```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Data;
using Microsoft.Windows.Controls;
using System.Collections;
using System.ComponentModel;

namespace KMFVolume
{
    /// <summary>
    /// Interaction logic for Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        KMFDataSet ds = new KMFDataSet();
        KMFDataSetTableAdapters.CategoryTableAdapter taCategory =
            new KMFDataSetTableAdapters.CategoryTableAdapter();
        KMFDataSetTableAdapters.SubCategoryTableAdapter taSubCategory =
            new KMFDataSetTableAdapters.SubCategoryTableAdapter();
        KMFDataSetTableAdapters.ProductTableAdapter taProduct =
            new KMFVolume.KMFDataSetTableAdapters.ProductTableAdapter();

        KMFDataSetTableAdapters.OrderTableAdapter taOrder =
            new KMFVolume.KMFDataSetTableAdapters.OrderTableAdapter();
        KMFDataSetTableAdapters.OrderItemTableAdapter taOrderItem =
            new KMFVolume.KMFDataSetTableAdapters.OrderItemTableAdapter();
        KMFDataSet.OrderItemDataTable dtOrderItem = new KMFDataSet.OrderItemDataTable();

        KMFDataSetTableAdapters.OrderCartonTableAdapter taOrderCartion =
            new KMFVolume.KMFDataSetTableAdapters.OrderCartonTableAdapter();
        KMFDataSetTableAdapters.ItemCartonTableAdapter taItemCarton =
            new KMFVolume.KMFDataSetTableAdapters.ItemCartonTableAdapter();
        KMFDataSetTableAdapters.QueriesTableAdapter KMFQuery =
            new KMFVolume.KMFDataSetTableAdapters.QueriesTableAdapter();

        KMFDataSetTableAdapters.UserTableAdapter taUser =
            new KMFVolume.KMFDataSetTableAdapters.UserTableAdapter();

        ICollectionView OrderCollection;

        const int ci20Foot = 1164;
        private int CurrentOrderID = 1;
        private int _nextItemNo = 0;

        public Window1()
        {
            InitializeComponent();
            InitData();
            tbInnerVolume.IsReadOnly = true;
        }

        private void InitData()
        {
            taOrder.ClearBeforeFill=true;
            GetOrder();
```

```
            dgOrder.SelectedIndex = 0;
            CurrentOrderID = int.Parse(ds.Order.Rows[0]["OrderID"].ToString());


            taUser.Fill(ds.User);
            cbUser.ItemsSource = ds.User;


            //GetOrderItem();
        }

        private void GetOrder()
        {
            taOrder.Fill(ds.Order);
            dgOrder.DataContext = ds.Order;
            OrderCollection = CollectionViewSource.GetDefaultView(ds.Order);
            dgOrder.IsSynchronizedWithCurrentItem = true;



        }

        private void GetOrderItem()
        {
            taCategory.Fill(ds.Category);
            taSubCategory.Fill(ds.SubCategory);
            taProduct.Fill(ds.Product);
            taOrderItem.FillBy(dtOrderItem, CurrentOrderID);
            ReNumber();

            Grid1.DataContext = ds.Category;
            dgItems.DataContext = dtOrderItem;
            dgItems.CanUserAddRows = false;

            GetCartons();

        }

        private void SetItemNo()
        {
            if (dtOrderItem.Rows.Count > 0)
            {
                _nextItemNo = int.Parse(dtOrderItem.Rows[dtOrderItem.Rows.Count - 1][
    "ItemNo"].ToString());
            }
            else
            {
                _nextItemNo = 0;
            }
        }

        private void GetCartons()
        {
            taOrderCartion.FillBy(ds.OrderCarton, CurrentOrderID);
            taItemCarton.Fill(ds.ItemCarton);

            dgCarton.DataContext = ds.OrderCarton;
            dgCarton.CanUserAddRows = false;

            tbInnerVolume.Text = KMFQuery.OrderVolume(CurrentOrderID) != null ? KMFQuery.
    OrderVolume(CurrentOrderID).ToString() : "0";
            tbCost.Text = taOrder.OrderCost(CurrentOrderID) != null ? Decimal.Parse
    (taOrder.OrderCost(CurrentOrderID).ToString()).ToString("C") : "$0.00";
        }

        private int NextItemNo()
```

```
        {
            _nextItemNo ++;
            return _nextItemNo;

        }

        private void ReNumber()
        {
            int i = 0;

            foreach (DataRow row in dtOrderItem.Rows)
            {
                i++;
                row["ItemNo"] = i;

            }
            _nextItemNo = i;

        }

        private void SetRowInfo(DataRow row, string kmf)
        {
            KMFDataSet.ProductDataTable product = new KMFDataSet.ProductDataTable();
            taProduct.FillBy(product,kmf);

            object casepack = string.IsNullOrEmpty(product.Rows[0]["InnerCasePack"].
    ToString()) ? product.Rows[0]["MasterCasePack"] : product.Rows[0]["InnerCasePack"];

            if (casepack == null)
            {

                throw new Exception("Can not add - there is no for this product");
            }
            if (product.Rows.Count > 0)
            {
                row["Quantity"] = casepack;
                row["Cost"] = product.Rows[0]["UnitCost"];
                row["ItemNo"] = NextItemNo();

            }


        }

        private void button1_Click(object sender, RoutedEventArgs e)
        {
            lbErr.Content = "";
            DataRow row = dtOrderItem.NewRow();
            row["OrderID"] = CurrentOrderID;
            row["KMF#"] = cbProduct.SelectedValue;
            SetRowInfo(row, cbProduct.SelectedValue.ToString());
            dtOrderItem.Rows.Add(row);
            ScrollToEnd(dgItems);
        }

        private void ScrollToEnd(DataGrid pgrid)
        {
            if (pgrid.Items.Count > 0)
            {
                var border = VisualTreeHelper.GetChild(pgrid, 0) as Decorator;
                if (border != null)
                {
                    var scroll = border.Child as ScrollViewer;
                    if (scroll != null) scroll.ScrollToEnd();
                }
            }
```

```
        }

        private void button2_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                taOrderItem.Update(dtOrderItem);
                GetCartons();
                ScrollToEnd(dgCarton);
                lbErr.Content = "Record Saved";
                lbErr.Foreground = Brushes.Red;
            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }
        }

        private void Contains_Click(object sender, RoutedEventArgs e)
        {
            DataRowView obj = ((FrameworkElement)sender).DataContext as DataRowView;
            int _containerid = int.Parse(obj["OrderCartonID"].ToString());
            ContainerContents fcontents = new ContainerContents(_containerid);
            fcontents.ShowDialog();

        }

        private void Delete_Click(object sender, RoutedEventArgs e)
        {
            DataRowView obj = ((FrameworkElement)sender).DataContext as DataRowView;
            obj.Delete();
        }


        private void tcKMF_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {

            if ((sender is TabControl) && (e.OriginalSource is TabControl))
            {
                if (CurrentOrderID < 0)
                {
                    tcKMF.SelectedIndex = 0;
                    e.Handled = true;
                }


                if (tcKMF.SelectedIndex > 0)
                {
                    SaveOrder();
                    GetOrderItem();
                }

                e.Handled = true;
            }
        }

        private void dgOrder_SelectedCellsChanged(object sender,                    ↙
    SelectedCellsChangedEventArgs e)
        {
            DataRowView rview = ((FrameworkElement)sender as DataGrid).CurrentItem as   ↙
    DataRowView;
            if (rview != null)
            {
                CurrentOrderID = int.Parse(rview["OrderID"].ToString());
            }
            // an alternative way
            if (OrderCollection != null)
```

```
            {
                DataRowView row1 = OrderCollection.CurrentItem as DataRowView;
                // set CurrentRowID
            }

        }

        private void button4_Click(object sender, RoutedEventArgs e)
        {
            DataRow row = ds.Order.NewOrderRow();
            row["OrderDate"] = DateTime.Now;
            ds.Order.Rows.Add(row);
        }

        private void SaveOrder()
        {
            try
            {
                taOrder.Update(ds.Order);

            }
            catch (Exception E)
            {
                MessageBox.Show(E.Message);
            }

        }

        private void button3_Click(object sender, RoutedEventArgs e)
        {
            SaveOrder();
        }

        private void tbInnerVolume_TextChanged(object sender, TextChangedEventArgs e)
        {
            decimal scratch = 0.00M;
            decimal.TryParse(tbInnerVolume.Text, out scratch);
            decimal perc = decimal.Round(( scratch/ ci20Foot) * 100,2);
            tbPerc.Text = perc.ToString() + "%";
        }



    }
}
```