="4" VerticalAlignment="Center">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition Width="75" />

                <ColumnDefinition Width="75" />
            </Grid.ColumnDefinitions>
            <TextBlock
                Grid.Column="0"
                Style="{StaticResource RowTextBlockStyle}"
                Text="员工性别" />
            <StackPanel
                Grid.Column="1"
                HorizontalAlignment="Center"
                VerticalAlignment="Center">
                <RadioButton
                    Content="♂ "
                    FontSize="14"
                    IsChecked="{Binding
EmployeeModel.EmployeeSex, Converter={StaticResource boolToIntConverter},
ConverterParameter=1}"
                    Template="{StaticResource
SexRadioButtonTemplate}" />

                </StackPanel>
                <StackPanel
                    Grid.Column="2"
                    HorizontalAlignment="Center"
                    VerticalAlignment="Center">
                    <RadioButton
                        Content="♀ "
                        FontSize="14"
                        IsChecked="{Binding
EmployeeModel.EmployeeSex, Converter={StaticResource boolToIntConverter},
ConverterParameter=0}"
                        Template="{StaticResource
SexRadioButtonTemplate}" />

                    </StackPanel>

                </Grid>
            </StackPanel>

```

```

<StackPanel Grid.Row="5" VerticalAlignment="Center">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />

            <ColumnDefinition Width="150" />
        </Grid.ColumnDefinitions>
        <TextBlock
            RowTextBlockStyle}" Text="员工学历" />
            Style="{StaticResource

        <TextBox
            Grid.Column="2"
            Width="150"
            HorizontalAlignment="Right"
            Template="{StaticResource TextBoxTemplate}"
            Text="{Binding
EmployeeModel.EmployeeEducation}" />
        </Grid>
    </StackPanel>
    <StackPanel Grid.Row="6" VerticalAlignment="Center">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />

                <ColumnDefinition Width="150" />
            </Grid.ColumnDefinitions>
            <TextBlock
                RowTextBlockStyle}" Text="员工地址" />
                Style="{StaticResource

            <TextBox
                Grid.Column="2"
                Width="150"
                HorizontalAlignment="Right"
                Template="{StaticResource TextBoxTemplate}"
                Text="{Binding
EmployeeModel.EmployeeAddress}" />
            </Grid>
        </StackPanel>
        <StackPanel Grid.Row="7" VerticalAlignment="Center">
            <Grid>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition />

                    <ColumnDefinition Width="150" />
                </Grid.ColumnDefinitions>
                <TextBlock
                    Style="{StaticResource

```

```

RowTextBlockStyle}" Text="员工邮箱" />

        <TextBox
            Grid.Column="2"
            Width="150"
            HorizontalAlignment="Right"
            Template="{StaticResource TextBoxTemplate}"
            Text="{Binding
EmployeeModel.EmployeeEmail}" />
        </Grid>
    </StackPanel>
    <StackPanel Grid.Row="8" VerticalAlignment="Center">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />

                <ColumnDefinition Width="150" />
            </Grid.ColumnDefinitions>
            <TextBlock
RowTextBlockStyle}" Text="员工备注" />

                <TextBox
                    Grid.Column="2"
                    Width="150"
                    HorizontalAlignment="Right"
                    Template="{StaticResource TextBoxTemplate}"
                    Text="{Binding
EmployeeModel.EmployeeRemark}" />
            </Grid>
        </StackPanel>
    </Grid>
</ScrollView>
</Border>
</Grid>
<Grid Grid.Column="1" Margin="24,0,0,0">
    <Grid.RowDefinitions>
        <RowDefinition Height="36" />
        <RowDefinition />
    </Grid.RowDefinitions>
    <Border
        Grid.Row="0"
        Background="#37DC8A"
        CornerRadius="12,12,0,0">
        <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="登陆信
息" />
    </Border>

```

```

        <Border
            Grid.Row="1"
            Background="#37DC8A"
            CornerRadius="0,0,12,12">
            <DataGrid
                Margin="0,0,0,12"
                Background="#37dc8a"
                BorderThickness="0"
                CellStyle="{StaticResource CellStyle}"
                ColumnHeaderStyle="{StaticResource ColumnHeaderStyle}"
                FontSize="14"
                ItemsSource="{Binding EmployeeDataSet.Tables[1]}"
                RowHeaderWidth="0"
                RowStyle="{StaticResource RowStyle}" />
            </Border>

        </Grid>
    </Grid>
    <Grid Grid.Row="1" Margin="0,24,0,0">
        <StackPanel
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            Orientation="Horizontal">
            <Button
                Command="{Binding FirstPageCommand}"
                Content="|&lt;"
                Style="{StaticResource OperateButtonStyle}" />
            <Button
                Command="{Binding PrePageCommand}"
                Content="&lt;"
                Style="{StaticResource OperateButtonStyle}" />
            <Button
                Command="{Binding NextPageCommand}"
                Content="&gt;"
                Style="{StaticResource OperateButtonStyle}" />
            <Button
                Command="{Binding LastPageCommand}"
                Content="&gt;|"
                Style="{StaticResource OperateButtonStyle}" />
            <Button
                Command="{Binding InsertCommand}"
                Content="新增"
                Style="{StaticResource OperateButtonStyle}" />
            <Button

```

```

        Command="{Binding DeleteCommand}"
        Content="删除"
        Style="{StaticResource OperateButtonStyle}" />
    <Button
        Command="{Binding SaveCommand}"
        Content="保存"
        Style="{StaticResource OperateButtonStyle}" />
</StackPanel>

</Grid>
</Grid>
</UserControl>

```

2.6.4 InView.xaml

```

<UserControl
    x:Class="wms.View.InView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:ui="clr-namespace:wms.Controls;assembly=wms.Controls"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">

    <UserControl.DataContext>
        <local:InViewModel />
    </UserControl.DataContext>

    <UserControl.Resources>
        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
    </UserControl.Resources>

    <Grid Margin="12">
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="1*" />
            </Grid.ColumnDefinitions>
        </Grid>
    </Grid>

```

```

        <ColumnDefinition Width="2*" />
    </Grid.ColumnDefinitions>

    <Grid Grid.Column="0">
        <Grid.RowDefinitions>
            <RowDefinition Height="36" />
            <RowDefinition />
        </Grid.RowDefinitions>
        <Border
            Grid.Row="0"
            Background="#37DC8A"
            CornerRadius="12,12,0,0">
            <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="入库单
信息" />
        </Border>
        <Border
            Grid.Row="1"
            Background="ffff"
            CornerRadius="0,0,12,12">
            <ScrollView VerticalScrollBarVisibility="Hidden">
                <Grid Margin="18">
                    <Grid.RowDefinitions>
                        <RowDefinition Height="48" />
                        <RowDefinition Height="48" />
                        <RowDefinition Height="48" />
                        <RowDefinition Height="48" />
                        <RowDefinition Height="48" />
                        <RowDefinition Height="48" />
                    </Grid.RowDefinitions>
                    <StackPanel Grid.Row="0" VerticalAlignment="Center">
                        <Grid>
                            <Grid.ColumnDefinitions>
                                <ColumnDefinition />
                                <ColumnDefinition Width="150" />
                            </Grid.ColumnDefinitions>
                            <TextBlock
                                Style="{StaticResource
RowTextBlockStyle}" Text="入库单 ID" />
                            <TextBox
                                Grid.Column="2"
                                Width="150"
                                HorizontalAlignment="Right"
                                Template="{StaticResource TextBoxTemplate}"
                                Text="{Binding DocumentModel.DocumentId,
Mode=TwoWay}" />

```



```

        </Grid>
    </StackPanel>
    <StackPanel Grid.Row="1" VerticalAlignment="Center">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition Width="150" />
            </Grid.ColumnDefinitions>
            <TextBlock
                RowTextBlockStyle}" Text="供应商 ID" />
                Style="{StaticResource
            <TextBlock
                Grid.Column="2"
                Width="150"
                TextAlignment="Center"
                VerticalAlignment="Center"
                Text="{Binding SupplierModel.SupplierId,
                Mode=TwoWay}" />
            </Grid>
        </StackPanel>
        <StackPanel Grid.Row="2" VerticalAlignment="Center">
            <Grid>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition />
                    <ColumnDefinition Width="150" />
                </Grid.ColumnDefinitions>
                <TextBlock
                    RowTextBlockStyle}" Text="供应商" />
                    Style="{StaticResource
                <ComboBox
                    Grid.Column="2"
                    Width="150"
                    HorizontalAlignment="Right"
                    DisplayMemberPath="supplier_name"
                    ItemsSource="{Binding InDataSet.Tables[2]}"
                    SelectedIndex="1"
                    SelectedValue="{Binding
                    SupplierModel.SupplierId, Mode=TwoWay}"
                    SelectedValuePath="supplier_id" />
                </Grid>
            </StackPanel>
            <StackPanel Grid.Row="3" VerticalAlignment="Center">
                <Grid>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition />

```

```

        <ColumnDefinition Width="100" />
    </Grid.ColumnDefinitions>
    <TextBlock
        Style="{StaticResource
RowTextBlockStyle}" Text="供应商信息版本号" />
    <TextBox
        Grid.Column="2"
        Width="100"
        HorizontalAlignment="Right"
        Template="{StaticResource TextBoxTemplate}"
        Text="{Binding SupplierModel.SupplierInfoId,
Mode=TwoWay}" />
    </Grid>
</StackPanel>
<StackPanel Grid.Row="4" VerticalAlignment="Center">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />

            <ColumnDefinition Width="150" />
        </Grid.ColumnDefinitions>
        <TextBlock
            Style="{StaticResource
RowTextBlockStyle}" Text="入库单时间" />
        <ui:ucDateTimePicker
            Grid.Column="1"
            Width="150"
            HorizontalAlignment="Right"
            SelectedDate="{Binding
DocumentModel.DocumentDate,Mode=TwoWay}"
            />
    </Grid>
</StackPanel>
<StackPanel Grid.Row="5" VerticalAlignment="Center">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition Width="75" />
            <ColumnDefinition Width="75" />
        </Grid.ColumnDefinitions>
        <TextBlock
            Grid.Column="0"
            Style="{StaticResource RowTextBlockStyle}"
            Text="状态" />
    </Grid>
</StackPanel>

```

```

Grid.Column="1"
HorizontalAlignment="Center"
VerticalAlignment="Center">
<RadioButton
    Content="已完成"
    FontSize="14"
    IsChecked="{Binding InModel.InStatus,
Converter={StaticResource boolToIntConverter}, ConverterParameter=1}"
    Template="{StaticResource
SexRadioButtonTemplate}" />
</StackPanel>
<StackPanel
    Grid.Column="2"
    HorizontalAlignment="Center"
    VerticalAlignment="Center">
<RadioButton
    Content="未完成"
    FontSize="14"
    IsChecked="{Binding InModel.InStatus,
Converter={StaticResource boolToIntConverter}, ConverterParameter=0}"
    Template="{StaticResource
SexRadioButtonTemplate}" />
</StackPanel>
</Grid>
</StackPanel>
</Grid>
</ScrollView>
</Border>
</Grid>
<Grid Grid.Column="1" Margin="24,0,0,0">
    <Grid.RowDefinitions>
        <RowDefinition Height="36" />
        <RowDefinition />
    </Grid.RowDefinitions>
    <Border
        Grid.Row="0"
        Background="#37DC8A"
        CornerRadius="12,12,0,0">
        <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="入库单
详情" />
    </Border>
    <Border
        Grid.Row="1"
        Background="#37DC8A"

```

```

        CornerRadius="0,0,12,12">
        <DataGrid
            Margin="0,0,0,12"
            Background="#37dc8a"
            BorderThickness="0"
            CellStyle="{StaticResource CellStyle}"
            ColumnHeaderStyle="{StaticResource ColumnHeaderStyle}"
            FontSize="14"
            ItemsSource="{Binding InDataSet.Tables[1]}"
            RowHeaderWidth="0"
            RowStyle="{StaticResource RowStyle}" />
    </Border>

</Grid>
</Grid>
<Grid Grid.Row="1" Margin="0,24,0,0">
    <StackPanel
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Orientation="Horizontal">
        <Button
            Command="{Binding FirstPageCommand}"
            Content="|&lt;"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding PrePageCommand}"
            Content="&lt;"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding NextPageCommand}"
            Content="&gt;"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding LastPageCommand}"
            Content="&gt;|"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding InsertCommand}"
            Content="新增"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding DeleteCommand}"
            Content="删除"
            Style="{StaticResource OperateButtonStyle}" />
    </StackPanel>
</Grid>

```

```

        <Button
            Command="{Binding SaveCommand}"
            Content="保存"
            Style="{StaticResource OperateButtonStyle}" />
    </StackPanel>

</Grid>
</Grid>
</UserControl>

```

2.6.5 LoginView.xaml

```

<Window
    x:Class="wms.View.LoginView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:common="clr-namespace:wms.Common"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

    xmlns:i="clr-namespace:System.Windows.Interactivity;assembly=System.Windows.Interactivity"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    Name="window"
    Title="登录易知"
    Width="320"
    Height="600"
    AllowsTransparency="True"
    Background="{x:Null}"
    FontFamily="Microsoft YaHei"
    FontWeight="ExtraLight"
    ResizeMode="NoResize"
    WindowStartupLocation="CenterScreen"
    WindowStyle="None"
    mc:Ignorable="d">
    <Window.DataContext>
        <local:LoginViewModel />
    </Window.DataContext>
    <Window.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="..\Assets/Styles/DefaultStyle.xaml" />
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Window.Resources>

```

```

        <ResourceDictionary>
            <SolidColorBrush x:Key="TextBox.Static.Border" Color="#ccc" />
            <SolidColorBrush x:Key="TextBox.MouseOver.Border" Color="#39DBD5"
/>

            <SolidColorBrush x:Key="TextBox.Focus.Border" Color="#36cfc9" />
            <ControlTemplate
                x:Key="LoginNameTextBoxTemplate"
TargetType="{x:Type TextBox}">
                <Border
                    x:Name="border"
                    Padding="6,0,6,0"
                    Background="{TemplateBinding Background}"
                    BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding BorderThickness}"
                    CornerRadius="24"
                    SnapsToDevicePixels="True">
                    <Grid>
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="40" />
                            <ColumnDefinition />
                        </Grid.ColumnDefinitions>
                        <TextBlock
                            Grid.Column="0"
                            HorizontalAlignment="Center"
                            VerticalAlignment="Center"
                            FontFamily=" ../Assets/Fonts/#iconfont"
                            FontSize="24"
                            Foreground="#DDD"
                            Text="&#xe623;" />
                        <ScrollViewer
                            x:Name="PART_ContentHost"
                            Grid.Column="1"
                            MinHeight="20"
                            VerticalAlignment="Center"
                            Focusable="false"
                            HorizontalScrollBarVisibility="Hidden"
                            VerticalScrollBarVisibility="Hidden" />
                    </Grid>

                </Border>
                <ControlTemplate.Triggers>
                    <Trigger Property="IsEnabled" Value="false">
                        <Setter
                            TargetName="border"
                            Property="Opacity"
Value="0.56" />
                    </Trigger>
                </ControlTemplate.Triggers>
            </ControlTemplate>
        </ResourceDictionary>
    
```

```

        <Trigger Property="IsMouseOver" Value="true">
            <Setter TargetName="border" Property="BorderBrush"
Value="{StaticResource TextBox.MouseOver.Border}" />
        </Trigger>
        <Trigger Property="IsKeyboardFocused" Value="true">
            <Setter TargetName="border" Property="BorderBrush"
Value="{StaticResource TextBox.Focus.Border}" />
        </Trigger>
    </ControlTemplate.Triggers>
</ControlTemplate>
<ControlTemplate                                x:Key="LoginPasswordBoxTemplate"
TargetType="{x:Type PasswordBox}">
    <Border
        x:Name="border"
        Padding="6,0,6,0"
        Background="{TemplateBinding Background}"
        BorderBrush="{TemplateBinding BorderBrush}"
        BorderThickness="{TemplateBinding BorderThickness}"
        CornerRadius="24"
        SnapsToDevicePixels="True">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="40" />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>
            <TextBlock
                Grid.Column="0"
                HorizontalAlignment="Center"
                VerticalAlignment="Center"
                FontFamily=" ../Assets/Fonts/#iconfont"
                FontSize="24"
                Foreground="#DDD"
                Text="&#xe603;" />
            <ScrollViewer
                x:Name="PART_ContentHost"
                Grid.Column="1"
                MinHeight="20"
                VerticalAlignment="Center"
                Focusable="false"
                HorizontalScrollBarVisibility="Hidden"
                VerticalScrollBarVisibility="Hidden" />
        </Grid>
    </Border>
</ControlTemplate.Triggers>

```

```

        <Trigger Property="IsEnabled" Value="false">
            <Setter TargetName="border" Property="Opacity"
Value="0.56" />
        </Trigger>
        <Trigger Property="IsMouseOver" Value="true">
            <Setter TargetName="border" Property="BorderBrush"
Value="{StaticResource TextBox.MouseOver.Border}" />
        </Trigger>
        <Trigger Property="IsKeyboardFocused" Value="true">
            <Setter TargetName="border" Property="BorderBrush"
Value="{StaticResource TextBox.Focus.Border}" />
        </Trigger>
    </ControlTemplate.Triggers>
</ControlTemplate>
<ControlTemplate x:Key="VerificationCodeTextBoxTemplate"
TargetType="{x:Type TextBox}">
    <Border
        x:Name="border"
        Padding="6,6,6,6"
        Background="{TemplateBinding Background}"
        BorderBrush="{TemplateBinding BorderBrush}"
        BorderThickness="{TemplateBinding BorderThickness}"
        CornerRadius="24"
        SnapsToDevicePixels="True">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="40" />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>
            <TextBlock
                Grid.Column="0"
                HorizontalAlignment="Center"
                VerticalAlignment="Center"
                FontFamily=" ../Assets/Fonts/#iconfont"
                FontSize="24"
                Foreground="#DDD"
                Text="&#xe656;" />
            <ScrollViewer
                x:Name="PART_ContentHost"
                Grid.Column="1"
                MinHeight="20"
                VerticalAlignment="Center"
                Focusable="false"
                HorizontalScrollBarVisibility="Hidden"

```



```

        VerticalScrollBarVisibility="Hidden" />
    </Grid>
</Border>
<ControlTemplate.Triggers>
    <Trigger Property="IsEnabled" Value="false">
        <Setter TargetName="border" Property="Opacity"
Value="0.56" />
    </Trigger>
    <Trigger Property="IsMouseOver" Value="true">
        <Setter TargetName="border" Property="BorderBrush"
Value="{StaticResource TextBox.MouseOver.Border}" />
    </Trigger>
    <Trigger Property="IsKeyboardFocused" Value="true">
        <Setter TargetName="border" Property="BorderBrush"
Value="{StaticResource TextBox.Focus.Border}" />
    </Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
<ControlTemplate x:Key="LoginButtonTemplate" TargetType="Button">
    <Border CornerRadius="32">
        <Border.Background>
            <LinearGradientBrush StartPoint="0,0" EndPoint="1,0">
                <GradientStop Offset="0" Color="#39DBD5" />
                <GradientStop Offset="1" Color="#37DC8A" />
            </LinearGradientBrush>
        </Border.Background>
        <Grid>
            <Border
                Name="back"
                Background="#22FFFFFF"
                BorderThickness="0"
                CornerRadius="32"
                Visibility="Hidden" />
            <ContentControl
                HorizontalAlignment="Center"
                VerticalAlignment="Center"
                Content="{TemplateBinding Content}"
                Foreground="{TemplateBinding Foreground}" />
        </Grid>
    </Border>
    <ControlTemplate.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter TargetName="back" Property="Visibility"
Value="Visible" />
        </Trigger>
    </ControlTemplate.Triggers>
</ControlTemplate>

```

```

        </Trigger>
        <Trigger Property="IsEnabled" Value="False">
            <Setter      TargetName="back"      Property="Visibility"
Value="Visible" />
            <Setter      TargetName="back"      Property="Background"
Value="#eee" />
            <Setter      TargetName="back"      Property="Opacity"
Value="0.5" />

        </Trigger>
    </ControlTemplate.Triggers>
</ControlTemplate>
</ResourceDictionary>
</ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
</Window.Resources>
<Border Background="#f7f7f7" CornerRadius="12">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="4*" />
            <RowDefinition Height="7*" />
            <RowDefinition Height="2*" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Border Background="#36cfc9" CornerRadius="12,12,0,0">
                <i:Interaction.Triggers>
                    <i:EventTrigger EventName="MouseLeftButtonDown">
                        <i:InvokeCommandAction      Command="{Binding
MoveWindowCommand}" CommandParameter="{Binding ElementName=window}" />
                    </i:EventTrigger>
                </i:Interaction.Triggers>
            </Border>
            <Button
                Margin="12"
                HorizontalAlignment="Right"
                VerticalAlignment="Top"
                Command="{Binding CloseWindowCommand}"
                CommandParameter="{Binding ElementName=window}"
                Content="✕"
                Style="{StaticResource WindowControlButtonStyle}" />
            <StackPanel Margin="0,0,0,32" VerticalAlignment="Bottom">
                <Border
                    Width="88"
                    Height="88"

```

```

        Margin="0,0,0,12"
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Background="White"
        CornerRadius="44">
        <Border HorizontalAlignment="Center">
            <TextBlock
                HorizontalAlignment="Center"
                VerticalAlignment="Center"
                FontFamily=" ../Assets/Fonts/#iconfont"
                FontSize="64"
                Foreground="#36cfc9"
                Text="&#xe609;" />
            </Border>
        </Border>
        <TextBlock
            HorizontalAlignment="Center"
            FontSize="18"
            Foreground="White"
            Text="易知仓库管理平台" />
    </StackPanel>
</Grid>
<Border
    Grid.Row="1"
    Background="Transparent"
    Visibility="{Binding ShowProgress}">
    <ProgressBar
        Height="3"
        VerticalAlignment="Top"
        Background="Transparent"
        BorderThickness="0"
        Foreground="#37DC8A"
        IsIndeterminate="True" />
    </Border>
<Grid
    Grid.Row="1"
    Margin="30,10"
    FocusManager.FocusedElement="{Binding ElementName=tb_LoginName}">

    <Grid.RowDefinitions>
        <RowDefinition />
        <RowDefinition />
        <RowDefinition />
        <RowDefinition />
    </Grid.RowDefinitions>

```

```

        <RowDefinition Height="24" />
    </Grid.RowDefinitions>
    <TextBox
        Name="tb_LoginName"
        Height="48"
        FontSize="16"
        Foreground="#666"
        Template="{StaticResource LoginNameTextBoxTemplate}"
        Text="{Binding
UpdateSourceTrigger=PropertyChanged}" LoginModel.LoginName,
        <TextBox.InputBindings>
            <KeyBinding
                Key="Enter"
                Command="{Binding LoginCommand}"
                CommandParameter="{Binding ElementName=window}" />
        </TextBox.InputBindings>
    </TextBox>
    <PasswordBox
        Grid.Row="1"
        Height="48"
        common:PasswordHelper.Attach="True"
        common:PasswordHelper.Password="{Binding
LoginModel.LoginPassword, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
        FontSize="16"
        Foreground="#666"
        Template="{StaticResource LoginPasswordBoxTemplate}">
        <PasswordBox.InputBindings>
            <KeyBinding
                Key="Enter"
                Command="{Binding LoginCommand}"
                CommandParameter="{Binding ElementName=window}" />
        </PasswordBox.InputBindings>
    </PasswordBox>
    <Grid Grid.Row="2">
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition Width="100" />
        </Grid.ColumnDefinitions>
        <TextBox
            Grid.Column="0"
            Height="48"
            FontSize="16"
            Template="{StaticResource VerificationCodeTextBoxTemplate}"
            Text="{Binding
VerificationCode,

```

```

UpdateSourceTrigger=PropertyChanged}]">
    <TextBox.InputBindings>
        <KeyBinding
            Key="Enter"
            Command="{Binding LoginCommand}"
            CommandParameter="{Binding ElementName=window}"
        />
    </TextBox.InputBindings>
</TextBox>
<Button
    x:Name="VerificationCode"
    Grid.Column="1"
    Width="72"
    Height="32"
    HorizontalAlignment="Center"
    VerticalAlignment="Center"
    Panel.ZIndex="1"
    Background="Transparent"
    Opacity="0.5"
    Cursor="Hand"
    Command="{Binding RefreshVerificationCodeCommand}">
    <Image Source="{Binding Source}" />
</Button>

</Grid>

<Button
    Grid.Row="3"
    Width="64"
    Height="64"
    BorderThickness="0"
    Command="{Binding LoginCommand}"
    CommandParameter="{Binding ElementName=window}"
    Content="→"
    FontSize="20"
    FontWeight="Bold"
    Foreground="White"
    Template="{StaticResource LoginButtonTemplate}" />
<TextBlock
    Grid.Row="4"
    Margin="0,6,0,0"
    HorizontalAlignment="Center"
    FontSize="14"
    Foreground="#36cfc9"

```

```

        LineHeight="22"
        Text="{Binding ErrorMessage}"
        TextWrapping="Wrap" />
    </Grid>
    <Grid Grid.Row="2" Margin="30,0">
        <Grid.RowDefinitions>
            <RowDefinition Height="20" />
            <RowDefinition />
        </Grid.RowDefinitions>
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition Width="30" />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>
            <Border
                VerticalAlignment="Center"
                BorderBrush="#DDD"
                BorderThickness="0,0,0,1" />
            <Border
                Grid.Column="2"
                VerticalAlignment="Center"
                BorderBrush="#DDD"
                BorderThickness="0,0,0,1" />
            <TextBlock
                Grid.Column="1"
                HorizontalAlignment="Center"
                VerticalAlignment="Center"
                Foreground="#CCC"
                Text="OR" />
        </Grid>
    </Grid>
    <UniformGrid
        Grid.Row="1"
        Margin="12"
        Columns="7">
        <UniformGrid.Resources>
            <Style TargetType="TextBlock">
                <Setter Property="Foreground" Value="#ccc" />
                <Setter Property="FontSize" Value="36" />
                <Setter Property="HorizontalAlignment" Value="Center" />
                <Setter Property="VerticalAlignment" Value="Center" />
                <Setter
                    Property="FontFamily"
Value="..\Assets/Fonts/#iconfont" />
            </Style>
        </UniformGrid.Resources>
    </UniformGrid>
    <Style.Triggers>

```

```

        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Foreground" Value="#36cfc9" />
        </Trigger>
    </Style.Triggers>
</Style>
</UniformGrid.Resources>
<TextBlock Text="&#xe711;" />
<Border />
<TextBlock Text="&#xe64e;" />
<Border />
<TextBlock Text="&#xe665;" />
<Border />
<TextBlock Text="&#xe641;" />
</UniformGrid>
</Grid>

</Grid>

</Border>
</Window>

```

2.6.6 MainView.xaml

```

<Window
    x:Class="wms.View.MainView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

    xmlns:i="clr-namespace:System.Windows.Interactivity;assembly=System.Windows.Interactivity"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:ui="clr-namespace:wms.Controls;assembly=wms.Controls"
    Name="window"
    Title="易知仓库管理平台"
    Width="960"
    Height="640"
    AllowsTransparency="True"
    Background="{x:Null}"
    FontFamily="Microsoft YaHei"
    FontWeight="ExtraLight"

```

```

ResizeMode="NoResize"
WindowStartupLocation="CenterScreen"
WindowState="Normal"
WindowStyle="None"
mc:Ignorable="d">
<Window.DataContext>
    <local:MainViewModel />
</Window.DataContext>
<Window.Resources>
    <ResourceDictionary>
        <ResourceDictionary.MergedDictionaries>
            <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
        </ResourceDictionary>
        <Style x:Key="NavButtonStyle" TargetType="RadioButton">
            <Setter Property="Foreground" Value="White" />
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="RadioButton">
                        <Border
                            Name="border"
                            Background="Transparent"
                            CornerRadius="4">
                            <ContentControl
                                Margin="12,4,12,4"
                                HorizontalAlignment="Center"
                                VerticalAlignment="Center"
                                Content="{TemplateBinding Content}"
                                FontSize="14" />
                            </ContentControl>
                        </Border>
                        <ControlTemplate.Triggers>
                            <Trigger
                                Value="True">
                                <Setter
                                    Property="Background" Value="#22FFFFFF" />
                                </Trigger>
                            <Trigger Property="IsChecked" Value="True">
                                <Setter
                                    Property="Background" Value="#44FFFFFF" />
                                </Trigger>
                        </ControlTemplate.Triggers>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>

```



```

<Style x:Key="ZoomWindowButtonStyle" TargetType="Button">
    <Setter Property="Foreground" Value="#fff" />
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="Button">
                <Border Background="Transparent">
                    <TextBlock
                        x:Name="tb"
                        HorizontalAlignment="Center"
                        VerticalAlignment="Center"
                        FontFamily="../Assets/Fonts/#iconfont"
                        FontSize="24" />
                </Border>
                <ControlTemplate.Triggers>
                    <Trigger
                        Property="IsMouseOver"
                        Value="True">
                        <Setter
                            TargetName="tb"
                            Property="Foreground" Value="#22FFFFFF" />
                    </Trigger>
                    <Trigger Property="IsPressed" Value="True">
                        <Setter
                            TargetName="tb"
                            Property="Foreground" Value="#44FFFFFF" />
                    </Trigger>
                    <Trigger Property="Uid" Value="1">
                        <Setter TargetName="tb" Property="Text"
                            Value="&#xe70e;" />
                    </Trigger>
                    <Trigger Property="Uid" Value="0">
                        <Setter TargetName="tb" Property="Text"
                            Value="&#xe676;" />
                    </Trigger>
                </ControlTemplate.Triggers>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
<Style x:Key="RefreshButtonStyle" TargetType="Button">
    <Style.Triggers>
        <Trigger Property="Uid" Value="1">
            <Setter
                Property="CommandParameter"
                Value="SalesView" />
        </Trigger>
        <Trigger Property="Uid" Value="2">
            <Setter
                Property="CommandParameter"

```

```

Value="SalesHistoryView" />
        </Trigger>
        <Trigger Property="Uid" Value="3">
            <Setter
                Property="CommandParameter"
Value="ProductView" />
        </Trigger>
        <Trigger Property="Uid" Value="4">
            <Setter Property="CommandParameter" Value="InView"
/>
        </Trigger>
        <Trigger Property="Uid" Value="5">
            <Setter
                Property="CommandParameter"
Value="OrderView" />
        </Trigger>
        <Trigger Property="Uid" Value="6">
            <Setter
                Property="CommandParameter"
Value="OutView" />
        </Trigger>
        <Trigger Property="Uid" Value="7">
            <Setter
                Property="CommandParameter"
Value="SupplierView" />
        </Trigger>
        <Trigger Property="Uid" Value="8">
            <Setter
                Property="CommandParameter"
Value="CustomerView" />
        </Trigger>
        <Trigger Property="Uid" Value="9">
            <Setter
                Property="CommandParameter"
Value="EmployeeView" />
        </Trigger>
        <Trigger Property="Uid" Value="10">
            <Setter
                Property="CommandParameter"
Value="DepartView" />
        </Trigger>
    </Style.Triggers>
</Style>
<Storyboard x:Key="UserInfoStoryboard">
    <DoubleAnimation
        Storyboard.TargetName="tt"
        Storyboard.TargetProperty="X"
        To="0"
        Duration="0:0:0.2" />
</Storyboard>
<Storyboard x:Key="CloseUserInfoStoryboard">

```

```

        <DoubleAnimation
            Storyboard.TargetName="tt"
            Storyboard.TargetProperty="X"
            Duration="0:0:0.1" />
    </Storyboard>
</ResourceDictionary>
</ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
</Window.Resources>
<Window.Triggers>
    <EventTrigger RoutedEvent="Button.Click" SourceName="btnUsreInfo">
        <BeginStoryboard Storyboard="{StaticResource UserInfoStoryboard}" />
    </EventTrigger>
    <EventTrigger RoutedEvent="Button.Click" SourceName="btnCloseUserInfo">
        <BeginStoryboard Storyboard="{StaticResource CloseUserInfoStoryboard}" />
    </EventTrigger>
</Window.Triggers>
<Border Background="#F5F5F6" CornerRadius="12">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="140" />
            <RowDefinition />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Border
                Height="140"
                VerticalAlignment="Top"
                Background="#36cfc9"
                CornerRadius="12,12,0,0">
                <i:Interaction.Triggers>
                    <i:EventTrigger EventName="MouseLeftButtonDown">
                        <i:InvokeCommandAction Command="{Binding
MoveWindowCommand}" CommandParameter="{Binding ElementName=window}" />
                    </i:EventTrigger>
                </i:Interaction.Triggers>
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition />
                    <RowDefinition Height="auto" />
                </Grid.RowDefinitions>
                <StackPanel
                    HorizontalAlignment="Right"
                    VerticalAlignment="Top"
                    Orientation="Horizontal">

```

```

<Button
    Margin="8"
    Command="{Binding MinWindowCommand}"
    CommandParameter="{Binding ElementName=window}"
    Content="⏮;"
    Style="{StaticResource WindowControlButtonStyle}" />
<Button
    Uid="{Binding IsNormal}"
    Margin="8"
    Command="{Binding ZoomWindowCommand}"
    CommandParameter="{Binding ElementName=window}"
    Style="{StaticResource ZoomWindowButtonStyle}" />
<Button
    Margin="8"
    Command="{Binding CloseWindowCommand}"
    CommandParameter="{Binding ElementName=window}"
    Content="⏮;"
    Style="{StaticResource WindowControlButtonStyle}" />
</StackPanel>
<Grid
    Grid.Row="0"
    Height="48"
    Margin="24,24,24,0"
    HorizontalAlignment="Left"
    VerticalAlignment="Top">
    <Grid.ColumnDefinitions>
        <ColumnDefinition />
        <ColumnDefinition />
        <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <Border
        Grid.Column="0"
        Width="48"
        Height="48"
        Background="White"
        CornerRadius="24">
        <TextBlock
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            FontFamily=" ../Assets/Fonts/#iconfont"
            FontSize="32"
            Foreground="#36cfc9"
            Text="⏮;" />
    </Border>

```

```

        <Border Grid.Column="1">
            <TextBlock
                Padding="8"
                VerticalAlignment="Center"
                FontSize="16"
                Foreground="White"
                Text="易知仓库管理平台" />
        </Border>
    </Grid>
    <Border
        Grid.Row="0"
        Height="48"
        Margin="0,48,0,0"
        HorizontalAlignment="Right"
        VerticalAlignment="Top"
        Background="#39DBD5"
        CornerRadius="24,0,0,24">
        <Grid Margin="12,0,12,0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>
            <Border
                Grid.Column="0"
                Width="36"
                Height="36"
                CornerRadius="18">
                <Border.Background>
                    <ImageBrush
                        ImageSource="{Binding
UserEntity.Logo}" />
                </Border.Background>
            </Border>
            <Border Grid.Column="1">
                <Button
                    Name="btnUsreInfo"
                    Margin="8"
                    HorizontalAlignment="Center"
                    VerticalAlignment="Center"
                    Cursor="Hand"
                    FontSize="14"
                    Foreground="White">
                    <Button.Template>
                        <ControlTemplate TargetType="Button">

```

```

                                <TextBlock                                Text="{Binding
UserEntity.LoginName}" />

                                </ControlTemplate>
                                </Button.Template>

                                </Button>

                                </Border>
                                <Border Grid.Column="2">
                                    <Button
                                        Command="{Binding LogoutCommand}"
                                        CommandParameter="{Binding
ElementName=window}"
                                        Content="退出"
                                        Style="{StaticResource
WindowControlButtonStyle}" />
                                </Border>
                                </Grid>

                                </Border>
                                <StackPanel
                                    Grid.Row="1"
                                    HorizontalAlignment="Left"
                                    VerticalAlignment="Center"
                                    Orientation="Horizontal">
                                    <RadioButton
                                        Command="{Binding NavChangedCommand}"
                                        CommandParameter="SalesView"
                                        Content="销售管理"
                                        IsChecked="True"
                                        Style="{StaticResource NavButtonStyle}" />
                                    <RadioButton
                                        Command="{Binding NavChangedCommand}"
                                        CommandParameter="SalesHistoryView"
                                        Content="历史销售"
                                        Style="{StaticResource NavButtonStyle}" />
                                    <RadioButton
                                        Command="{Binding NavChangedCommand}"
                                        CommandParameter="ProductView"
                                        Content="商品管理"
                                        Style="{StaticResource NavButtonStyle}" />
                                    <RadioButton
                                        Command="{Binding NavChangedCommand}"
                                        CommandParameter="InView"

```

```

        Content="入库管理"
        Style="{StaticResource NavButtonStyle}" />
<RadioButton
    Command="{Binding NavChangedCommand}"
    CommandParameter="OrderView"
    Content="订单管理"
    Style="{StaticResource NavButtonStyle}" />
<RadioButton
    Command="{Binding NavChangedCommand}"
    CommandParameter="OutView"
    Content="出库管理"
    Style="{StaticResource NavButtonStyle}" />
<RadioButton
    Command="{Binding NavChangedCommand}"
    CommandParameter="SupplierView"
    Content="供应商管理"
    Style="{StaticResource NavButtonStyle}" />
<RadioButton
    Command="{Binding NavChangedCommand}"
    CommandParameter="CustomerView"
    Content="顾客管理"
    Style="{StaticResource NavButtonStyle}" />
<RadioButton
    Command="{Binding NavChangedCommand}"
    CommandParameter="EmployeeView"
    Content="员工管理"
    Style="{StaticResource NavButtonStyle}" />
<RadioButton
    Command="{Binding NavChangedCommand}"
    CommandParameter="DepartView"
    Content="部门管理"
    Style="{StaticResource NavButtonStyle}" />
</StackPanel>
</Grid>
</Border>
</Grid>
<Grid Grid.Row="1">
    <StackPanel
        Width="64"
        HorizontalAlignment="Right"
        Panel.ZIndex="1">
        <ui:FloatButton
            Uid="{Binding NavNumber}"
            Command="{Binding NavChangedCommand}"

```

```

        Style="{StaticResource RefreshButtonStyle}" />
    </StackPanel>

    <ContentControl Content="{Binding MainContent}" />
</Grid>
<Border
    Grid.RowSpan="2"
    Width="200"
    HorizontalAlignment="Right"
    Background="#aaa"
    CornerRadius="12"
    Opacity="0.9">
    <Border.Effect>
        <DropShadowEffect
            BlurRadius="12"
            Direction="180"
            Opacity="0.3"
            ShadowDepth="0"
            Color="Gray" />
    </Border.Effect>
    <Border.RenderTransform>
        <TranslateTransform x:Name="tt" X="210" />
    </Border.RenderTransform>
</Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="140" />
        <RowDefinition />
    </Grid.RowDefinitions>
    <Button
        Name="btnCloseUserInfo"
        Margin="8"
        HorizontalAlignment="Right"
        VerticalAlignment="Top"
        Content="⌵"
        Style="{StaticResource WindowControlButtonStyle}" />

    <Border
        Width="64"
        Height="64"
        CornerRadius="32">
        <Border.Effect>
            <DropShadowEffect
                BlurRadius="12"
                Direction="0"

```



```

        Opacity="0.3"
        ShadowDepth="0"
        Color="White" />
    </Border.Effect>
    <Border.Background>
        <ImageBrush ImageSource="{Binding LoginModel.Logo}" />
    </Border.Background>

</Border>

<Grid Grid.Row="1" Margin="20,0">
    <Grid.RowDefinitions>
        <RowDefinition Height="48" />
        <RowDefinition Height="48" />
    </Grid.RowDefinitions>
    <StackPanel
        Grid.Row="0"
        VerticalAlignment="Center"
        Orientation="Horizontal">
        <TextBlock
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            FontSize="14"
            Foreground="White"
            Text="员工名" />
        <TextBlock
            Margin="10,0,0,0"
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            FontSize="14"
            Foreground="White"
            Text="{Binding EmployeeModel.EmployeeName}" />
    </StackPanel>
    <StackPanel
        Grid.Row="1"
        VerticalAlignment="Center"
        Orientation="Horizontal">
        <TextBlock
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            FontSize="14"
            Foreground="White"
            Text="性 别" />
        <RadioButton

```

```

        Margin="10,0,0,0"
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Content=" ♂ "
        FontSize="14"
        Foreground="White"
        IsChecked="{Binding EmployeeModel.EmployeeSex,
Converter={StaticResource boolToIntConverter}, ConverterParameter=1}"
        Template="{StaticResource SexRadioButtonTemplate}" />
<RadioButton
    Margin="10,0,0,0"
    HorizontalAlignment="Center"
    VerticalAlignment="Center"
    Content=" ♀ "
    FontSize="14"
    Foreground="White"
    IsChecked="{Binding EmployeeModel.EmployeeSex,
Converter={StaticResource boolToIntConverter}, ConverterParameter=0}"
    Template="{StaticResource SexRadioButtonTemplate}" />
</StackPanel>
</Grid>
</Grid>
</Border>
</Grid>
</Border>
</Window>

```

2.6.7 OrderView.xaml

```

<UserControl
    x:Class="wms.View.OrderView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:ui="clr-namespace:wms.Controls;assembly=wms.Controls"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">
    <UserControl.DataContext>

```

```

        <local:OrderViewModel />
    </UserControl.DataContext>
    <UserControl.Resources>
        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
    </UserControl.Resources>
    <Grid Margin="12">
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="1*" />
                <ColumnDefinition Width="2*" />
            </Grid.ColumnDefinitions>

            <Grid Grid.Column="0">
                <Grid.RowDefinitions>
                    <RowDefinition Height="36" />
                    <RowDefinition />
                </Grid.RowDefinitions>
                <Border
                    Grid.Row="0"
                    Background="#37DC8A"
                    CornerRadius="12,12,0,0">
                    <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="订单信
息" />
                </Border>
                <Border
                    Grid.Row="1"
                    Background="#fff"
                    CornerRadius="0,0,12,12">
                    <ScrollView VerticalScrollBarVisibility="Hidden">
                        <Grid Margin="18">
                            <Grid.RowDefinitions>
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                            </Grid.RowDefinitions>
                            <StackPanel Grid.Row="0" VerticalAlignment="Center">
                                <Grid>

```

```

        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition Width="150" />
        </Grid.ColumnDefinitions>
        <TextBlock
            RowTextBlockStyle}" Text="订单 ID" />
            Style="{StaticResource

        <TextBox
            Grid.Column="2"
            Width="150"
            HorizontalAlignment="Right"
            Template="{StaticResource TextBoxTemplate}"
            Text="{Binding DocumentModel.DocumentId}"
        />

    </Grid>
</StackPanel>
<StackPanel Grid.Row="1" VerticalAlignment="Center">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition Width="150" />
        </Grid.ColumnDefinitions>
        <TextBlock
            RowTextBlockStyle}" Text="顾客 ID" />
            Style="{StaticResource

        <TextBlock
            Grid.Column="2"
            Width="150"
            TextAlignment="Center"
            VerticalAlignment="Center"
            Text="{Binding CustomerModel.CustomerId,
            Mode=TwoWay}" />

    </Grid>
</StackPanel>
<StackPanel Grid.Row="2" VerticalAlignment="Center">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />

            <ColumnDefinition Width="150" />
        </Grid.ColumnDefinitions>
        <TextBlock
            RowTextBlockStyle}" Text="顾客" />
            Style="{StaticResource

        <ComboBox
            Grid.Column="2"
            Width="150"

```

```

HorizontalAlignment="Right"
DisplayMemberPath="customer_name"
ItemsSource="{Binding
OrderDataSet.Tables[2]}"

SelectedIndex="0"
SelectedValue="{Binding
CustomerModel.CustomerId}"

SelectedValuePath="customer_id" />
</Grid>
</StackPanel>
<StackPanel Grid.Row="3" VerticalAlignment="Center">
<Grid>
<Grid.ColumnDefinitions>
<ColumnDefinition />
<ColumnDefinition Width="100" />
</Grid.ColumnDefinitions>
<TextBlock
RowTextBlockStyle" Text="顾客信息版本号" />
<TextBox
Grid.Column="2"
Width="100"
HorizontalAlignment="Right"
Template="{StaticResource TextBoxTemplate}"
Text="{Binding CustomerModel.CustomerInfoId,
Mode=TwoWay}" />
</Grid>
</StackPanel>
<StackPanel Grid.Row="4" VerticalAlignment="Center">
<Grid>
<Grid.ColumnDefinitions>
<ColumnDefinition />

<ColumnDefinition Width="150" />
</Grid.ColumnDefinitions>
<TextBlock
RowTextBlockStyle" Text="订单时间" />
<ui:ucDateTimePicker
Grid.Column="1"
Width="150"
HorizontalAlignment="Right"
SelectedDate="{Binding
DocumentModel.DocumentDate, Mode=TwoWay}" />

</Grid>

```

```

</StackPanel>
<StackPanel Grid.Row="5" VerticalAlignment="Center">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition Width="75" />
            <ColumnDefinition Width="75" />
        </Grid.ColumnDefinitions>
        <TextBlock
            Grid.Column="0"
            Style="{StaticResource RowTextBlockStyle}"
            Text="状态" />
        <StackPanel
            Grid.Column="1"
            HorizontalAlignment="Center"
            VerticalAlignment="Center">
            <RadioButton
                Content="已完成"
                FontSize="14"
                IsChecked="{Binding
OrderModel.OrderStatus, Converter={StaticResource boolToIntConverter},
ConverterParameter=1}"
                Template="{StaticResource
SexRadioButtonTemplate}" />
            </StackPanel>
        <StackPanel
            Grid.Column="2"
            HorizontalAlignment="Center"
            VerticalAlignment="Center">
            <RadioButton
                Content="未完成"
                FontSize="14"
                IsChecked="{Binding
OrderModel.OrderStatus, Converter={StaticResource boolToIntConverter},
ConverterParameter=0}"
                Template="{StaticResource
SexRadioButtonTemplate}" />
            </StackPanel>
        </Grid>
    </StackPanel>
</Grid>
</ScrollView>
</Border>

```

```

</Grid>
<Grid Grid.Column="1" Margin="24,0,0,0">
    <Grid.RowDefinitions>
        <RowDefinition Height="36" />
        <RowDefinition />
    </Grid.RowDefinitions>
    <Border
        Grid.Row="0"
        Background="#37DC8A"
        CornerRadius="12,12,0,0">
        <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="订单详
情" />
    </Border>
    <Border
        Grid.Row="1"
        Background="#37DC8A"
        CornerRadius="0,0,12,12">
        <DataGrid
            Margin="0,0,0,12"
            Background="#37dc8a"
            BorderThickness="0"
            CellStyle="{StaticResource CellStyle}"
            ColumnHeaderStyle="{StaticResource ColumnHeaderStyle}"
            FontSize="14"
            ItemsSource="{Binding OrderDataSet.Tables[1]}"
            RowHeaderWidth="0"
            RowStyle="{StaticResource RowStyle}" />
        </Border>
    </Grid>
</Grid>
<Grid Grid.Row="1" Margin="0,24,0,0">
    <StackPanel
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Orientation="Horizontal">
        <Button
            Command="{Binding FirstPageCommand}"
            Content="|<"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding PrePageCommand}"
            Content="<"
            Style="{StaticResource OperateButtonStyle}" />
    </StackPanel>
</Grid>

```

```

        <Button
            Command="{Binding NextPageCommand}"
            Content=">"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding LastPageCommand}"
            Content=">|"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding InsertCommand}"
            Content="新增"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding DeleteCommand}"
            Content="删除"
            Style="{StaticResource OperateButtonStyle}" />
        <Button
            Command="{Binding SaveCommand}"
            Content="保存"
            Style="{StaticResource OperateButtonStyle}" />
    </StackPanel>

    </Grid>
</Grid>
</UserControl>

```

2.6.8 OutView.xaml

```

<UserControl
    x:Class="wms.View.OutView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

    xmlns:ui="clr-namespace:wms.Controls;assembly=wms.Controls"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">
    <UserControl.DataContext>

```



```

        <local:OutViewModel />
    </UserControl.DataContext>
    <UserControl.Resources>
        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
    </UserControl.Resources>
    <Grid Margin="12">
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="1*" />
                <ColumnDefinition Width="2*" />
            </Grid.ColumnDefinitions>

            <Grid Grid.Column="0">
                <Grid.RowDefinitions>
                    <RowDefinition Height="36" />
                    <RowDefinition />
                </Grid.RowDefinitions>
                <Border
                    Grid.Row="0"
                    Background="#37DC8A"
                    CornerRadius="12,12,0,0">
                    <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="出库单
信息" />
                </Border>
                <Border
                    Grid.Row="1"
                    Background="#fff"
                    CornerRadius="0,0,12,12">
                    <ScrollView VerticalScrollBarVisibility="Hidden">
                        <Grid Margin="18">
                            <Grid.RowDefinitions>
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                                <RowDefinition Height="48" />
                            </Grid.RowDefinitions>
                            <StackPanel Grid.Row="0" VerticalAlignment="Center">
                                <Grid>

```

```

        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition Width="150" />
        </Grid.ColumnDefinitions>
        <TextBlock
            RowTextBlockStyle}" Text="出库单 ID" />
            Style="{StaticResource
        />
        <TextBox
            Grid.Column="2"
            Width="150"
            HorizontalAlignment="Right"
            Template="{StaticResource TextBoxTemplate}"
            Text="{Binding DocumentModel.DocumentId}"
        />
    </Grid>
</StackPanel>

<StackPanel Grid.Row="1" VerticalAlignment="Center">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition Width="150" />
        </Grid.ColumnDefinitions>
        <TextBlock
            RowTextBlockStyle}" Text="顾客 ID" />
            Style="{StaticResource
        />
        <TextBlock
            Grid.Column="2"
            Width="150"
            TextAlignment="Center"
            VerticalAlignment="Center"
            Text="{Binding CustomerModel.CustomerId,
        Mode=TwoWay}" />
    </Grid>
</StackPanel>

<StackPanel Grid.Row="2" VerticalAlignment="Center">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition Width="150" />
        </Grid.ColumnDefinitions>
        <TextBlock
            RowTextBlockStyle}" Text="顾客" />
            Style="{StaticResource
        />
        <ComboBox
            Grid.Column="2"

```

```

Width="150"
HorizontalAlignment="Right"
DisplayMemberPath="customer_name"
ItemsSource="{Binding OutDataSet.Tables[2]}"
SelectedIndex="0"
SelectedValue="{Binding
CustomerModel.CustomerId}"
SelectedValuePath="customer_id" />
</Grid>
</StackPanel>

<StackPanel Grid.Row="3" VerticalAlignment="Center">
<Grid>
<Grid.ColumnDefinitions>
<ColumnDefinition />
<ColumnDefinition Width="100" />
</Grid.ColumnDefinitions>
<TextBlock Style="{StaticResource
RowTextBlockStyle}" Text="顾客信息版本号" />
<TextBox
Grid.Column="2"
Width="100"
HorizontalAlignment="Right"
Template="{StaticResource TextBoxTemplate}"
Text="{Binding CustomerModel.CustomerInfoId,
Mode=TwoWay}" />
</Grid>
</StackPanel>
<StackPanel Grid.Row="4" VerticalAlignment="Center">
<Grid>
<Grid.ColumnDefinitions>
<ColumnDefinition />
<ColumnDefinition Width="150" />
</Grid.ColumnDefinitions>
<TextBlock Style="{StaticResource
RowTextBlockStyle}" Text="出库单时间" />
<ui:ucDateTimePicker
Grid.Column="1"
Width="150"
HorizontalAlignment="Right"
SelectedDate="{Binding
DocumentModel.DocumentDate, Mode=TwoWay}" />

```

```

        </Grid>
    </StackPanel>
    <StackPanel Grid.Row="5" VerticalAlignment="Center">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition Width="75" />
                <ColumnDefinition Width="75" />
            </Grid.ColumnDefinitions>
            <TextBlock
                Grid.Column="0"
                Style="{StaticResource RowTextBlockStyle}"
                Text="状态" />
            <StackPanel
                Grid.Column="1"
                HorizontalAlignment="Center"
                VerticalAlignment="Center">
                <RadioButton
                    Content="已完成"
                    FontSize="14"
                    IsChecked="{Binding OutModel.OutStatus,
Converter={StaticResource boolToIntConverter}, ConverterParameter=1}"
                    Template="{StaticResource
SexRadioButtonTemplate}" />
            </StackPanel>
            <StackPanel
                Grid.Column="2"
                HorizontalAlignment="Center"
                VerticalAlignment="Center">
                <RadioButton
                    Content="未完成"
                    FontSize="14"
                    IsChecked="{Binding OutModel.OutStatus,
Converter={StaticResource boolToIntConverter}, ConverterParameter=0}"
                    Template="{StaticResource
SexRadioButtonTemplate}" />
            </StackPanel>
        </Grid>
    </StackPanel>
</Grid>
</ScrollViewer>
</Border>
</Grid>

```

详情" />

```
<Grid Grid.Column="1" Margin="24,0,0,0">
  <Grid.RowDefinitions>
    <RowDefinition Height="36" />
    <RowDefinition />
  </Grid.RowDefinitions>
  <Border
    Grid.Row="0"
    Background="#37DC8A"
    CornerRadius="12,12,0,0">
    <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="出库单
详情" />
  </Border>
  <Border
    Grid.Row="1"
    Background="#37DC8A"
    CornerRadius="0,0,12,12">
    <DataGrid
      Margin="0,0,0,12"
      Background="#37dc8a"
      BorderThickness="0"
      CellStyle="{StaticResource CellStyle}"
      ColumnHeaderStyle="{StaticResource ColumnHeaderStyle}"
      FontSize="14"
      ItemsSource="{Binding OutDataSet.Tables[1]}"
      RowHeaderWidth="0"
      RowStyle="{StaticResource RowStyle}" />
    </Border>
  </Grid>
</Grid>
<Grid Grid.Row="1" Margin="0,24,0,0">
  <StackPanel
    HorizontalAlignment="Center"
    VerticalAlignment="Center"
    Orientation="Horizontal">
    <Button
      Command="{Binding FirstPageCommand}"
      Content="|&lt;"
      Style="{StaticResource OperateButtonStyle}" />
    <Button
      Command="{Binding PrePageCommand}"
      Content="&lt;"
      Style="{StaticResource OperateButtonStyle}" />
    <Button
```

```

        Command="{Binding NextPageCommand}"
        Content="&gt;"
        Style="{StaticResource OperateButtonStyle}" />
    <Button
        Command="{Binding LastPageCommand}"
        Content="&lt;|"
        Style="{StaticResource OperateButtonStyle}" />
    <Button
        Command="{Binding InsertCommand}"
        Content="新增"
        Style="{StaticResource OperateButtonStyle}" />
    <Button
        Command="{Binding DeleteCommand}"
        Content="删除"
        Style="{StaticResource OperateButtonStyle}" />
    <Button
        Command="{Binding SaveCommand}"
        Content="保存"
        Style="{StaticResource OperateButtonStyle}" />
</StackPanel>

</Grid>
</Grid>
</UserControl>

```

2.6.9 ProductView.xaml

```

<UserControl
    x:Class="wms.View.ProductView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">
    <UserControl.DataContext>
        <local:ProductViewModel />
    </UserControl.DataContext>
    <UserControl.Resources>

```

```

        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />

    </UserControl.Resources>
    <Grid Margin="24">
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>

        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>
            <Grid Grid.Row="0" Grid.Column="0">
                <Grid.RowDefinitions>
                    <RowDefinition Height="36" />
                    <RowDefinition />
                </Grid.RowDefinitions>
                <Border
                    Grid.Row="0"
                    Background="#37DC8A"
                    CornerRadius="12, 12, 0, 0">
                    <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="商品
分级代码" />
                </Border>
                <Border
                    Grid.Row="1"
                    Background="#37DC8A"
                    CornerRadius="0, 0, 12, 12">
                    <DataGrid
                        Margin="12, 0, 12, 12"
                        Background="#37dc8a"
                        BorderThickness="0"
                        ColumnHeaderStyle="{StaticResource ColumnHeaderStyle}"
                        RowStyle="{StaticResource RowStyle}"
                        CellStyle="{StaticResource CellStyle}"
                        FontSize="14"
                        ItemsSource="{Binding ProductDataSet.Tables[0]}"
                        RowHeaderWidth="0" />
                    </Border>
                </Grid>
            <Grid Grid.Column="1" HorizontalAlignment="Center">

```

```

<Grid.RowDefinitions>
    <RowDefinition Height="36" />
    <RowDefinition />
</Grid.RowDefinitions>
<Border
    Grid.Row="0"
    Background="#37DC8A"
    CornerRadius="12, 12, 0, 0">
    <StackPanel Orientation="Horizontal">
        <TextBlock
            Width="80"
            Style="{StaticResource ColumnTextBlockStyle}"
            Text="商品 Id" />
        <TextBlock
            Width="140"
            Style="{StaticResource ColumnTextBlockStyle}"
            Text="商品名称" />
        <TextBlock
            Width="80"
            Style="{StaticResource ColumnTextBlockStyle}"
            Text="数量" />
    </StackPanel>
</Border>
<Border
    Grid.Row="1"
    Background="#fff"
    CornerRadius="0, 0, 12, 12">

    <ScrollViewer VerticalScrollBarVisibility="Hidden">
        <ItemsControl Margin="0, 4, 0, 4" ItemsSource="{Binding
StoreList}">

            <ItemsControl.ItemsPanel>
                <ItemsPanelTemplate>
                    <StackPanel />
                </ItemsPanelTemplate>
            </ItemsControl.ItemsPanel>
            <ItemsControl.ItemTemplate>
                <DataTemplate>
                    <StackPanel Orientation="Horizontal">
                        <TextBlock
                            Width="80"
                            Style="{StaticResource RowTextBlockStyle}"
                            Text="{Binding Id}" />
                        <TextBlock

```



```

                                Width="140"
                                Style="{StaticResource RowTextBlockStyle}"
                                Text="{Binding ProductName}" />
                            <TextBlock
                                Width="80"
                                Style="{StaticResource RowTextBlockStyle}"
                                Text="{Binding Amt}" />
                            </StackPanel>
                        </DataTemplate>
                    </ItemsControl.ItemTemplate>
                </ItemsControl>
            </ScrollView>
        </Border>

    </Grid>
</Grid>
<Grid Grid.Row="1" Margin="0, 24, 0, 0">
    <StackPanel
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Orientation="Horizontal">
        <Button
            Command="{Binding SaveCommand}"
            Content="保存"
            Style="{StaticResource OperateButtonStyle}" />
    </StackPanel>

</Grid>
</Grid>
</UserControl>

```

2.6.10 SalesHistoryView.xaml

```

<UserControl
    x:Class="wms.View.SalesHistoryView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    d:DesignHeight="450"

```

```

d:DesignWidth="800"
mc:Ignorable="d">
<UserControl.DataContext>
    <local:SalesHistoryViewModel />
</UserControl.DataContext>
<UserControl.Resources>
    <ResourceDictionary>
        <ResourceDictionary.MergedDictionaries>
            <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
            <ResourceDictionary>
                <ControlTemplate x:Key="OperateButtonTemplate" TargetType="Button">
                    <Border CornerRadius="4">
                        <Border.Background>
                            <LinearGradientBrush StartPoint="0,0" EndPoint="1,0">
                                <GradientStop Offset="0" Color="#39DBD5" />
                                <GradientStop Offset="1" Color="#37DC8A" />
                            </LinearGradientBrush>
                        </Border.Background>
                        <Grid>
                            <Border
                                Name="back"
                                Background="#22FFFFFF"
                                CornerRadius="4"
                                Visibility="Hidden" />
                            <ContentControl
                                HorizontalAlignment="Center"
                                VerticalAlignment="Center"
                                Content="{TemplateBinding Content}"
                                Foreground="{TemplateBinding Foreground}" />
                        </Grid>
                    </Border>
                    <ControlTemplate.Triggers>
                        <Trigger Property="IsMouseOver" Value="True">
                            <Setter TargetName="back" Property="Visibility"
Value="Visible" />
                        </Trigger>
                        <Trigger Property="IsEnabled" Value="False">
                            <Setter TargetName="back" Property="Visibility"
Value="Visible" />
                            <Setter TargetName="back" Property="Background"
Value="#37DC8A" />
                            <Setter Property="Foreground" Value="#AAA" />
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </ResourceDictionary>
        </ResourceDictionary.MergedDictionaries>
    </ResourceDictionary>
</UserControl.Resources>

```

```

        </ControlTemplate>
    </ResourceDictionary>
    </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
</UserControl.Resources>
<Grid Margin="24" HorizontalAlignment="Center">
    <Grid.RowDefinitions>
        <RowDefinition Height="36" />
        <RowDefinition />
    </Grid.RowDefinitions>
    <Border
        Grid.Row="0"
        Background="#37DC8A"
        CornerRadius="12,12,0,0">
        <StackPanel Orientation="Horizontal">
            <TextBlock
                Width="80"
                Style="{StaticResource ColumnTextBlockStyle}"
                Text="年份" />
            <TextBlock
                Width="40"
                Style="{StaticResource ColumnTextBlockStyle}"
                Text="月份" />
            <TextBlock
                Width="40"
                Style="{StaticResource ColumnTextBlockStyle}"
                Text="商品 Id" />
            <TextBlock
                Width="100"
                Style="{StaticResource ColumnTextBlockStyle}"
                Text="商品名" />
            <TextBlock
                Width="100"
                Style="{StaticResource ColumnTextBlockStyle}"
                Text="月销售量" />
            <TextBlock
                Width="100"
                Style="{StaticResource ColumnTextBlockStyle}"
                Text="月销售额" />
            <TextBlock
                Width="100"
                Style="{StaticResource ColumnTextBlockStyle}"
                Text="年销售量" />
            <TextBlock

```

```

        Width="100"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="年销售额" />
    </StackPanel>
</Border>
<Border
    Grid.Row="1"
    Background="#fff"
    CornerRadius="0,0,12,12">
    <ScrollView VerticalScrollBarVisibility="Hidden">
        <ItemsControl Margin="0,4,0,4" ItemsSource="{Binding SalesHistoryList}">
            <ItemsControl.ItemsPanel>
                <ItemsPanelTemplate>
                    <StackPanel />
                </ItemsPanelTemplate>
            </ItemsControl.ItemsPanel>
            <ItemsControl.ItemTemplate>
                <DataTemplate>
                    <StackPanel Orientation="Horizontal">
                        <TextBlock
                            Width="80"
                            Style="{StaticResource RowTextBlockStyle}"
                            Text="{Binding Hyear}" />
                        <TextBlock
                            Width="40"
                            Style="{StaticResource RowTextBlockStyle}"
                            Text="{Binding Hmon}" />
                        <TextBlock
                            Width="40"
                            Style="{StaticResource RowTextBlockStyle}"
                            Text="{Binding Id}" />
                        <TextBlock
                            Width="100"
                            Style="{StaticResource RowTextBlockStyle}"
                            Text="{Binding ProductName}" />
                        <TextBlock
                            Width="100"
                            Style="{StaticResource RowTextBlockStyle}"
                            Text="{Binding Monqty}" />
                        <TextBlock
                            Width="100"
                            Style="{StaticResource RowTextBlockStyle}"
                            Text="{Binding Monsalesroom}" />
                        <TextBlock

```

```

        Width="100"
        Style="{StaticResource RowTextBlockStyle}"
        Text="{Binding Yearqty}" />
    <TextBlock
        Width="100"
        Style="{StaticResource RowTextBlockStyle}"
        Text="{Binding Yearsalesroom}" />
    </StackPanel>
</DataTemplate>
</ItemsControl.ItemTemplate>
</ItemsControl>
</ScrollViewer>
</Border>
</Grid>
</UserControl>

```

2.6.11 SalesView.xaml

```

<UserControl
    x:Class="wms.View.SalesView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">
    <UserControl.DataContext>
        <local:SalesViewModel />
    </UserControl.DataContext>
    <UserControl.Resources>
        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
    </UserControl.Resources>
    <Grid Margin="24" HorizontalAlignment="Center">
        <Grid.RowDefinitions>
            <RowDefinition Height="36" />
            <RowDefinition />
        </Grid.RowDefinitions>
        <Border
            Grid.Row="0"

```

```

Background="#37DC8A"
CornerRadius="12,12,0,0">
<StackPanel Orientation="Horizontal">
    <TextBlock
        Width="120"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="单据号" />
    <TextBlock
        Width="140"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="日期" />
    <TextBlock
        Width="80"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="用户 Id" />
    <TextBlock
        Width="40"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="商品 Id" />
    <TextBlock
        Width="80"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="商品名" />
    <TextBlock
        Width="80"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="价格" />
    <TextBlock
        Width="40"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="数量" />
    <TextBlock
        Width="80"
        Style="{StaticResource ColumnTextBlockStyle}"
        Text="总计" />
</StackPanel>
</Border>
<Border
    Grid.Row="1"
    Background="#fff"
    CornerRadius="0,0,12,12">
    <ScrollViewer VerticalScrollBarVisibility="Hidden">
        <ItemsControl Margin="0,4,0,4" ItemsSource="{Binding SalesList}">
            <ItemsControl.ItemsPanel>

```

```

        <ItemsPanelTemplate>
            <StackPanel />
        </ItemsPanelTemplate>
    </ItemsControl.ItemsPanel>
    <ItemsControl.ItemTemplate>
        <DataTemplate>
            <StackPanel Orientation="Horizontal">
                <TextBlock
                    Width="120"
                    Style="{StaticResource RowTextBlockStyle}"
                    Text="{Binding DocumentId}" />
                <TextBlock
                    Width="140"
                    Style="{StaticResource RowTextBlockStyle}"
                    Text="{Binding DocumentDate}" />
                <TextBlock
                    Width="80"
                    Style="{StaticResource RowTextBlockStyle}"
                    Text="{Binding CustomerId}" />
                <TextBlock
                    Width="40"
                    Style="{StaticResource RowTextBlockStyle}"
                    Text="{Binding Id}" />
                <TextBlock
                    Width="80"
                    Style="{StaticResource RowTextBlockStyle}"
                    Text="{Binding ProductName}" />
                <TextBlock
                    Width="80"
                    Style="{StaticResource RowTextBlockStyle}"
                    Text="{Binding Price}" />
                <TextBlock
                    Width="40"
                    Style="{StaticResource RowTextBlockStyle}"
                    Text="{Binding Quantity}" />
                <TextBlock
                    Width="80"
                    Style="{StaticResource RowTextBlockStyle}"
                    Text="{Binding Total}" />
            </StackPanel>
        </DataTemplate>
    </ItemsControl.ItemTemplate>
</ItemsControl>
</ScrollView>

```

```
        </Border>
    </Grid>
</UserControl>
```

2.6.12 SupplierView.xaml

```
<UserControl
    x:Class="wms.View.SupplierView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:wms.ViewModel"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    d:DesignHeight="450"
    d:DesignWidth="800"
    mc:Ignorable="d">
    <UserControl.DataContext>
        <local:SupplierViewModel />
    </UserControl.DataContext>
    <UserControl.Resources>
        <ResourceDictionary Source="../Assets/Styles/DefaultStyle.xaml" />
    </UserControl.Resources>
    <Grid Margin="24">
        <Grid.RowDefinitions>
            <RowDefinition Height="3*" />
            <RowDefinition Height="auto" />
        </Grid.RowDefinitions>

        <Grid Grid.Row="0" Grid.Column="0">
            <Grid.RowDefinitions>
                <RowDefinition Height="36" />
                <RowDefinition />
            </Grid.RowDefinitions>
            <Border
                Grid.Row="0"
                Background="#37DC8A"
                CornerRadius="12, 12, 0, 0">
                <TextBlock Style="{StaticResource ColumnTextBlockStyle}" Text="供应商信
息" />
            </Border>
        </Grid>
    </Grid>
</UserControl>
```



```

</Border>
<Border
    Grid.Row="1"
    Background="#37DC8A"
    CornerRadius="0,0,12,12">
    <DataGrid
        Margin="12,0,12,12"
        Background="#37dc8a"
        BorderThickness="0"
        CellStyle="{StaticResource CellStyle}"
        ColumnHeaderStyle="{StaticResource ColumnHeaderStyle}"
        FontSize="14"
        ItemsSource="{Binding SupplierDataSet.Tables[0]}"
        RowHeaderWidth="0"
        RowStyle="{StaticResource RowStyle}" />
    </Border>

</Grid>

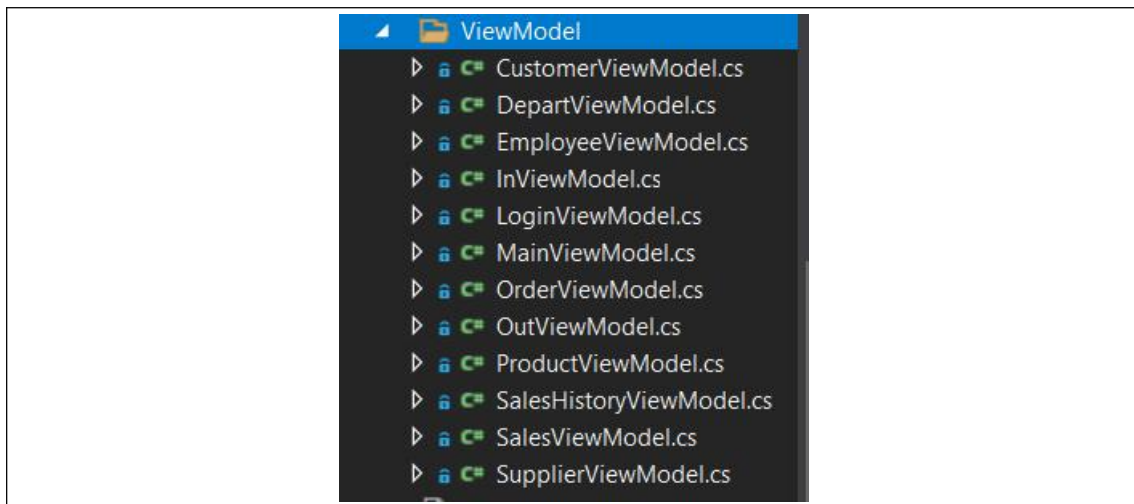
<Grid Grid.Row="1" Margin="0,24,0,0">
    <StackPanel
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Orientation="Horizontal">
        <Button
            Command="{Binding SaveCommand}"
            Content="保存"
            Style="{StaticResource OperateButtonStyle}" />
    </StackPanel>

</Grid>
</Grid>
</UserControl>

```

2.7 wms\ViewModel\

存放逻辑操作类



2.7.1 CustomerViewModel.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using wms.Common;
using wms.DataAccess;
using wms.Model;

namespace wms.ViewModel
{
    public class CustomerViewModel : NotifyBase
    {
        public CustomerModel CustomerModel { get; set; } = new CustomerModel();

        public CommandBase SaveCommand { get; set; }

        private DataSet _supplierDataSet;
        public DataSet CustomerDataSet
        {
            get { return _supplierDataSet; }
            set
            {
                _supplierDataSet = value;
                this.DoNotify();
            }
        }
    }
}
```

```

    }
}

public CustomerViewModel()
{
    //初始化操作
    CustomerDataSet = new DataSet();
    initCustomer();
    //保存命令
    this.SaveCommand = new CommandBase();
    this.SaveCommand.DoExecute = new Action<object>(Save);
    this.SaveCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
}

private void initCustomer()
{
    CustomerDataSet.Tables.Add(ServerDataAccess.GetInstance().GetCustomer());
}

public void Save(object obj)
{
    ServerDataAccess.GetInstance().UpdateCustomer(CustomerDataSet.Tables[0]);
}
}
}

```

2.7.2 DepartViewModel.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using wms.Common;
using wms.DataAccess;
using wms.Model;

namespace wms.ViewModel
{
    public class DepartViewModel : NotifyBase

```

```

{
    public DepartModel DepartModel { get; set; } = new DepartModel();

    public CommandBase SaveCommand { get; set; }

    private DataSet _departDataSet;
    public DataSet DepartDataSet
    {
        get { return _departDataSet; }
        set
        {
            _departDataSet = value;
            this.DoNotify();
        }
    }

    public DepartViewModel()
    {
        //初始化操作
        DepartDataSet = new DataSet();
        initDepart();
        //保存命令
        this.SaveCommand = new CommandBase();
        this.SaveCommand.DoExecute = new Action<object>(Save);
        this.SaveCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    }

    private void initDepart()
    {
        DepartDataSet.Tables.Add(ServerDataAccess.GetInstance().GetDepart());
    }

    public void Save(object obj)
    {
        ServerDataAccess.GetInstance().UpdateDepart(DepartDataSet.Tables[0]);
    }
}

```

2.7.3 EmployeeViewModel.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using wms.Common;
using wms.DataAccess;
using wms.Model;

namespace wms.ViewModel
{
    public class EmployeeViewModel : NotifyBase
    {
        public EmployeeModel EmployeeModel { get; set; } = new EmployeeModel();
        public DepartModel DepartModel { get; set; } = new DepartModel();
        public CommandBase SaveCommand { get; set; }
        public CommandBase FirstPageCommand { get; set; }
        public CommandBase PrePageCommand { get; set; }
        public CommandBase NextPageCommand { get; set; }
        public CommandBase LastPageCommand { get; set; }
        public CommandBase InsertCommand { get; set; }
        public CommandBase DeleteCommand { get; set; }

        private int _position = 0;
        public int Position
        {
            get
            {
                return _position;
            }
            set
            {
                _position = value;
                //处理_position--越界问题, 让 position 指向第一条记录
                if (_position < 0)
                {
                    _position = 0;
                }
                //处理_position++的越界问题, 让_position 指向最后一条记录
                if (_position >= EmployeeDataSet.Tables[0].Rows.Count)
                {
                    _position = EmployeeDataSet.Tables[0].Rows.Count - 1;
                }
                initEmployee(Position); //数据表当前行数据 属性
            }
        }
    }
}
```

```

        initLogin();
    }
}

private DataSet _employeeDataSet;
public DataSet EmployeeDataSet
{
    get { return _employeeDataSet; }
    set
    {
        _employeeDataSet = value;
        this.DoNotify();
    }
}

public EmployeeViewModel()
{
    EmployeeDataSet = new DataSet();
    initEmployee(Position);
    initLogin();
    initDepart();
    //首页命令
    this.FirstPageCommand = new CommandBase();
    this.FirstPageCommand.DoExecute = new Action<object>(FirstPage);
    this.FirstPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    //上一页命令
    this.PrePageCommand = new CommandBase();
    this.PrePageCommand.DoExecute = new Action<object>(PrePage);
    this.PrePageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    //下一页命令
    this.NextPageCommand = new CommandBase();
    this.NextPageCommand.DoExecute = new Action<object>(NextPage);
    this.NextPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    //末页命令
    this.LastPageCommand = new CommandBase();
    this.LastPageCommand.DoExecute = new Action<object>(LastPage);
    this.LastPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    //新增命令
    this.InsertCommand = new CommandBase();
    this.InsertCommand.DoExecute = new Action<object>(Insert);
    this.InsertCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
}

```

```

        //删除命令
        this.DeleteCommand = new CommandBase();
        this.DeleteCommand.DoExecute = new Action<object>(Delete);
        this.DeleteCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

        //保存命令
        this.SaveCommand = new CommandBase();
        this.SaveCommand.DoExecute = new Action<object>(Save);
        this.SaveCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    }

    private void initEmployee(int Position)
    {
        EmployeeDataSet.Tables.Add(ServerDataAccess.GetInstance().GetEmployee());
        EmployeeModel.EmployeeId =
EmployeeDataSet.Tables[0].Rows[Position]["employee_id"].ToString();
        DepartModel.DepartId =
EmployeeDataSet.Tables[0].Rows[Position]["depart_id"].ToString();
        EmployeeModel.EmployeeName =
EmployeeDataSet.Tables[0].Rows[Position]["employee_name"].ToString();
        EmployeeModel.EmployeePhone =
EmployeeDataSet.Tables[0].Rows[Position]["employee_phone"].ToString();
        EmployeeModel.EmployeeSex =
(int)EmployeeDataSet.Tables[0].Rows[Position]["employee_sex"];
        EmployeeModel.EmployeeEducation =
EmployeeDataSet.Tables[0].Rows[Position]["employee_education"].ToString();
        EmployeeModel.EmployeeAddress =
EmployeeDataSet.Tables[0].Rows[Position]["employee_address"].ToString();
        EmployeeModel.EmployeeEmail =
EmployeeDataSet.Tables[0].Rows[Position]["employee_email"].ToString();
        EmployeeModel.EmployeeRemark =
EmployeeDataSet.Tables[0].Rows[Position]["employee_remark"].ToString();

    }

    private void initLogin()
    {
        EmployeeDataSet.Tables.Add(ServerDataAccess.GetInstance().GetLogin());
        EmployeeDataSet.Tables[1].DefaultView.RowFilter = "员工id=' " +
EmployeeModel.EmployeeId + "' ";
    }

    private void initDepart()
    {
        EmployeeDataSet.Tables.Add(ServerDataAccess.GetInstance().GetDepart());
    }

```

```

    }

    public void FirstPage(object obj)
    {
        Position = 0;
    }

    public void PrePage(object obj)
    {
        Position--;
    }

    public void NextPage(object obj)
    {
        Position++;
    }

    public void LastPage(object obj)
    {
        Position = EmployeeDataSet.Tables[0].Rows.Count - 1;
    }

    public void Insert(object obj)
    {
        DataRow tr = EmployeeDataSet.Tables[0].NewRow();
        tr["employee_sex"] = 1;
        EmployeeDataSet.Tables[0].Rows.Add(tr);
        Position = EmployeeDataSet.Tables[0].Rows.Count;
    }

    public void Delete(object obj)
    {
        if (Position > 0)
        {
            EmployeeDataSet.Tables[0].Rows.RemoveAt(Position);
            Position--;
        }
    }

    public void Save(object obj)
    {
        EmployeeDataSet.Tables[0].Rows[Position]["employee_id"] =
EmployeeModel.EmployeeId;
        EmployeeDataSet.Tables[0].Rows[Position]["depart_id"] =
DepartModel.DepartId;
        EmployeeDataSet.Tables[0].Rows[Position]["employee_name"] =
EmployeeModel.EmployeeName;
        EmployeeDataSet.Tables[0].Rows[Position]["employee_phone"] =
EmployeeModel.EmployeePhone;
        EmployeeDataSet.Tables[0].Rows[Position]["employee_sex"] =

```



```

EmployeeModel.EmployeeSex;
        EmployeeDataSet.Tables[0].Rows[Position]["employee_education"] =
EmployeeModel.EmployeeEducation;
        EmployeeDataSet.Tables[0].Rows[Position]["employee_address"] =
EmployeeModel.EmployeeAddress;
        EmployeeDataSet.Tables[0].Rows[Position]["employee_email"] =
EmployeeModel.EmployeeEmail;
        EmployeeDataSet.Tables[0].Rows[Position]["employee_remark"] =
EmployeeModel.EmployeeRemark;
        ServerDataAccess.GetInstance().UpdateEmployee(EmployeeDataSet.Tables[0]);

        ServerDataAccess.GetInstance().UpdateLogin(EmployeeDataSet.Tables[1]);
    }
}
}

```

2.7.4 InViewModel.cs

```

using System;
using System.Data;
using wms.Common;
using wms.DataAccess;
using wms.Model;

namespace wms.ViewModel
{
    public class InViewModel:NotifyBase
    {
        public DocumentModel DocumentModel { get; set; } = new DocumentModel();
        public InModel InModel { get; set; } = new InModel();
        public SupplierModel SupplierModel { get; set; } = new SupplierModel();
        public CommandBase SaveCommand { get; set; }
        public CommandBase FirstPageCommand { get; set; }
        public CommandBase PrePageCommand { get; set; }
        public CommandBase NextPageCommand { get; set; }
        public CommandBase LastPageCommand { get; set; }
        public CommandBase InsertCommand { get; set; }
        public CommandBase DeleteCommand { get; set; }
        private int _position = 0;
        public int Position
        {

```

```

        get
        {
            return _position;
        }
        set
        {
            _position = value;
            //处理_position--越界问题, 让 position 指向第一条记录
            if (_position < 0)
                _position = 0;
            //处理_position++的越界问题, 让_position 指向最后一条记录
            if (_position >= InDataSet.Tables[0].Rows.Count)
                _position = InDataSet.Tables[0].Rows.Count - 1;
            initIn(Position); //数据表当前行数据 属性
            initInInfo();
        }
    }

    private DataSet _inDataSet;
    public DataSet InDataSet
    {
        get { return _inDataSet; }
        set
        {
            _inDataSet = value;
            this.DoNotify();
        }
    }

    private DataSet _documentDataSet;
    public DataSet DocumentDataSet
    {
        get { return _documentDataSet; }
        set
        {
            _documentDataSet = value;
            this.DoNotify();
        }
    }

    public InViewModel()
    {
        //初始化操作
        InDataSet = new DataSet();
        DocumentDataSet = new DataSet();
        initIn(Position);
        initInInfo();
    }

```

```

        initCurrentSupplier();
        //首页命令
        this.FirstPageCommand = new CommandBase();
        this.FirstPageCommand.DoExecute = new Action<object>(FirstPage);
        this.FirstPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

        //上一页命令
        this.PrePageCommand = new CommandBase();
        this.PrePageCommand.DoExecute = new Action<object>(PrePage);
        this.PrePageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

        //下一页命令
        this.NextPageCommand = new CommandBase();
        this.NextPageCommand.DoExecute = new Action<object>(NextPage);
        this.NextPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

        //末页命令
        this.LastPageCommand = new CommandBase();
        this.LastPageCommand.DoExecute = new Action<object>(LastPage);
        this.LastPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

        //新增命令
        this.InsertCommand = new CommandBase();
        this.InsertCommand.DoExecute = new Action<object>(Insert);
        this.InsertCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

        //删除命令
        this.DeleteCommand = new CommandBase();
        this.DeleteCommand.DoExecute = new Action<object>(Delete);
        this.DeleteCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

        //保存命令
        this.SaveCommand = new CommandBase();
        this.SaveCommand.DoExecute = new Action<object>(Save);
        this.SaveCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    }

    private void initCurrentSupplier()
    {
        InDataSet.Tables.Add(ServerDataAccess.GetInstance().GetCurrentSupplier());
    }

    private void initIn(int Position)

```

```

{

    DocumentDataSet.Tables.Add(ServerDataAccess.GetInstance().GetDocumentIn());
    InDataSet.Tables.Add(ServerDataAccess.GetInstance().GetIn());
    DocumentModel.DocumentId =
InDataSet.Tables[0].Rows[Position]["document_id"].ToString();
    SupplierModel.SupplierId =
InDataSet.Tables[0].Rows[Position]["supplier_id"].ToString();
    SupplierModel.SupplierInfoId =
InDataSet.Tables[0].Rows[Position]["supplier_info_id"].ToString();
    DocumentModel.DocumentDate =
(DateTime)InDataSet.Tables[0].Rows[Position]["document_date"];
    InModel.InStatus = (int)InDataSet.Tables[0].Rows[Position]["in_status"];

}

private void initInInfo()
{
    InDataSet.Tables.Add(ServerDataAccess.GetInstance().GetInInfo());

    InDataSet.Tables[1].DefaultView.RowFilter = "单号=' " +
DocumentModel.DocumentId + "' ";
}

public void FirstPage(object obj)
{
    Position = 0;
}

public void PrePage(object obj)
{
    Position--;
}

public void NextPage(object obj)
{
    Position++;
}

public void LastPage(object obj)
{
    Position = InDataSet.Tables[0].Rows.Count - 1;
}

public void Insert(object obj)
{
    DataRow tr = InDataSet.Tables[0].NewRow();
    tr["in_status"] = 0;
    tr["document_date"] = DateTime.Now;
}

```

```

        InDataSet.Tables[0].Rows.Add(tr);
        DocumentDataSet.Tables[0].Rows.Add();
        Position = InDataSet.Tables[0].Rows.Count;
    }

    public void Delete(object obj)
    {
        if (Position > 0)
        {
            InDataSet.Tables[0].Rows[Position].Delete();
            Position--;
        }
    }

    public void Save(object obj)
    {
        DocumentDataSet.Tables[0].Rows[Position]["document_id"] =
        DocumentModel.DocumentId;
        DocumentDataSet.Tables[0].Rows[Position]["document_date"] =
        DocumentModel.DocumentDate;

        InDataSet.Tables[0].Rows[Position]["document_id"] =
        DocumentModel.DocumentId;
        InDataSet.Tables[0].Rows[Position]["supplier_id"] =
        SupplierModel.SupplierId;
        InDataSet.Tables[0].Rows[Position]["supplier_info_id"] =
        SupplierModel.SupplierInfoId;
        InDataSet.Tables[0].Rows[Position]["document_date"] =
        DocumentModel.DocumentDate;
        InDataSet.Tables[0].Rows[Position]["in_status"] = InModel.InStatus;
        //数据库操作
        ServerDataAccess.GetInstance().UpdateDocumentIn(DocumentDataSet.Tables[0]);
        ServerDataAccess.GetInstance().UpdateInInfo(InDataSet.Tables[1]);
        ServerDataAccess.GetInstance().UpdateIn(InDataSet.Tables[0]);
    }
}

```

2.7.5 LoginViewModel.cs

```

using System;
using System.Threading.Tasks;
using System.Windows;

```

```

using System.Windows.Media;
using System.Windows.Media.Imaging;
using wms.Common;
using wms.DataAccess;
using wms.Model;

namespace wms.ViewModel
{
    //用到了除了 Model 里的其他属性
    public class LoginViewModel : NotifyBase
    {
        public LoginModel LoginModel { get; set; } = new LoginModel();
        public CommandBase CloseWindowCommand { get; set; }
        public CommandBase MoveWindowCommand { get; set; }
        public CommandBase LoginCommand { get; set; }
        public CommandBase RefreshVerificationCodeCommand { get; set; }

        //定义 model 以外的属性
        private string validCodeStr;
        private string _verificationCode;

        public string VerificationCode
        {
            {
                get { return _verificationCode; }
                set
                {
                    _verificationCode = value;
                    this.DoNotify();
                }
            }
        }

        private ImageSource _source;
        public ImageSource Source
        {
            {
                get { return _source; }
                set { _source = value; this.DoNotify(); }
            }
        }

        private string _errorMessage;

        public string ErrorMessage
        {
            {
                get { return _errorMessage; }
            }
        }
    }
}

```

```

        set { _errorMessage = value; this.DoNotify(); }
    }

    private Visibility _showProgress = Visibility.Collapsed;

    public Visibility ShowProgress
    {
        get { return _showProgress; }
        set
        {
            _showProgress = value; this.DoNotify();
            LoginCommand.RaiseCanExecuteChanged();
        }
    }

    public LoginViewModel()
    {
        LoginModel.LoginName = "admin";
        LoginModel.LoginPassword = "123456";
        GetImages();
        VerificationCode = validCodeStr;
        //关闭窗口命令
        this.CloseWindowCommand = new CommandBase();
        this.CloseWindowCommand.DoExecute = new Action<object>((o) =>
        {
            (o as Window).Close();
        });
        this.CloseWindowCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //移动窗口命令
        this.MoveWindowCommand = new CommandBase();
        this.MoveWindowCommand.DoExecute = new Action<object>((o) =>
        {
            (o as Window).DragMove();
        });
        this.MoveWindowCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //登录命令
        this.LoginCommand = new CommandBase();
        this.LoginCommand.DoExecute = new Action<object>(DoLogin);
        //登录按钮可用与显示进度条互斥
        this.LoginCommand.DoCanExecute = new Func<object, bool>((o) => { return
ShowProgress == Visibility.Collapsed; });
        //刷新验证码命令

```

```

        this.RefreshVerificationCodeCommand = new CommandBase();
        this.RefreshVerificationCodeCommand.DoExecute = new Action<object>((o) =>
        {
            GetImages();
        });
        this.RefreshVerificationCodeCommand.DoCanExecute = new Func<object, bool>((o)
=> { return true; });
    }

    /// <summary>
    /// 执行登录
    /// </summary>
    /// <param name="o"></param>
    private void DoLogin(object o)
    {
        this.ShowProgress = Visibility.Visible;
        this.ErrorMessage = "";

        if (string.IsNullOrEmpty(LoginModel.LoginName))
        {
            this.ErrorMessage = "请输入用户名";
            this.ShowProgress = Visibility.Collapsed;
            return;
        }
        if (string.IsNullOrEmpty(LoginModel.LoginPassword))
        {
            this.ErrorMessage = "请输入密码";
            this.ShowProgress = Visibility.Collapsed;
            return;
        }
        if (string.IsNullOrEmpty(VerificationCode))
        {
            this.ErrorMessage = "请输入验证码";
            this.ShowProgress = Visibility.Collapsed;
            return;
        }
        if (VerificationCode.ToLower() != validCodeStr)
        {
            this.ErrorMessage = "验证码输入不正确";
            this.ShowProgress = Visibility.Collapsed;
            return;
        }
        //异步多线程控制
        Task.Run(new Action(() =>

```



```

        {
            try
            {
                var userInfo =
ServerDataAccess.GetInstance().CheckUserInfo(LoginModel.LoginName,
LoginModel.LoginPassword);
                if (userInfo == null)
                {
                    throw new Exception("登录失败!用户名或密码错误");
                }

                GlobalValues.UserInfo = userInfo;

                Application.Current.Dispatcher.Invoke(new Action(() =>
                {
                    (o as Window).DialogResult = true;
                }));
            }
            catch (Exception ex)
            {
                this.ErrorMessage = ex.Message;
            }
            finally
            {
                Application.Current.Dispatcher.Invoke(new Action(() =>
                {
                    this.ShowProgress = Visibility.Collapsed;
                }));
            }
        }
    }

    private void GetImages()
    {
        ValidCode validCode = new ValidCode(4, ValidCode.CodeType.Alphas);
        Source = BitmapFrame.Create(validCode.CreateCheckCodeImage());
        validCodeStr = validCode.CheckCode.ToLower();
    }
}

```

2.7.6 MainViewModel.cs

```
using System;
using System.Reflection;
using System.Windows;
using wms.Common;
using wms.DataAccess.DataEntity;
using wms.Model;

namespace wms.ViewModel
{
    public class MainViewModel : NotifyBase
    {
        public EmployeeModel EmployeeModel { get; set; } = new EmployeeModel();
        public LoginModel LoginModel { get; set; } = new LoginModel();
        public UserEntity UserEntity { get; set; } = new UserEntity();
        public CommandBase MinWindowCommand { get; set; }
        public CommandBase ZoomWindowCommand { get; set; }
        public CommandBase CloseWindowCommand { get; set; }
        public CommandBase NavChangedCommand { get; set; }
        public CommandBase MoveWindowCommand { get; set; }
        public CommandBase LogoutCommand { get; set; }

        private FrameworkElement _mainContent;

        public FrameworkElement MainContent
        {
            get { return _mainContent; }
            set { _mainContent = value; this.DoNotify(); }
        }

        private int _isNormal;

        public int IsNormal
        {
            get { return _isNormal; }
            set { _isNormal = value; this.DoNotify(); }
        }

        private int _navNumber;

        public int NavNumber
        {

```

```

        get { return _navNumber; }
        set { _navNumber = value; this.DoNotify(); }
    }

    public MainViewModel()
    {
        //通过全局变量给 UserInfo 赋值
        EmployeeModel.EmployeeSex = GlobalValues.UserInfo.EmployeeSex;
        EmployeeModel.EmployeeName = GlobalValues.UserInfo.EmployeeName;
        LoginModel.Logo = GlobalValues.UserInfo.Logo;
        //通过全局变量给 UserEntity 赋值
        this.UserEntity = GlobalValues.UserInfo;
        //默认窗口状态
        IsNormal = 1;
        //默认子用户控件
        DoNavChanged("SalesView");
        //默认导航条目数
        NavNumber = 1;
        //最小化窗口命令
        this.MinWindowCommand = new CommandBase();
        this.MinWindowCommand.DoExecute = new Action<object>((o) =>
        {
            (o as Window).WindowState = WindowState.Minimized;
        });
        this.MinWindowCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //放大缩小窗口命令
        this.ZoomWindowCommand = new CommandBase();
        this.ZoomWindowCommand.DoExecute = new Action<object>((o) =>
        {
            (o as Window).WindowState = (o as Window).WindowState ==
WindowState.Maximized ?
                WindowState.Normal : WindowState.Maximized;
            IsNormal = (o as Window).WindowState == WindowState.Maximized ? 0 : 1;
        });
        this.ZoomWindowCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //关闭窗口命令
        this.CloseWindowCommand = new CommandBase();
        this.CloseWindowCommand.DoExecute = new Action<object>((o) =>
        {
            (o as Window).Close();
        });
        this.CloseWindowCommand.DoCanExecute = new Func<object, bool>((o) => { return

```

```

true; });

//移动窗口命令
this.MoveWindowCommand = new CommandBase();
this.MoveWindowCommand.DoExecute = new Action<object>((o) =>
{
    (o as Window).DragMove();
});
this.MoveWindowCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

//展示子用户控件
this.NavChangedCommand = new CommandBase();
this.NavChangedCommand.DoExecute = new Action<object>(DoNavChanged);
this.NavChangedCommand.DoCanExecute = new Func<object, bool>((o) => true);
//注销命令
this.LogoutCommand = new CommandBase();
this.LogoutCommand.DoExecute = new Action<object>((o) =>
{
    (o as Window).DialogResult = true;
});
this.LogoutCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
}

/// <summary>
/// 导航栏改变
/// </summary>
/// <param name="obj"></param>
private void DoNavChanged(object obj)
{
    if (obj.ToString().Equals("SalesView"))
    {
        NavNumber = 1;
    }
    if (obj.ToString().Equals("SalesHistoryView"))
    {
        NavNumber = 2;
    }
    if (obj.ToString().Equals("ProductView"))
    {
        NavNumber = 3;
    }
    if (obj.ToString().Equals("InView"))

```

```

        {
            NavNumber = 4;
        }
        if (obj.ToString().Equals("OrderView"))
        {
            NavNumber = 5;
        }
        if (obj.ToString().Equals("OutView"))
        {
            NavNumber = 6;
        }
        if (obj.ToString().Equals("SupplierView"))
        {
            NavNumber = 7;
        }
        if (obj.ToString().Equals("CustomerView"))
        {
            NavNumber = 8;
        }
        if (obj.ToString().Equals("EmployeeView"))
        {
            NavNumber = 9;
        }
        if (obj.ToString().Equals("DepartView"))
        {
            NavNumber = 10;
        }
        Type type = Type.GetType("wms.View." + obj.ToString());
        ConstructorInfo cti = type.GetConstructor(System.Type.EmptyTypes);
        this.MainContent = (FrameworkElement)cti.Invoke(null);
    }

}

```

2.7.7 OrderViewModel.cs

```

using System;
using System.Data;
using wms.Common;
using wms.DataAccess;

```

```

using wms.Model;

namespace wms.ViewModel
{
    public class OrderViewModel : NotifyBase
    {
        public DocumentModel DocumentModel { get; set; } = new DocumentModel();
        public OrderModel OrderModel { get; set; } = new OrderModel();
        public CustomerModel CustomerModel { get; set; } = new CustomerModel();
        public CommandBase SaveCommand { get; set; }
        public CommandBase FirstPageCommand { get; set; }
        public CommandBase PrePageCommand { get; set; }
        public CommandBase NextPageCommand { get; set; }
        public CommandBase LastPageCommand { get; set; }
        public CommandBase InsertCommand { get; set; }
        public CommandBase DeleteCommand { get; set; }
        private int _position = 0;
        public int Position
        {
            get
            {
                return _position;
            }
            set
            {
                _position = value;
                //处理_position一越界问题, 让 position 指向第一条记录
                if (_position < 0)
                {
                    _position = 0;
                }
                //处理_position++的越界问题, 让_position 指向最后一条记录
                if (_position >= OrderDataSet.Tables[0].Rows.Count)
                {
                    _position = OrderDataSet.Tables[0].Rows.Count - 1;
                }
                initOrder(Position); //数据表当前行数据 属性
                initOrderInfo();
            }
        }
        private DataSet _orderDataSet;
        public DataSet OrderDataSet
        {
            get { return _orderDataSet; }
            set
            {
                _orderDataSet = value;
                this.DoNotify();
            }
        }
    }
}

```

```

    }
}

private DataSet _documentDataSet;
public DataSet DocumentDataSet
{
    get { return _documentDataSet; }
    set
    {
        _documentDataSet = value;
        this.DoNotify();
    }
}

public OrderViewModel()
{
    //初始化操作
    OrderDataSet = new DataSet();
    DocumentDataSet = new DataSet();
    initOrder(Position);
    initOrderInfo();
    initCurrentCustomer();
    //首页命令
    this.FirstPageCommand = new CommandBase();
    this.FirstPageCommand.DoExecute = new Action<object>(FirstPage);
    this.FirstPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

    //上一页命令
    this.PrePageCommand = new CommandBase();
    this.PrePageCommand.DoExecute = new Action<object>(PrePage);
    this.PrePageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

    //下一页命令
    this.NextPageCommand = new CommandBase();
    this.NextPageCommand.DoExecute = new Action<object>(NextPage);
    this.NextPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

    //末页命令
    this.LastPageCommand = new CommandBase();
    this.LastPageCommand.DoExecute = new Action<object>(LastPage);
    this.LastPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

    //新增命令
    this.InsertCommand = new CommandBase();
    this.InsertCommand.DoExecute = new Action<object>(Insert);
    this.InsertCommand.DoCanExecute = new Func<object, bool>((o) => { return

```

```

true; });

//删除命令
this.DeleteCommand = new CommandBase();
this.DeleteCommand.DoExecute = new Action<object>(Delete);
this.DeleteCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });

//保存命令
this.SaveCommand = new CommandBase();
this.SaveCommand.DoExecute = new Action<object>(Save);
this.SaveCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
}

private void initCurrentCustomer()
{

OrderDataSet.Tables.Add(ServerDataAccess.GetInstance().GetCurrentCustomer());

}

private void initOrder(int Position)
{

DocumentDataSet.Tables.Add(ServerDataAccess.GetInstance().GetDocumentOrder());
OrderDataSet.Tables.Add(ServerDataAccess.GetInstance().GetOrder());
DocumentModel.DocumentId =
OrderDataSet.Tables[0].Rows[Position]["document_id"].ToString();
CustomerModel.CustomerId =
OrderDataSet.Tables[0].Rows[Position]["customer_id"].ToString();
CustomerModel.CustomerInfoId =
OrderDataSet.Tables[0].Rows[Position]["customer_info_id"].ToString();
DocumentModel.DocumentDate =
(DateTime)OrderDataSet.Tables[0].Rows[Position]["document_date"];
OrderModel.OrderStatus =
(int)OrderDataSet.Tables[0].Rows[Position]["order_status"];

}

private void initOrderInfo()
{

OrderDataSet.Tables.Add(ServerDataAccess.GetInstance().GetOrderInfo());

OrderDataSet.Tables[1].DefaultView.RowFilter = "单号=' " +
DocumentModel.DocumentId + "' ";

}

```



```

public void FirstPage(object obj)
{
    Position = 0;
}

public void PrePage(object obj)
{
    Position--;
}

public void NextPage(object obj)
{
    Position++;
}

public void LastPage(object obj)
{
    Position = OrderDataSet.Tables[0].Rows.Count - 1;
}

public void Insert(object obj)
{
    DataRow tr = OrderDataSet.Tables[0].NewRow();
    tr["order_status"] = 0;
    tr["document_date"] = DateTime.Now;
    OrderDataSet.Tables[0].Rows.Add(tr);
    DocumentDataSet.Tables[0].Rows.Add();
    Position = OrderDataSet.Tables[0].Rows.Count;
}

public void Delete(object obj)
{
    if (Position > 0)
    {
        OrderDataSet.Tables[0].Rows[Position].Delete();
        Position--;
    }
}

public void Save(object obj)
{
    DocumentDataSet.Tables[0].Rows[Position]["document_id"] =
DocumentModel.DocumentId;
    DocumentDataSet.Tables[0].Rows[Position]["document_date"] =
DocumentModel.DocumentDate;

    OrderDataSet.Tables[0].Rows[Position]["document_id"] =
DocumentModel.DocumentId;
    OrderDataSet.Tables[0].Rows[Position]["customer_id"] =

```

```

CustomerModel.CustomerId;
        OrderDataSet.Tables[0].Rows[Position]["customer_info_id"] =
CustomerModel.CustomerInfoId;
        OrderDataSet.Tables[0].Rows[Position]["document_date"] =
DocumentModel.DocumentDate;
        OrderDataSet.Tables[0].Rows[Position]["order_status"] =
OrderModel.OrderStatus;
        //数据库操作

ServerDataAccess.GetInstance().UpdateDocumentOrder(DocumentDataSet.Tables[0]);
        ServerDataAccess.GetInstance().UpdateOrderInfo(OrderDataSet.Tables[1]);
        ServerDataAccess.GetInstance().UpdateOrder(OrderDataSet.Tables[0]);
    }
}
}

```

2.7.8 OutViewModel.cs

```

using System;
using System.Data;
using wms.Common;
using wms.DataAccess;
using wms.Model;

namespace wms.ViewModel
{
    public class OutViewModel : NotifyBase
    {
        public DocumentModel DocumentModel { get; set; } = new DocumentModel();
        public OutModel OutModel { get; set; } = new OutModel();
        public CustomerModel CustomerModel { get; set; } = new CustomerModel();
        public CommandBase SaveCommand { get; set; }
        public CommandBase FirstPageCommand { get; set; }
        public CommandBase PrePageCommand { get; set; }
        public CommandBase NextPageCommand { get; set; }
        public CommandBase LastPageCommand { get; set; }
        public CommandBase InsertCommand { get; set; }
        public CommandBase DeleteCommand { get; set; }
        private int _position = 0;
        public int Position
        {

```

```

        get
        {
            return _position;
        }
        set
        {
            _position = value;
            //处理_position--越界问题, 让 position 指向第一条记录
            if (_position < 0)
                _position = 0;
            //处理_position++的越界问题, 让_position 指向最后一条记录
            if (_position >= OutDataSet.Tables[0].Rows.Count)
                _position = OutDataSet.Tables[0].Rows.Count - 1;
            initOut(Position); //数据表当前行数据 属性
            initOutInfo();
        }
    }

    private DataSet _outDataSet;
    public DataSet OutDataSet
    {
        get { return _outDataSet; }
        set
        {
            _outDataSet = value;
            this.DoNotify();
        }
    }

    private DataSet _documentDataSet;
    public DataSet DocumentDataSet
    {
        get { return _documentDataSet; }
        set
        {
            _documentDataSet = value;
            this.DoNotify();
        }
    }

    public OutViewModel()
    {
        //初始化操作
        OutDataSet = new DataSet();
        DocumentDataSet = new DataSet();
        initOut(Position);
    }

```

```

        initOutInfo();
        initCurrentCustomer();
        //首页命令
        this.FirstPageCommand = new CommandBase();
        this.FirstPageCommand.DoExecute = new Action<object>(FirstPage);
        this.FirstPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //上一页命令
        this.PrePageCommand = new CommandBase();
        this.PrePageCommand.DoExecute = new Action<object>(PrePage);
        this.PrePageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //下一页命令
        this.NextPageCommand = new CommandBase();
        this.NextPageCommand.DoExecute = new Action<object>(NextPage);
        this.NextPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //末页命令
        this.LastPageCommand = new CommandBase();
        this.LastPageCommand.DoExecute = new Action<object>(LastPage);
        this.LastPageCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //新增命令
        this.InsertCommand = new CommandBase();
        this.InsertCommand.DoExecute = new Action<object>(Insert);
        this.InsertCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //删除命令
        this.DeleteCommand = new CommandBase();
        this.DeleteCommand.DoExecute = new Action<object>(Delete);
        this.DeleteCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        //保存命令
        this.SaveCommand = new CommandBase();
        this.SaveCommand.DoExecute = new Action<object>(Save);
        this.SaveCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    }

    private void initCurrentCustomer()
    {
        OutDataSet.Tables.Add(ServerDataAccess.GetInstance().GetCurrentCustomer());
    }

```

```

private void initOut(int Position)
{
    DocumentDataSet.Tables.Add(ServerDataAccess.GetInstance().GetDocumentOut());
    OutDataSet.Tables.Add(ServerDataAccess.GetInstance().GetOut());
    DocumentModel.DocumentId =
    OutDataSet.Tables[0].Rows[Position]["document_id"].ToString();
    CustomerModel.CustomerId =
    OutDataSet.Tables[0].Rows[Position]["customer_id"].ToString();
    CustomerModel.CustomerInfoId =
    OutDataSet.Tables[0].Rows[Position]["customer_info_id"].ToString();
    DocumentModel.DocumentDate =
    (DateTime)OutDataSet.Tables[0].Rows[Position]["document_date"];
    OutModel.OutStatus = (int)OutDataSet.Tables[0].Rows[Position]["out_status"];
}

private void initOutInfo()
{
    OutDataSet.Tables.Add(ServerDataAccess.GetInstance().GetOutInfo());
    OutDataSet.Tables[1].DefaultView.RowFilter = "单号=' " +
    DocumentModel.DocumentId + "' ";
}

public void FirstPage(object obj)
{
    Position = 0;
}

public void PrePage(object obj)
{
    Position--;
}

public void NextPage(object obj)
{
    Position++;
}

public void LastPage(object obj)
{
    Position = OutDataSet.Tables[0].Rows.Count - 1;
}

public void Insert(object obj)
{
    DataRow tr = OutDataSet.Tables[0].NewRow();
    tr["out_status"] = 0;
    tr["document_date"] = DateTime.Now;
}

```

```

        OutDataSet.Tables[0].Rows.Add(tr);
        DocumentDataSet.Tables[0].Rows.Add();
        Position = OutDataSet.Tables[0].Rows.Count;
    }
    public void Delete(object obj)
    {
        if (Position > 0)
        {
            OutDataSet.Tables[0].Rows[Position].Delete();//替换
OutDataSet.Tables[0].Rows.RemoveAt(Position);
            Position--;
        }
    }
    public void Save(object obj)
    {
        DocumentDataSet.Tables[0].Rows[Position]["document_id"] =
DocumentModel.DocumentId;
        DocumentDataSet.Tables[0].Rows[Position]["document_date"] =
DocumentModel.DocumentDate;

        OutDataSet.Tables[0].Rows[Position]["document_id"] =
DocumentModel.DocumentId;
        OutDataSet.Tables[0].Rows[Position]["customer_id"] =
CustomerModel.CustomerId;
        OutDataSet.Tables[0].Rows[Position]["customer_info_id"] =
CustomerModel.CustomerInfoId;
        OutDataSet.Tables[0].Rows[Position]["document_date"] =
DocumentModel.DocumentDate;
        OutDataSet.Tables[0].Rows[Position]["out_status"] = OutModel.OutStatus;

        //数据库操作

        ServerDataAccess.GetInstance().UpdateDocumentOut(DocumentDataSet.Tables[0]);
        ServerDataAccess.GetInstance().UpdateOutInfo(OutDataSet.Tables[1]);
        ServerDataAccess.GetInstance().UpdateOut(OutDataSet.Tables[0]);
    }
}
}

```

2.7.9 ProductViewModel.cs

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Data;
using System.Data.SqlClient;
using System.Threading.Tasks;
using System.Windows;
using wms.Common;
using wms.DataAccess;
using wms.Model;

namespace wms.ViewModel
{
    public class ProductViewModel:NotifyBase
    {
        public ProductModel ProductModel { get; set; } = new ProductModel();
        public ObservableCollection<ProductModel> StoreList { get; set; } = new
ObservableCollection<ProductModel>();
        private List<ProductModel> storeAll;
        public CommandBase SaveCommand { get; set; }

        private DataSet _productDataSet;
        public DataSet ProductDataSet
        {
            get { return _productDataSet; }
            set
            {
                _productDataSet = value;
                this.DoNotify();
            }
        }

        public ProductViewModel()
        {
            //初始化操作
            ProductDataSet = new DataSet();
            initProduct();
            initStore();
            //保存命令
            this.SaveCommand = new CommandBase();
        }
    }
}
```

```

        this.SaveCommand.DoExecute = new Action<object>(Save);
        this.SaveCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
    }

    private void initStore()
    {
        Task.Run(new Action(async () => {
            storeAll = ServerDataAccess.GetInstance().GetStore();
            await Task.Delay(500);
            Application.Current.Dispatcher.Invoke(new Action(() =>
            {
                StoreList.Clear();
                foreach (var item in storeAll)
                    StoreList.Add(item);
            }));
        }));
    }

    private void initProduct()
    {
        ProductDataSet.Tables.Add(ServerDataAccess.GetInstance().GetProduct());
    }

    public void Save(object obj)
    {
        ServerDataAccess.GetInstance().UpdateProduct(ProductDataSet.Tables[0]);
    }
}

```

2.7.10 SalesHistoryViewModel.cs

```

using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Threading.Tasks;
using System.Windows;
using wms.DataAccess;
using wms.Model;

namespace wms.ViewModel

```



```

{
    public class SalesHistoryViewModel
    {
        public ObservableCollection<SalesHistoryModel> SalesHistoryList { get; set; } = new
ObservableCollection<SalesHistoryModel>();
        private List<SalesHistoryModel> salesHistoryAll;
        public SalesHistoryViewModel()
        {
            InitSalesHistoryList();
        }

        private void InitSalesHistoryList()
        {
            /*SalesHistoryList = new
ObservableCollection<SalesHistoryModel>(ServerDataAccess.GetInstance().GetSalesHistory(
));*/

            Task.Run(new Action(async () => {
                salesHistoryAll = ServerDataAccess.GetInstance().GetSalesHistory();
                await Task.Delay(500);
                Application.Current.Dispatcher.Invoke(new Action(() =>
                {
                    SalesHistoryList.Clear();
                    foreach (var item in salesHistoryAll)
                        SalesHistoryList.Add(item);
                }));
            }));
        }
    }
}

```

2.7.11 SalesViewModel.cs

```

using System.Collections.Generic;
using System.Collections.ObjectModel;
using wms.Model;
using wms.DataAccess;
using System.Threading.Tasks;
using System;
using System.Windows;

namespace wms.ViewModel

```

```

{
    public class SalesViewModel
    {
        public ObservableCollection<SalesModel> SalesList { get; set; } = new
ObservableCollection<SalesModel>();
        private List<SalesModel> salesAll;
        public SalesViewModel()
        {
            InitSalesList();
        }

        private void InitSalesList()
        {
            /*SalesList = new
ObservableCollection<SalesModel>(ServerDataAccess.GetInstance().GetSales());*/
            Task.Run(new Action(async () => {
                salesAll = ServerDataAccess.GetInstance().GetSales();
                await Task.Delay(500);
                Application.Current.Dispatcher.Invoke(new Action(() =>
                {
                    SalesList.Clear();
                    foreach (var item in salesAll)
                        SalesList.Add(item);
                }));
            }));
        }
    }
}

```

2.7.12 SupplierViewModel.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using wms.Common;
using wms.DataAccess;
using wms.Model;

```

```

namespace wms.ViewModel
{
    public class SupplierViewModel: NotifyBase
    {
        public SupplierModel SupplierModel { get; set; } = new SupplierModel();

        public CommandBase SaveCommand { get; set; }

        private DataSet _supplierDataSet;
        public DataSet SupplierDataSet
        {
            get { return _supplierDataSet; }
            set
            {
                _supplierDataSet = value;
                this.DoNotify();
            }
        }

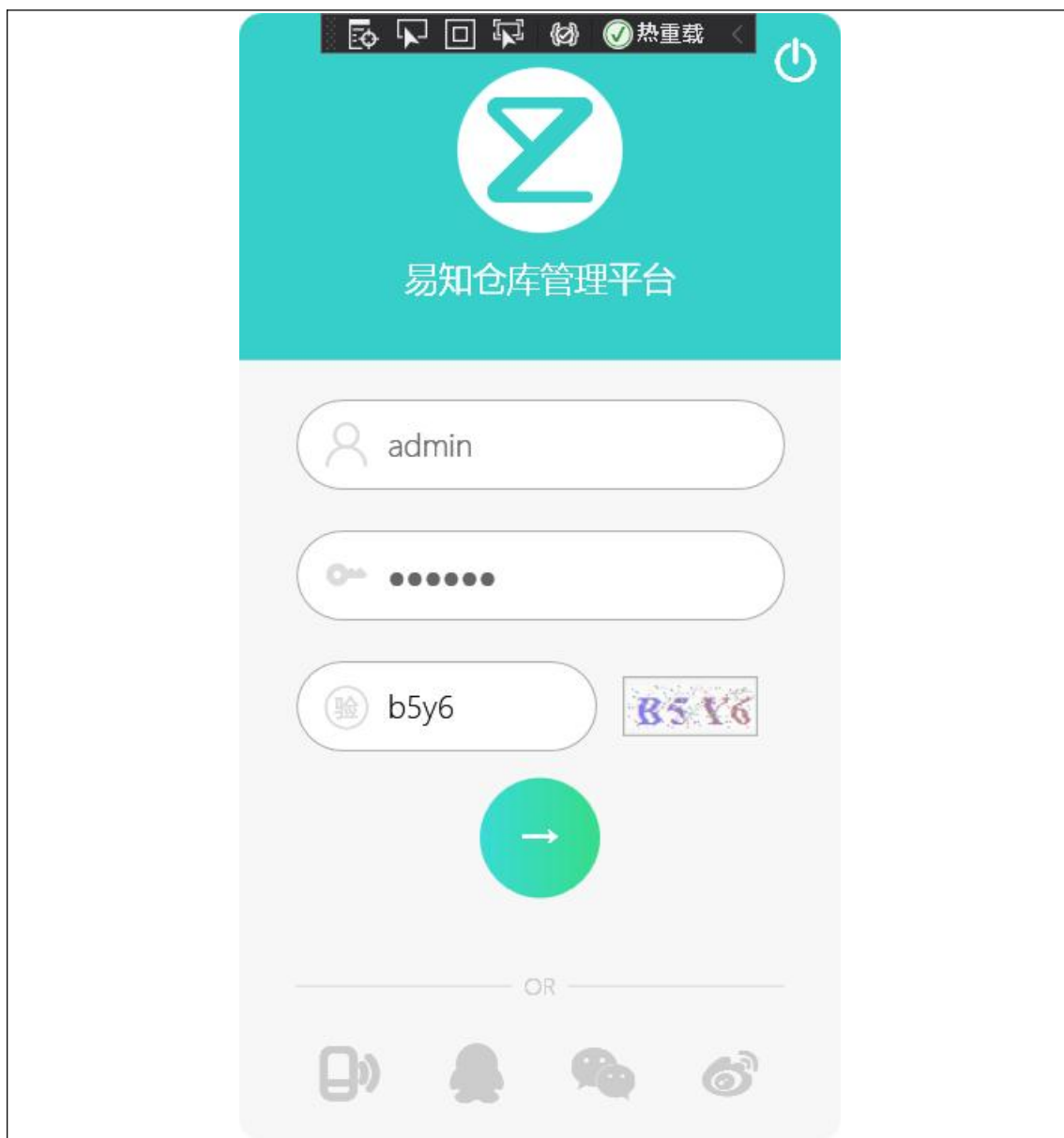
        public SupplierViewModel()
        {
            //初始化操作
            SupplierDataSet = new DataSet();
            initSupplier();
            //保存命令
            this.SaveCommand = new CommandBase();
            this.SaveCommand.DoExecute = new Action<object>(Save);
            this.SaveCommand.DoCanExecute = new Func<object, bool>((o) => { return
true; });
        }

        private void initSupplier()
        {
            SupplierDataSet.Tables.Add(ServerDataAccess.GetInstance().GetSupplier());
        }

        public void Save(object obj)
        {
            ServerDataAccess.GetInstance().UpdateSupplier(SupplierDataSet.Tables[0]);
        }
    }
}

```

四、包含数据的界面截图以及调试过程





易知仓库管理平台

admin

销售管理

历史销售

商品管理

入库管理

订单管理

出库管理

供应商管理

顾客管理

员工管理

部门管理

商品分级代码

id	code	pid	productname
0		0	
1	01	0	食品类
2	01	1	方便食品
3	01	2	方便面
4	02	2	火腿肠
5	03	2	面包
6	02	1	休闲食品
7	01	6	薯片
8	02	6	罐头
9	03	6	果冻

商品Id

商品名称

数量

0		3733
1	食品类	310
2	方便食品	170
3	方便面	100
4	火腿肠	70
5	面包	0
6	休闲食品	140
7	薯片	0
8	罐头	0
9	果脯	140
10	日用品类	960
11	洗化用品	510
12	洗发露	100
13	沐浴露	110

保存

易知仓库管理平台

admin

销售管理

历史销售

商品管理

入库管理

订单管理

出库管理

供应商管理

顾客管理

员工管理

部门管理

入库单信息

入库单ID

IN1000000001

供应商ID

S00002

供应商

捷成食品

供应商信息版本号

01

入库单时间

05/10/2020 12:15 AM

状态

☒ 已完成 ☐ 未完成

入库单详情

商品id	单号	数量	价格	备注
3	IN1000000001	100	10.0000	
4	IN1000000001	160	10.0000	
5	IN1000000001	190	10.0000	
7	IN1000000001	150	10.0000	
8	IN1000000001	150	10.0000	
9	IN1000000001	140	10.0000	

<

<

>

>

新增

删除

保存

145

易知仓库管理平台

admin

销售管理

历史销售

商品管理

入库管理

订单管理

出库管理

供应商管理

顾客管理

员工管理

部门管理

供应商信息

供应商ID	供应商信息版本号	信息变更时间	名称	地址	联系电话	
S00001	01	5/7/2019 10:08:25 PM	保洁日用	哈尔滨道里湘江路国庆大厦36号	18948566268	
S00001	02	5/8/2020 10:08:25 PM	保洁日化	哈尔滨道里湘江路国庆大厦40号	18948566268	
S00001	03	5/9/2021 10:08:25 PM	保洁	哈尔滨道里湘江路国庆大厦41号	18948566200	
S00002	01	5/9/2019 10:08:25 PM	捷成食品	黑龙江伊春	13674583486	
S00003	01	5/9/2019 10:08:25 PM	洗发香水	广州的新路36号	18904588666	
S00004	01	5/9/2019 10:08:25 PM	广州佳美	山东省烟台市新桥大厦6号	13674589874	
S00005	01	5/9/2019 10:08:25 PM	姐妹洗发	上海	18804588666	
S00006	01	5/9/2019 10:08:25 PM	宝洁日化	广州	14785236921	
S00007	01	5/9/2019 10:08:25 PM	季吉乐	深圳市建设镇568号	18475569555	
S00008	01	5/9/2019 10:08:25 PM	合三士日化	上海	18881588555	

保存

顾客信息

顾客ID	顾客信息版本号	信息变更时间	名称	地址	联系电话	
C00001	01	5/7/2019 8:16:38 PM	刘岩	上海市浦东新区金海路2360号	18018594399	
C00001	02	5/8/2020 8:16:39 PM	刘岩	上海市浦东新区金海路10号	18018594399	
C00001	03	5/9/2021 8:16:40 PM	刘博文	上海市浦东新区金海路2360号	18018593333	
C00002	01	5/9/2019 8:16:40 PM	韩黄	河南省鹤壁市淇滨区而能物路想辰春形社区14号	13718195424	
C00003	01	5/9/2019 8:16:40 PM	冯异院	河南省许昌市魏都区知而路形禧社区92号	18273294375	
C00004	01	5/9/2019 8:16:40 PM	张极	河南省周口市鹿邑县所耆身路恒犹流社区40号	15699075013	
C00005	01	5/9/2019 8:16:40 PM	郑角鵠	河南省驻马店市上蔡县其在路远舒社区52号	17272152878	
C00006	01	5/9/2019 8:16:40 PM	郑始通	河南省开封市禹王台区而后路而归社区8号	17516062617	
C00007	01	5/9/2019 8:16:40 PM	尤澳	河南省安阳市林州市之知路其出社区87号	13492128223	
C00008	01	5/9/2019 8:16:40 PM	吕林	河南省焦作市沁阳市在平生路德后德合社区87号	14083788448	

保存

147

易知仓库管理平台

销售管理 历史销售 商品管理 入库管理 订单管理 出库管理 供应商管理 顾客管理 员工管理 部门管理

部门信息

departid departname

D00001	总经理室	
D00002	监督室	
D00003	管理部	
D00004	会计科	
D00005	业务部	
D00006	生管科	
D00007	品保科	
D00008	制造部	
D00009	压合科	
D00010	生管科	

保存

员工名 刘博文

性 别 ☒ 男 ☐ 女

易知仓库管理平台

销售管理 历史销售 商品管理 入库管理 订单管理 出库管理 供应商管理 顾客管理 员工管理 部门管理

员工信息

员工ID

E00001

部门名称

总经理室

员工姓名

刘博文

员工电话

(010)-11111

员工性别

☒ 男 ☐ 女

员工学历

研究生

员工地址

北京市海淀区

员工邮箱

15881464660@qq.com

员工备注

登陆信息

员工id	登录用户名	密码	头像	管理员权限	生效	允许登录
E00001	admin	51A70A1E25F9E6A8954F54D6DF93580D	https://z3.ax1x.com/2021/05/26/25OHRU.jpg	1	1	1