Automative Parts Inventory System

Auto Parts Y&W

Team Members: Wei & Yas

Purpose of the system

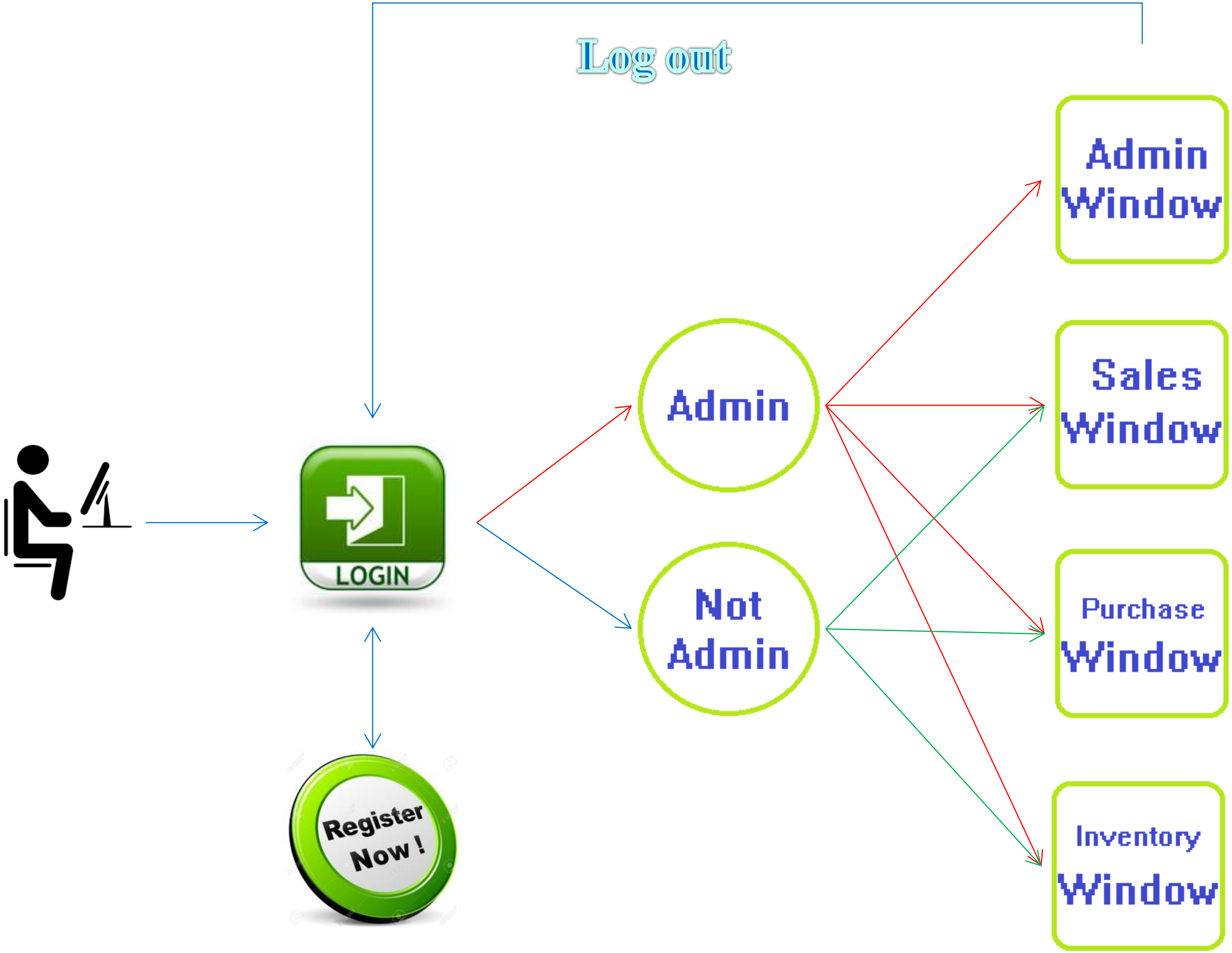
Designing Software that enables user to:

- Keep track of Auto parts sales,
- Place Sales Orders
- Purchase more parts
- Keep track of all Purchases (Inward Shipments)
- Keep track of all Sales (Outward Shipment)
- Monitoring Inventory at all times
- Print Reports

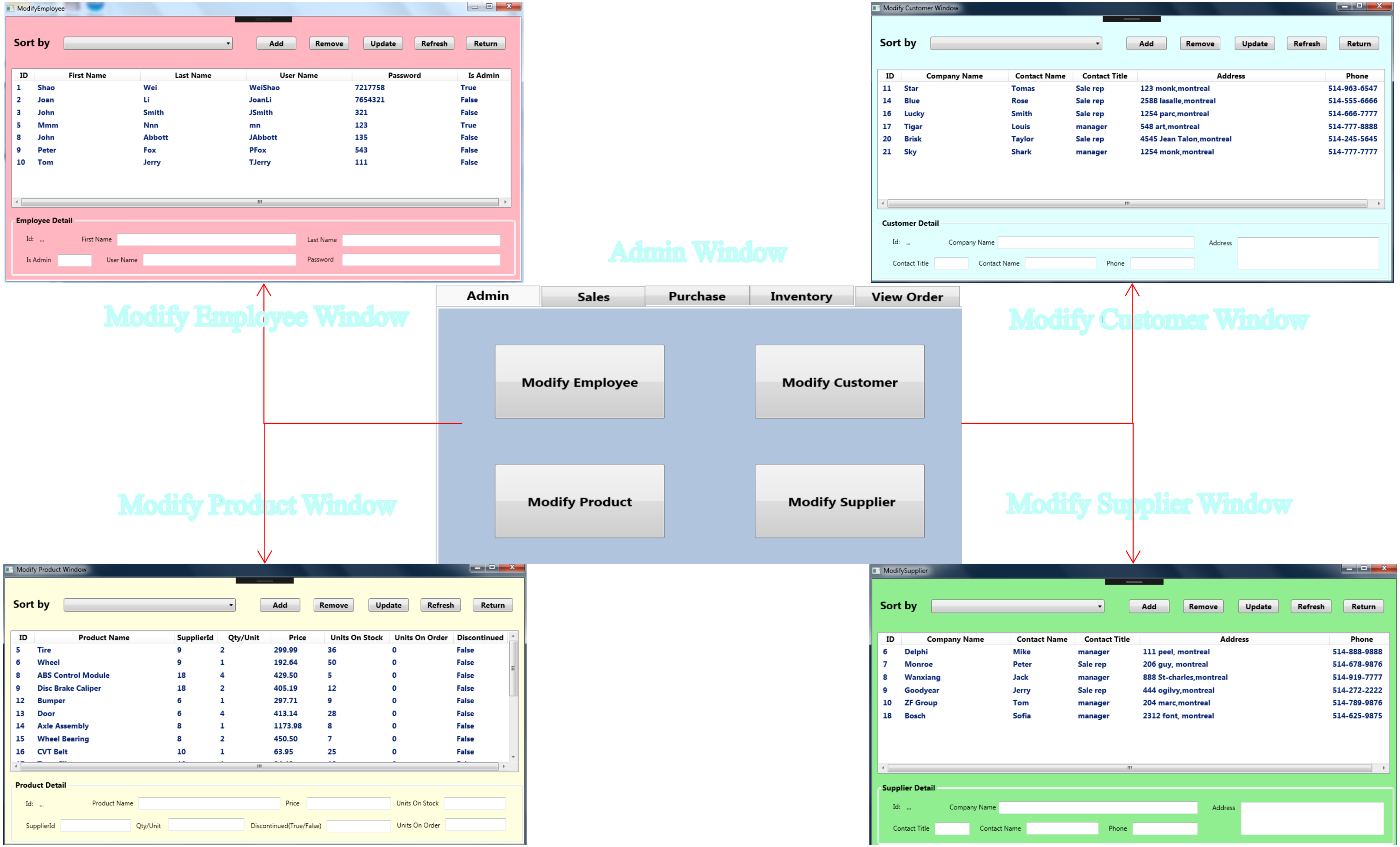
Solution Overview

- Design Simplicity
- User Friendly
- Easy Access
- Maximum Functionality

Solution Overview



Solution Overview



ModifyEmployee

ID	First Name	Last Name	User Name	Password	Is Admin
1	Shao	Wei	WeiShao	7217758	True
2	Joan	Li	JoanLi	7654321	False
3	John	Smith	JSmith	321	False
5	Mmm	Nnn	mn	123	True
8	John	Abbott	JAbbott	135	False
9	Peter	Fox	PFox	543	False
10	Tom	Jerry	TJerry	111	False

Employee Detail

Id: ... First Name Last Name

Is Admin User Name Password

Modify Customer Window

ID	Company Name	Contact Name	Contact Title	Address	Phone
11	Star	Tomas	Sale rep	123 monk,montreal	514-963-6547
14	Blue	Rose	Sale rep	2588 lasalle,montreal	514-555-6666
16	Lucky	Smith	Sale rep	1254 parc,montreal	514-666-7777
17	Tigar	Louis	manager	548 art,montreal	514-777-8888
20	Brisk	Taylor	Sale rep	4545 Jean Talon,montreal	514-245-5645
21	Sky	Shark	manager	1254 monk,montreal	514-777-7777

Customer Detail

Id: ... Company Name Address

Contact Title Contact Name Phone

Modify Product Window

ID	Product Name	SupplierId	Qty/Unit	Price	Units On Stock	Units On Order	Discontinued
5	Tire	9	2	299.99	36	0	False
6	Wheel	9	1	192.64	50	0	False
8	ABS Control Module	18	4	429.50	5	0	False
9	Disc Brake Caliper	18	2	405.19	12	0	False
12	Bumper	6	1	297.71	9	0	False
13	Door	6	4	413.14	28	0	False
14	Axle Assembly	8	1	1173.98	8	0	False
15	Wheel Bearing	8	2	450.50	7	0	False
16	CVT Belt	10	1	63.95	25	0	False

Product Detail

Id: ... Product Name Price Units On Stock

SupplierId Qty/Unit Discontinued(True/False) Units On Order

ModifySupplier

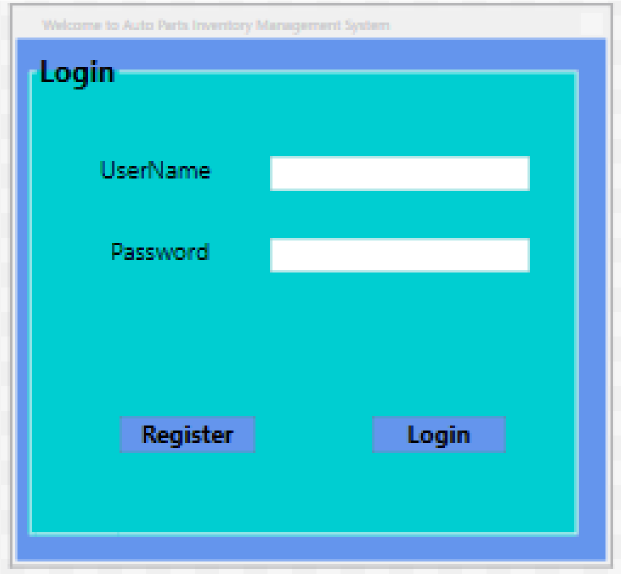
ID	Company Name	Contact Name	Contact Title	Address	Phone
6	Delphi	Mike	manager	111 peel, montreal	514-888-9888
7	Monroe	Peter	Sale rep	206 guy, montreal	514-678-9876
8	Wanxiang	Jack	manager	888 St-charles,montreal	514-919-7777
9	Goodyear	Jerry	Sale rep	444 ogilvy,montreal	514-272-2222
10	ZF Group	Tom	manager	204 marc,montreal	514-789-9876
18	Bosch	Sofia	manager	2312 font, montreal	514-625-9875

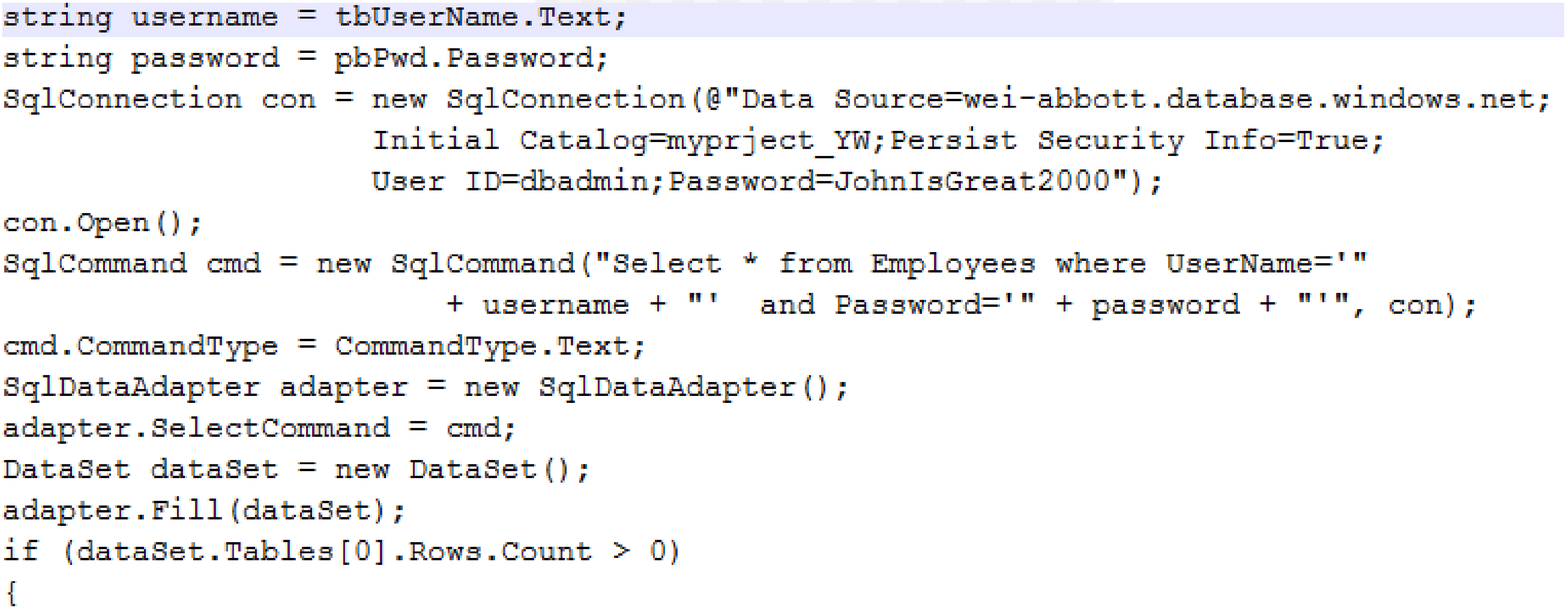
Supplier Detail

Id: ... Company Name Address

Contact Title Contact Name Phone

Login Screen

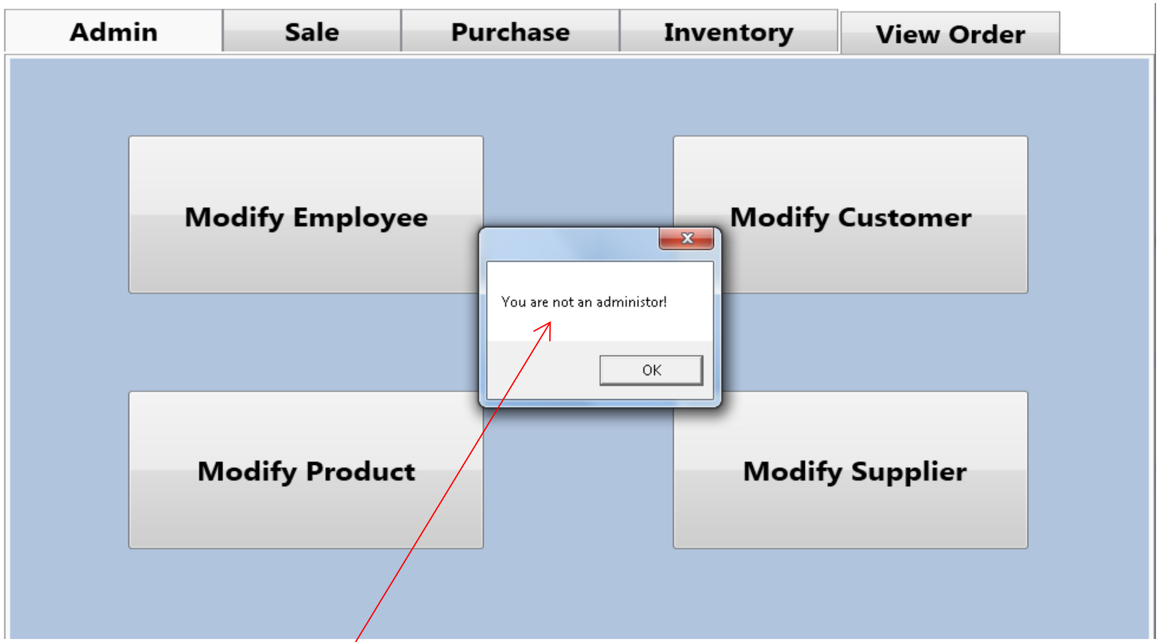




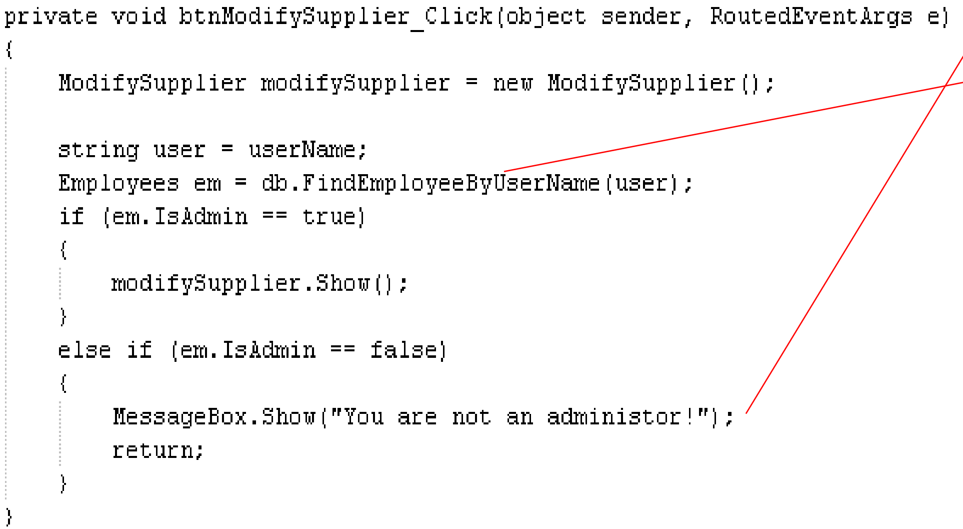
```
string username = tbUserName.Text;
string password = pbPwd.Password;
SqlConnection con = new SqlConnection(@"Data Source=wei-abbott.database.windows.net;
                    Initial Catalog=myprject_YW;Persist Security Info=True;
                    User ID=dbadmin;Password=JohnIsGreat2000");
con.Open();
SqlCommand cmd = new SqlCommand("Select * from Employees where UserName='"
                        + username + "'  and Password='" + password + "'", con);
cmd.CommandType = CommandType.Text;
SqlDataAdapter adapter = new SqlDataAdapter();
adapter.SelectCommand = cmd;
DataSet dataSet = new DataSet();
adapter.Fill(dataSet);
if (dataSet.Tables[0].Rows.Count > 0)
{
```

Challenges and Solutions

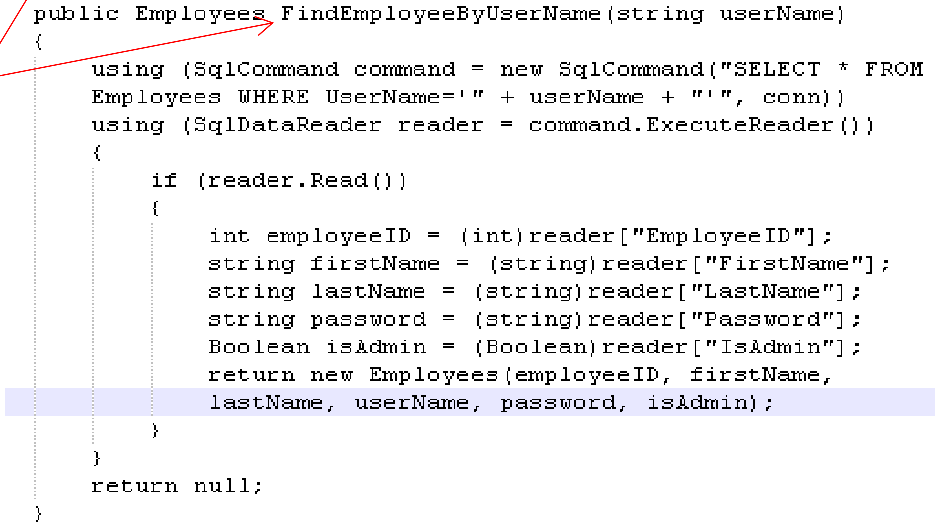
To prevent a non-admin user to access the admin window, some designers make the admin window unactivated, but I show a dialog to achieve it.



You are not an administor!



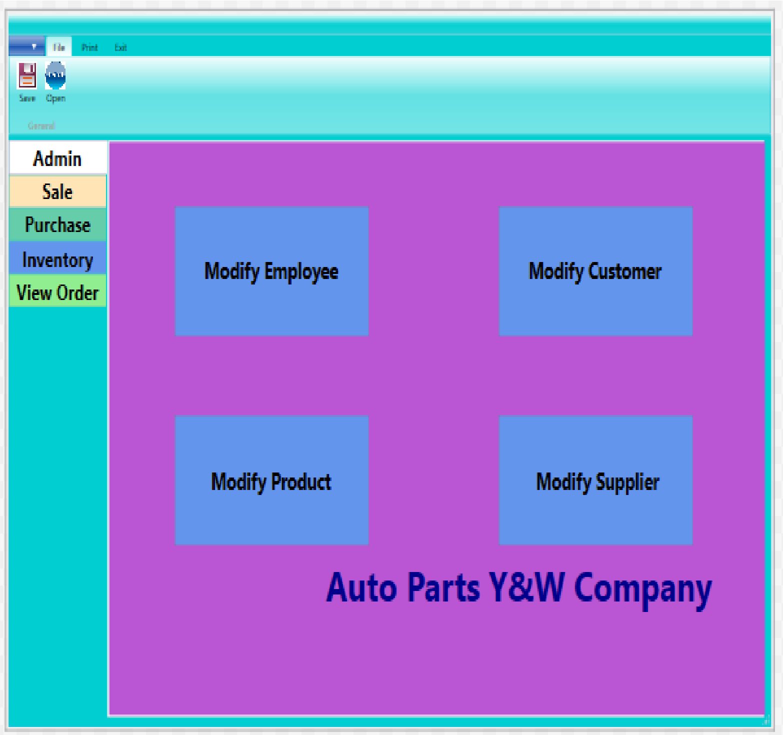
```
private void btnModifySupplier_Click(object sender, RoutedEventArgs e)
{
    ModifySupplier modifySupplier = new ModifySupplier();

    string user = userName;
    Employees em = db.FindEmployeeByUserName(user);
    if (em.IsAdmin == true)
    {
        modifySupplier.Show();
    }
    else if (em.IsAdmin == false)
    {
        MessageBox.Show("You are not an administor!");
        return;
    }
}
```



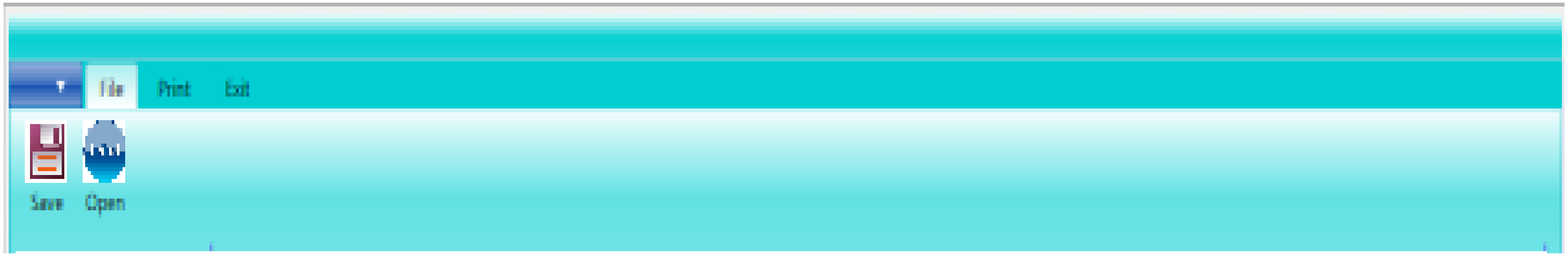
```
public Employees FindEmployeeByUserName(string userName)
{
    using (SqlCommand command = new SqlCommand("SELECT * FROM
    Employees WHERE UserName='" + userName + "'", conn))
    using (SqlDataReader reader = command.ExecuteReader())
    {
        if (reader.Read())
        {
            int employeeID = (int)reader["EmployeeID"];
            string firstName = (string)reader["FirstName"];
            string lastName = (string)reader["LastName"];
            string password = (string)reader["Password"];
            Boolean isAdmin = (Boolean)reader["IsAdmin"];
            return new Employees(employeeID, firstName,
            lastName, userName, password, isAdmin);
        }
    }
    return null;
}
```

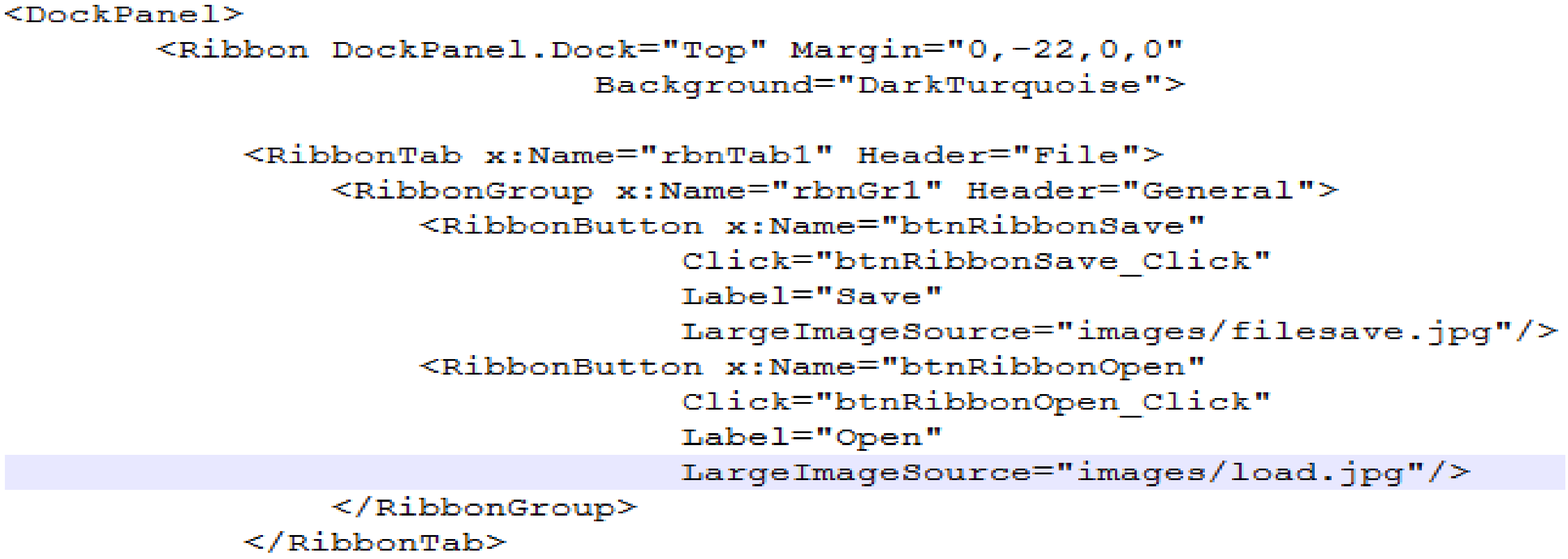
Tab Pages

```
<TabItem Header="Admin" Background="MediumOrchid"
                             HorizontalAlignment="Left"
                     Height="40" VerticalAlignment="Top"
             Width="150" FontSize="24" FontWeight="Bold">
     <Grid>
         <Grid Background="MediumOrchid" Margin="0,0,-54,0">
             <Grid.ColumnDefinitions>
                 <ColumnDefinition Width="243*"/>
                 <ColumnDefinition Width="238*"/>
             </Grid.ColumnDefinitions>
             <Button Content="Modify Employee"
                             Background="CornflowerBlue"
                             HorizontalAlignment="Left"
                             Margin="100,73,0,0"
                             VerticalAlignment="Top" Width="300"
                             Height="150" FontSize="25"
                             FontWeight="Bold"/>
         </Grid>
     </Grid>
 </TabItem>
```

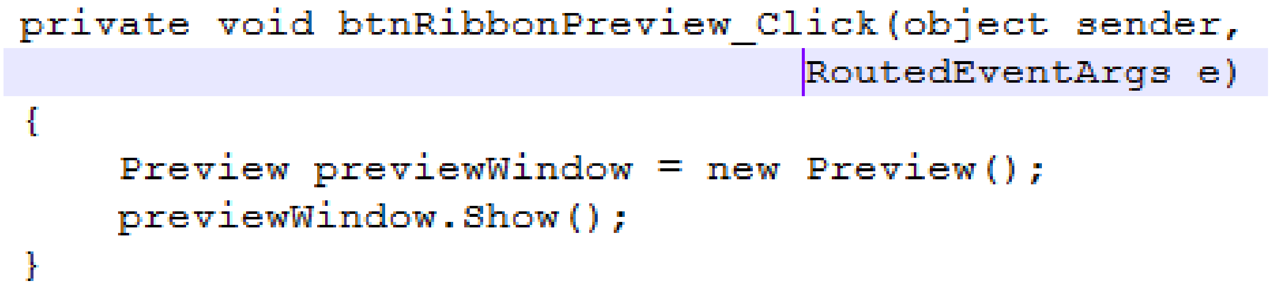


Challenges and Solutions:



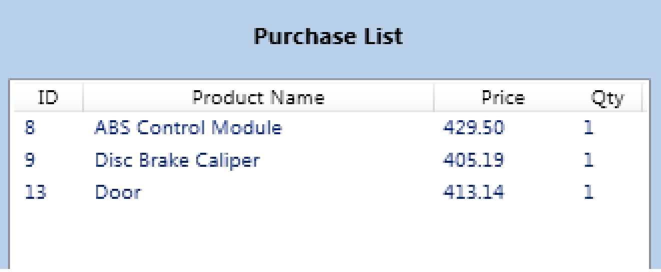


```
<DockPanel>
       <Ribbon DockPanel.Dock="Top" Margin="0,-22,0,0"
                                   Background="DarkTurquoise">

           <RibbonTab x:Name="rbnTab1" Header="File">
               <RibbonGroup x:Name="rbnGr1" Header="General">
                   <RibbonButton x:Name="btnRibbonSave"
                               Click="btnRibbonSave_Click"
                               Label="Save"
                               LargeImageSource="images/filesave.jpg"/>
                   <RibbonButton x:Name="btnRibbonOpen"
                               Click="btnRibbonOpen_Click"
                               Label="Open"
                               LargeImageSource="images/load.jpg"/>
               </RibbonGroup>
           </RibbonTab>
```



```
private void btnRibbonPreview_Click(object sender,
                              RoutedEventArgs e)
{
    Preview previewWindow = new Preview();
    previewWindow.Show();
}
```

Challenges and Solutions



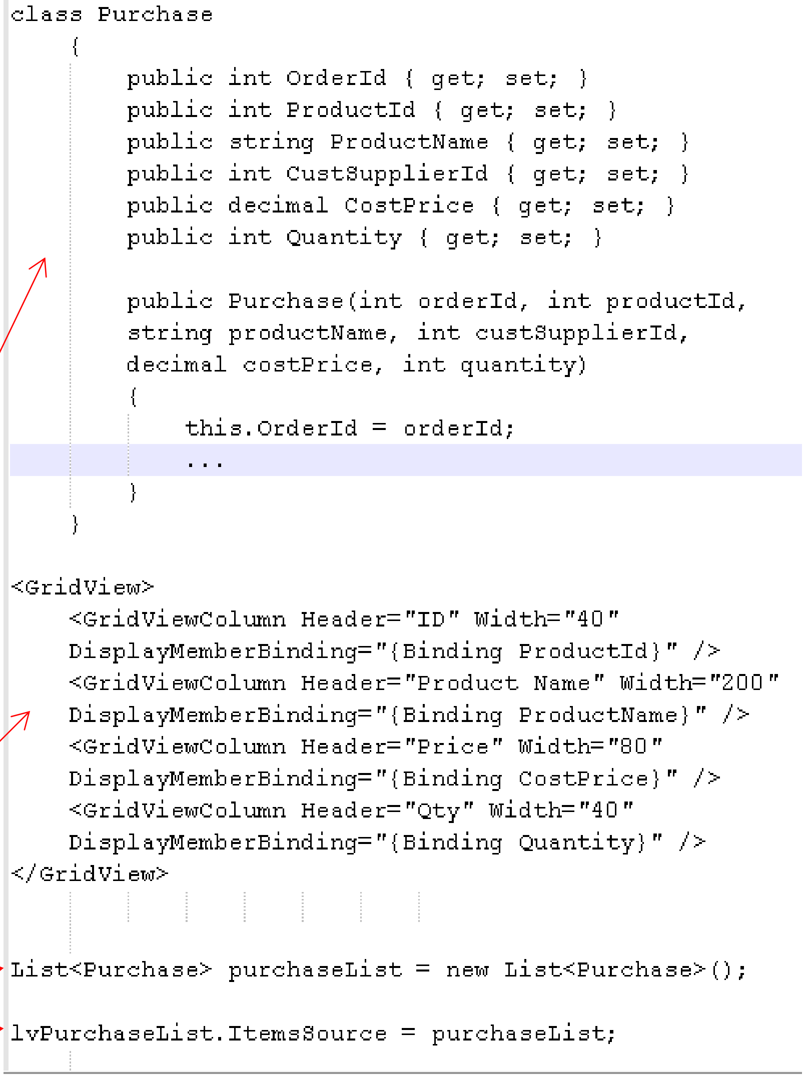
Purchase List

ID	Product Name	Price	Qty
8	ABS Control Module	429.50	1
9	Disc Brake Caliper	405.19	1
13	Door	413.14	1

Binding two tables to one listview.
Table Products: ProductId, ProductName
Table OrderDetails: Price, Quantity

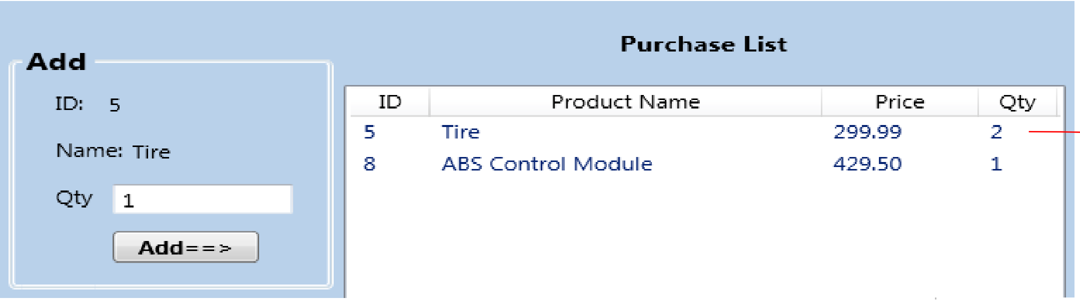
Solution:

- **Create a new object class including ProductId, ProductName, Price, and Quantity;**
- **Don’t need to add one more table in Database;**
- **Binding the new object in xaml file;**
- **Define object class to list in xaml.cs file;**
- **Define listview’s source is the list.**



```
class Purchase
    {
        public int OrderId { get; set; }
        public int ProductId { get; set; }
        public string ProductName { get; set; }
        public int CustSupplierId { get; set; }
        public decimal CostPrice { get; set; }
        public int Quantity { get; set; }

        public Purchase(int orderId, int productId,
        string productName, int custSupplierId,
        decimal costPrice, int quantity)
        {
            this.OrderId = orderId;
            ...
        }
    }

<GridView>
    <GridViewColumn Header="ID" Width="40"
    DisplayMemberBinding="{Binding ProductId}" />
    <GridViewColumn Header="Product Name" Width="200"
    DisplayMemberBinding="{Binding ProductName}" />
    <GridViewColumn Header="Price" Width="80"
    DisplayMemberBinding="{Binding CostPrice}" />
    <GridViewColumn Header="Qty" Width="40"
    DisplayMemberBinding="{Binding Quantity}" />
</GridView>


List<Purchase> purchaseList = new List<Purchase>();

lvPurchaseList.ItemsSource = purchaseList;
```

Challenges and Solutions



Add

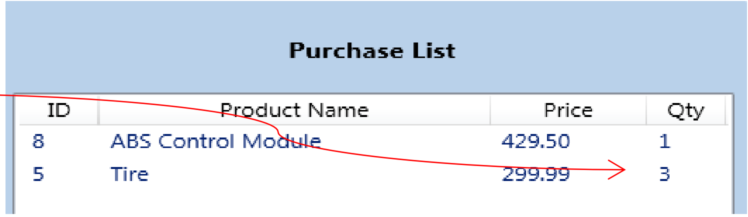
ID: 5

Name: Tire

Qty 1

Purchase List

ID	Product Name	Price	Qty
5	Tire	299.99	2
8	ABS Control Module	429.50	1



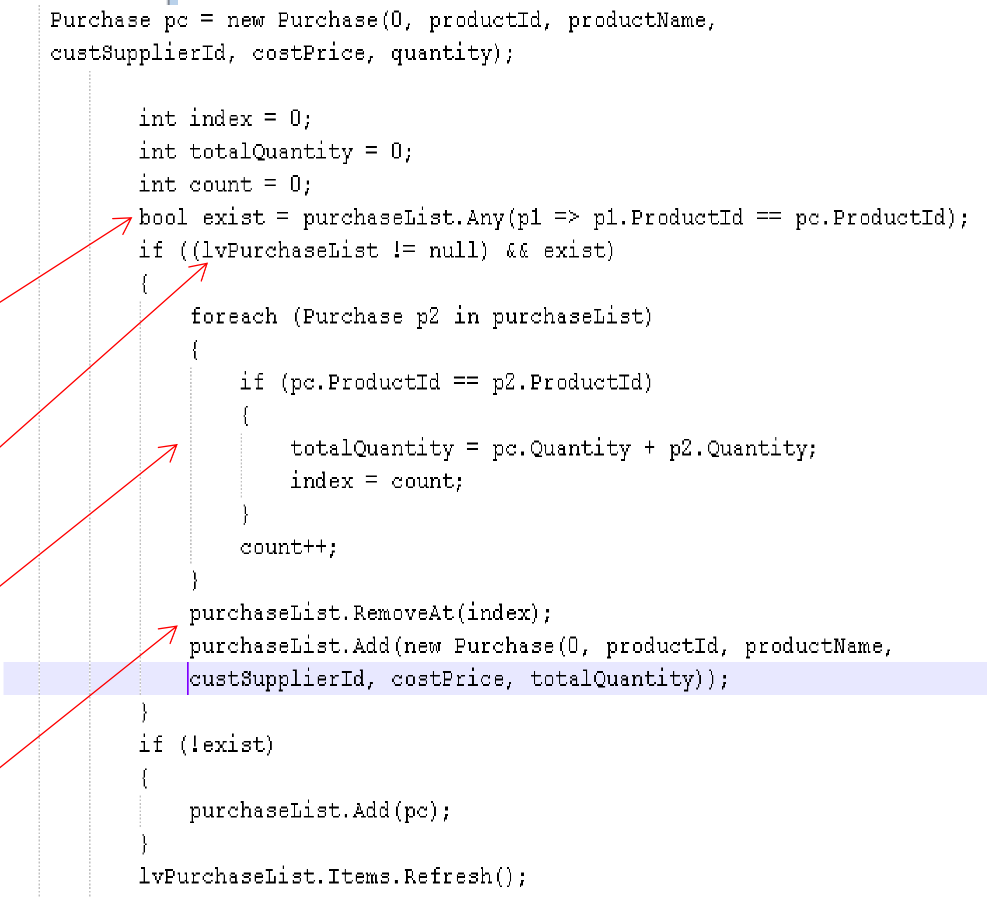
Purchase List

ID	Product Name	Price	Qty
8	ABS Control Module	429.50	1
5	Tire	299.99	3

Adding more than once for the same product from the product listview to the purchase listview.

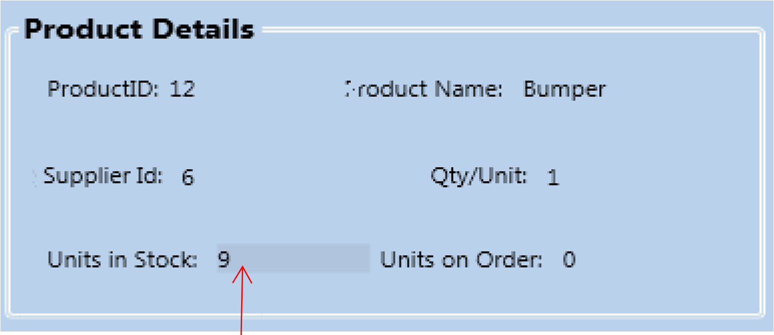
Solution:

- **Define a boolean “exist” to check whether the product Id is exist in the purchase listview;**
- **If exist is true and the purchase listview is not empty, go to the loop of the purchase list;**
- **In the loop, find the index and add the quantity;**
- **Outside the loop, remove the old purchase object at the index, and add the new one with new quantity.**



```
Purchase pc = new Purchase(0, productId, productName,
custSupplierId, costPrice, quantity);

       int index = 0;
       int totalQuantity = 0;
       int count = 0;
       bool exist = purchaseList.Any(p1 => p1.ProductId == pc.ProductId);
       if ((lvPurchaseList != null) && exist)
       {
           foreach (Purchase p2 in purchaseList)
           {
               if (pc.ProductId == p2.ProductId)
               {
                   totalQuantity = pc.Quantity + p2.Quantity;
                   index = count;
               }
               count++;
           }
           purchaseList.RemoveAt(index);
           purchaseList.Add(new Purchase(0, productId, productName,
           custSupplierId, costPrice, totalQuantity));
       }
       if (!exist)
       {
           purchaseList.Add(pc);
       }
       lvPurchaseList.Items.Refresh();
```

Challenges and Solutions

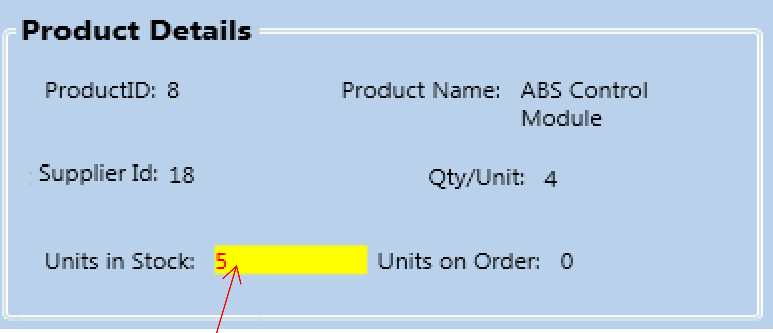


Product Details

ProductID: 12 ⠀ :roduct Name: Bumper

Supplier Id: 6 ⠀ Qty/Unit: 1

Units in Stock: 9 ⠀ Units on Order: 0



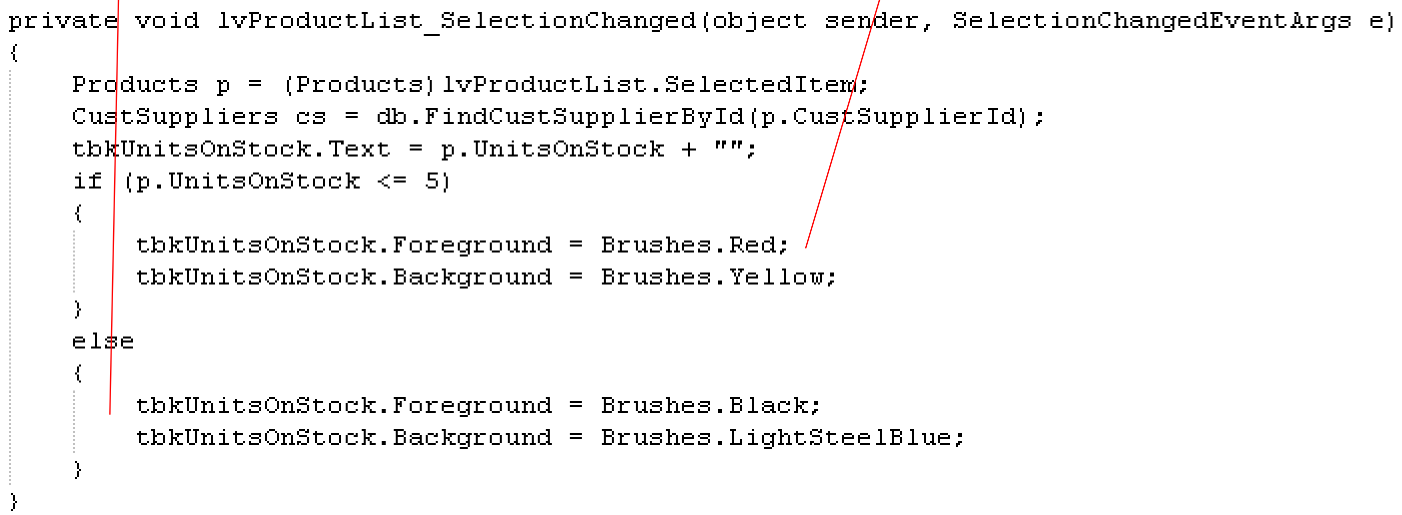
Product Details

ProductID: 8 ⠀ Product Name: ABS Control Module

Supplier Id: 18 ⠀ Qty/Unit: 4

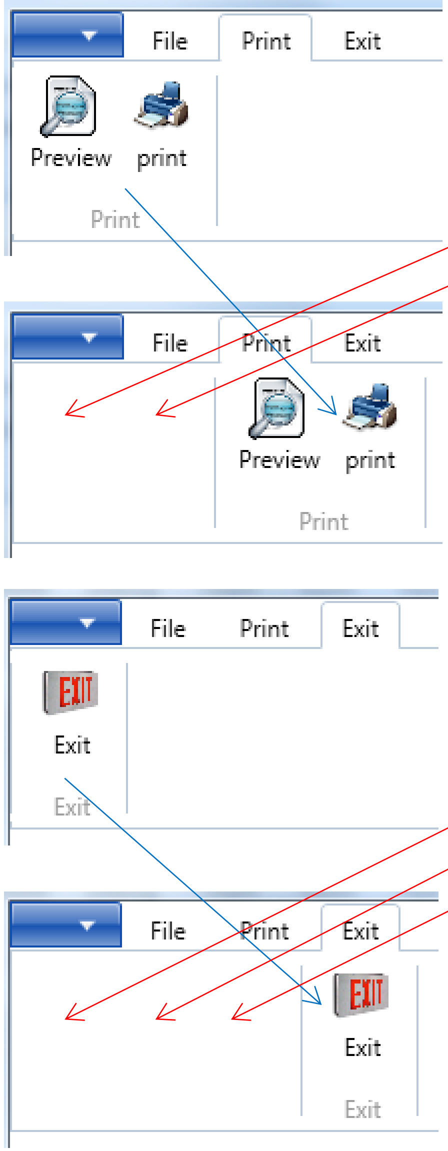
Units in Stock: 5 ⠀ Units on Order: 0

To show the lower quantity in inventory, usually show a dialog to warn user, but I choose another way that it is to highlight the quantity text and the text field .



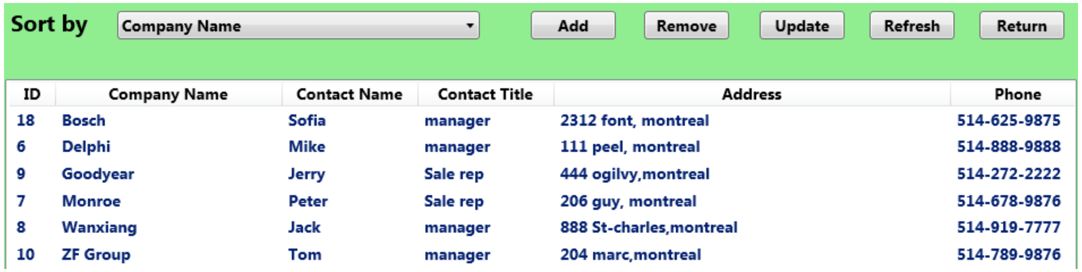
```
private void lvProductList_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    Products p = (Products)lvProductList.SelectedItem;
    CustSuppliers cs = db.FindCustSupplierById(p.CustSupplierId);
    tbkUnitsOnStock.Text = p.UnitsOnStock + "";
    if (p.UnitsOnStock <= 5)
    {
        tbkUnitsOnStock.Foreground = Brushes.Red;
        tbkUnitsOnStock.Background = Brushes.Yellow;
    }
    else
    {
        tbkUnitsOnStock.Foreground = Brushes.Black;
        tbkUnitsOnStock.Background = Brushes.LightSteelBlue;
    }
}
```

What we learned



```
<RibbonTab x:Name="rbnTab4" Header="Print">
    <RibbonGroup Header="">
        <RibbonButton Label=""  LargeImageSource="images/White.png"/>
        <RibbonButton Label=""  LargeImageSource="images/White.png"/>
    </RibbonGroup>

    <RibbonGroup x:Name="rbnGr2" Header="Print">
        <RibbonButton x:Name="btnRibbonPreview" Label="Preview"
        Click="btnRibbonPreview_Click" LargeImageSource="images/preview.ico"/>
        <RibbonButton x:Name="btnRibbonPrint" Label="print"
        Click="btnRibbonPrint_Click" LargeImageSource="images/print.ico"/>
    </RibbonGroup>
</RibbonTab>

<RibbonTab x:Name="rbnTab3" Header="Exit">
    <RibbonGroup Header="">
        <RibbonButton Label=""  LargeImageSource="images/White.png"/>
        <RibbonButton Label=""  LargeImageSource="images/White.png"/>
        <RibbonButton Label=""  LargeImageSource="images/White.png"/>
    </RibbonGroup>

    <RibbonGroup x:Name="rbnGr3" Header="Exit">
        <RibbonButton x:Name="btnRibbonExit" Click="btnRibbonExit_Click"
        Label="Exit" LargeImageSource="images/exit.png"/>
    </RibbonGroup>
</RibbonTab>
```

What we learned

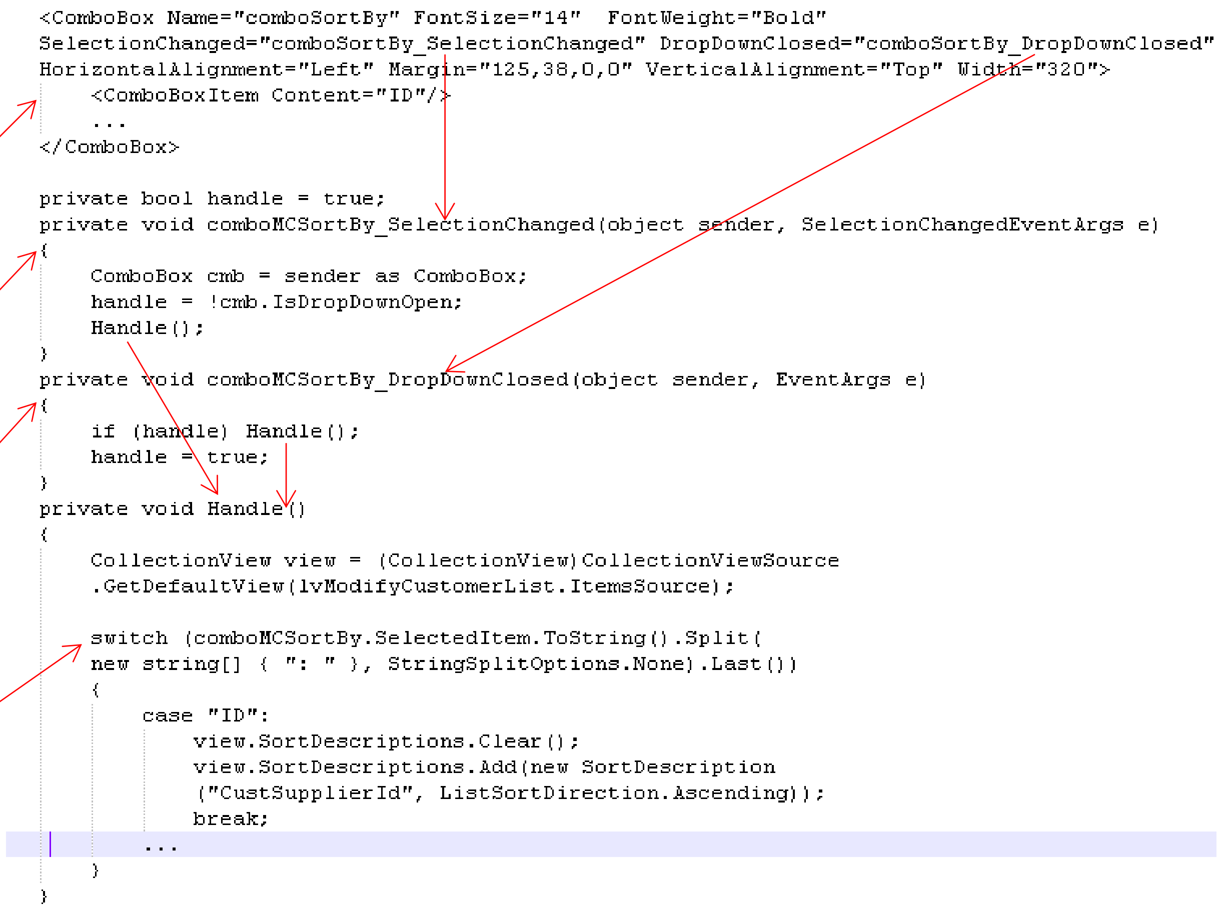


ID	Company Name	Contact Name	Contact Title	Address	Phone
18	Bosch	Sofia	manager	2312 font, montreal	514-625-9875
6	Delphi	Mike	manager	111 peel, montreal	514-888-9888
9	Goodyear	Jerry	Sale rep	444 ogilvy,montreal	514-272-2222
7	Monroe	Peter	Sale rep	206 guy, montreal	514-678-9876
8	Wanxiang	Jack	manager	888 St-charles,montreal	514-919-7777
10	ZF Group	Tom	manager	204 marc,montreal	514-789-9876

To sort the list in the listview.

Solution:

- **Design the combobox code in xaml file, and assign the column’s name you want to sort in the combobox;**
- **Design the selection-changed event in the xaml.cs file;**
- **Design the dropdown-closed event in the xaml.cs file;**
- **By using the switch-case loop function, active the sort function according to different case.**

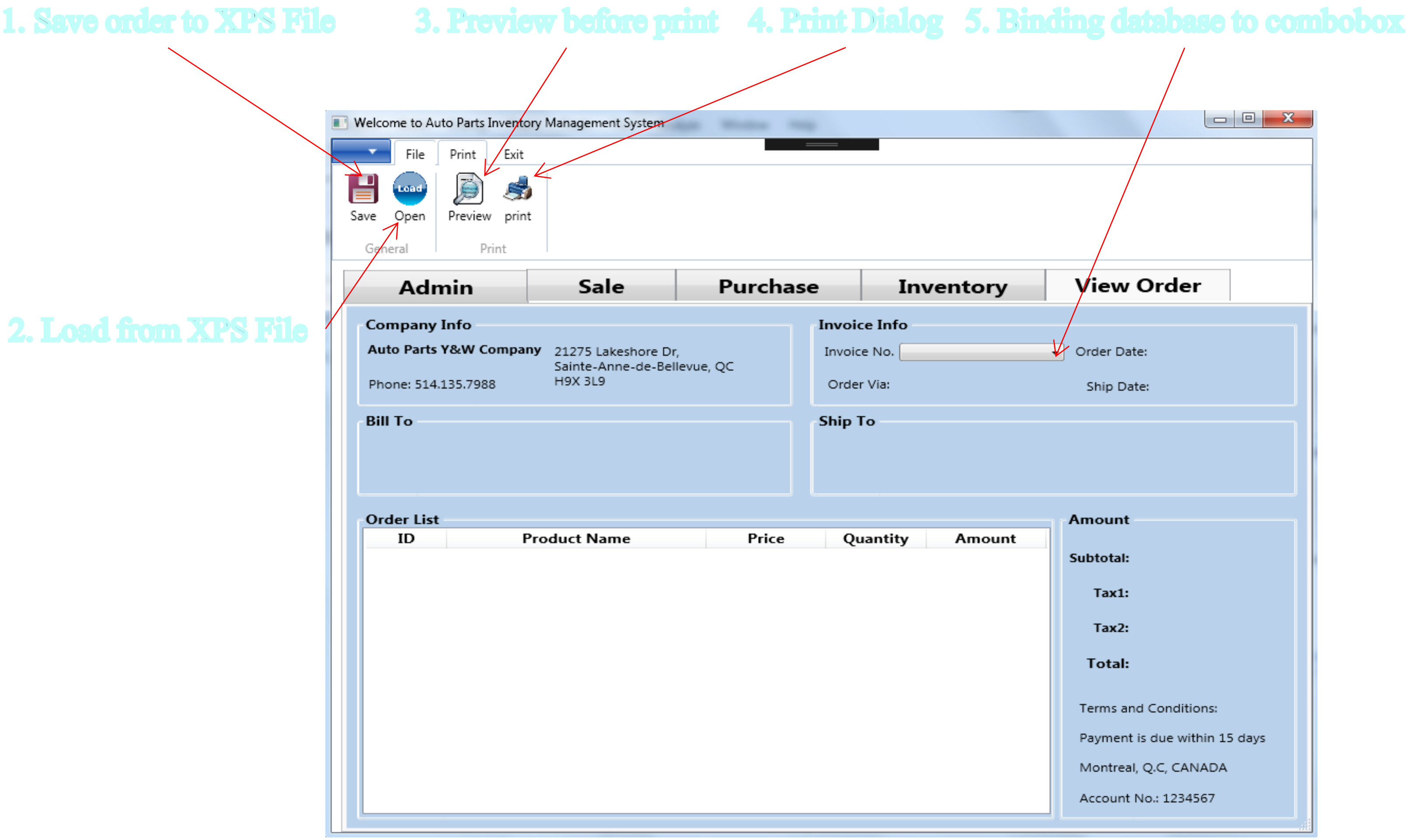


```
<ComboBox Name="comboSortBy" FontSize="14"  FontWeight="Bold"
SelectionChanged="comboSortBy_SelectionChanged" DropDownClosed="comboSortBy_DropDownClosed"
HorizontalAlignment="Left" Margin="125,38,0,0" VerticalAlignment="Top" Width="320">
    <ComboBoxItem Content="ID"/>
    ...
</ComboBox>

private bool handle = true;
private void comboMCSortBy_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    ComboBox cmb = sender as ComboBox;
    handle = !cmb.IsDropDownOpen;
    Handle();
}
private void comboMCSortBy_DropDownClosed(object sender, EventArgs e)
{
    if (handle) Handle();
    handle = true;
}
private void Handle()
{
    CollectionView view = (CollectionView)CollectionViewSource
    .GetDefaultView(lvModifyCustomerList.ItemsSource);

    switch (comboMCSortBy.SelectedItem.ToString().Split(
    new string[] { ": " }, StringSplitOptions.None).Last())
    {
        case "ID":
            view.SortDescriptions.Clear();
            view.SortDescriptions.Add(new SortDescription
            ("CustSupplierId", ListSortDirection.Ascending));
            break;
        ...
    }
}
```

What we learned

- Azure Database Connectivity (Centralized Database)
- Implementing Ribbon
- Simplifying Database and Software Design for Maximizing User Friendly and Easy to Navigate

Future Work



1. Save order to XPS File

3. Preview before print

4. Print Dialog

5. Binding database to combobox

2. Load from XPS File

Welcome to Auto Parts Inventory Management System

Company Info

Auto Parts Y&W Company

21275 Lakeshore Dr,
Sainte-Anne-de-Bellevue, QC
H9X 3L9

Phone: 514.135.7988

Invoice Info

Invoice No.

Order Date:

Order Via:

Ship Date:

Bill To

Ship To

Order List

ID	Product Name	Price	Quantity	Amount

Amount

Subtotal:

Tax1:

Tax2:

Total:

Terms and Conditions:

Payment is due within 15 days

Montreal, Q.C, CANADA

Account No.: 1234567

Summary

- Designing Database and setting inter-relations
- Designing Overall Screen Design, (was a gradual progress, changes were made as we discovered requirements and new challenges)
- Implementing Ribbon
- Inserting into tables using SQL commands
- Setting color flags when items run low in stock
- Great Learning Process