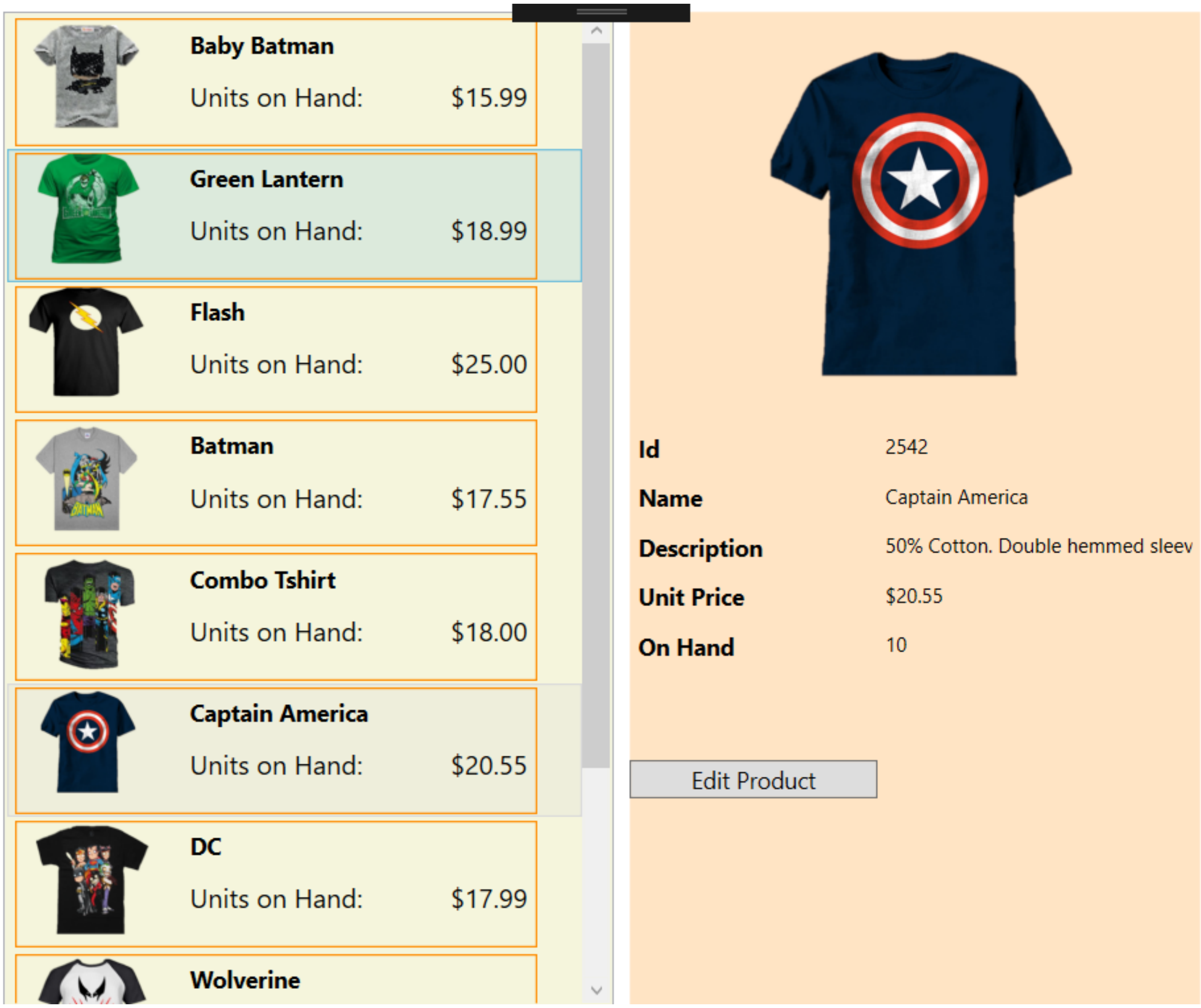
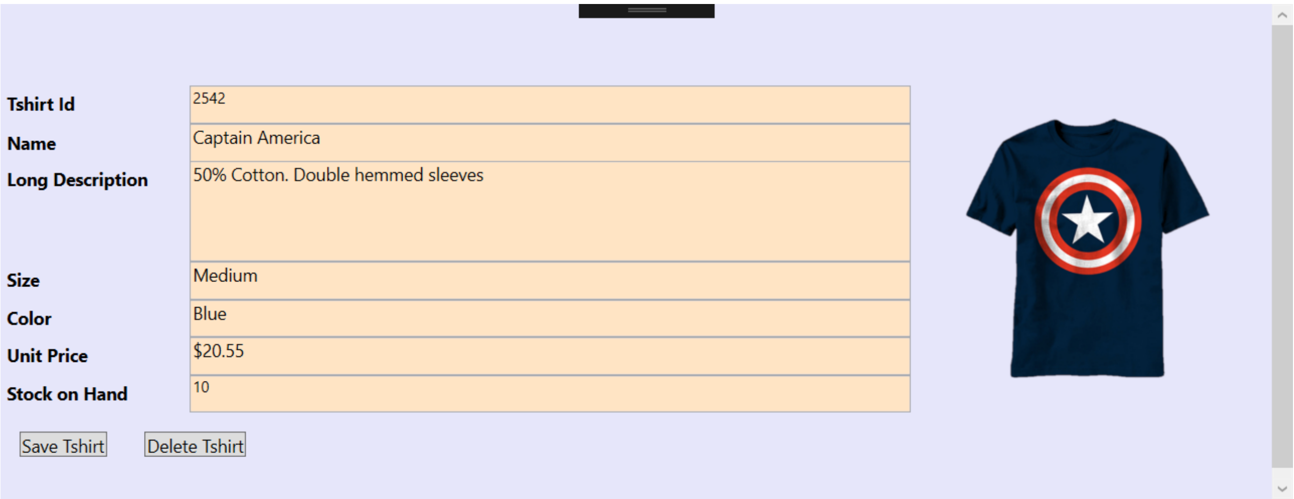


Baby Batman	
Units on Hand:	$15.99
Green Lantern	
Units on Hand:	$18.99
Flash	
Units on Hand:	$25.00
Batman	
Units on Hand:	$17.55
Combo Tshirt	
Units on Hand:	$18.00
Captain America	
Units on Hand:	$20.55
DC	
Units on Hand:	$17.99
Wolverine	

Id	2542
Name	Captain America
Description	50% Cotton. Double hemmed sleev
Unit Price	$20.55
On Hand	10



Tshirt Id	2542
Name	Captain America
Long Description	50% Cotton. Double hemmed sleeves
Size	Medium
Color	Blue
Unit Price	$20.55
Stock on Hand	10

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using TshirtStockMgtApp;
using TshirtStockMgtApp.DAL;
using TshirtStockMgtApp.Model;


namespace Driver
{
    class Program
    {
        private static TshirtRepository _tshirtRepository = new TshirtRepository();
        static void Main(string[] args)
        {
            List<Tshirt> _list = new List<TshirtStockMgtApp.Model.Tshirt>();
            _list = _tshirtRepository.GetTshirts();
            foreach (var item in _list)
            {
                Console.WriteLine(item.TshirtID);
            }
            Console.ReadKey();
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.ComponentModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Input;
using TshirtStockMgtApp.App.Messages;
using TshirtStockMgtApp.App.Services;
using TshirtStockMgtApp.App.Utility;
using TshirtStockMgtApp.Model;

namespace TshirtStockMgtApp.App.ViewModel
{
    public class TshirtDetailViewModel
    {
        public event PropertyChangedEventHandler PropertyChanged;
```

```
private void RaisePropertyChanged(string propertyName)
{
    if (PropertyChanged != null)
        PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
}
private Tshirt _selectedTshirt;
private TshirtDataService _tshirtDataService;
//public ObservableCollection<Category> Categories { get; set; }
public ICommand SaveCommand { get; set; }
public ICommand DeleteCommand { get; set; }

public Tshirt SelectedTshirt
{
    get { return _selectedTshirt; }
    set
    {
        _selectedTshirt = value;
        RaisePropertyChanged("SelectedTshirt");
    }
}

public TshirtDetailViewModel()
{
    SaveCommand = new CustomCommand(SaveTshirt, CanSaveTshirt);
    DeleteCommand = new CustomCommand(DeleteTshirt, CanDeleteTshirt);
    Messenger.Default.Register<Tshirt>(this, OnTshirtReceived);
    _tshirtDataService = new TshirtDataService();
    //LoadCategories();
}

//private void LoadCategories()
//{
//    Categories = _categoryDataService.GetCategories().ToObservableCollection();
//}

private void OnTshirtReceived(Tshirt tshirt)
{
    SelectedTshirt = tshirt;
    //SelectedCategory = _categoryDataService.GetCategory(SelectedTshirt.CategoryID);
}

private bool CanDeleteTshirt(object obj)
{
    return true;
}
```

```
        private void DeleteTshirt(object obj)
        {
            _tshirtDataService.DeleteTshirt(SelectedTshirt);
            Messenger.Default.Send<UpdateListMessage>(new UpdateListMessage());
        }

        private bool CanSaveTshirt(object obj)
        {
            return true;
        }

        private void SaveTshirt(object obj)
        {
            //SelectedTshirt.CategoryID = SelectedCategory.CategoryID;
            _tshirtDataService.UpdateTshirt(SelectedTshirt);
            Messenger.Default.Send<UpdateListMessage>(new UpdateListMessage());
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.ComponentModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Input;
using TshirtStockMgtApp.App.Extensions;
using TshirtStockMgtApp.App.Messages;
using TshirtStockMgtApp.App.Services;
using TshirtStockMgtApp.App.Utility;
using TshirtStockMgtApp.Model;

namespace TshirtStockMgtApp.App.ViewModel
{
    public class TshirtOverviewViewModel : INotifyPropertyChanged
    {
        public event PropertyChangedEventHandler PropertyChanged;
        private TshirtDataService _tshirtDataService;
        private DialogService _dialogService;

        public ICommand EditCommand { get; set; }
        private void RaisePropertyChanged(string propertyName)
        {
            if (PropertyChanged != null)
        PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
}
private ObservableCollection<Tshirt> _tshirt;
public ObservableCollection<Tshirt> Tshirt
{
    get
    {
        return _tshirt;
    }
    set
    {
        _tshirt = value;
        RaisePropertyChanged("Tshirts");
    }
}
private Tshirt _selectedTshirt;
public Tshirt SelectedTshirt
{
    get
    {
        return _selectedTshirt;
    }
    set
    {
        _selectedTshirt = value;
        RaisePropertyChanged("SelectedTshirt");
    }
}
public TshirtOverviewViewModel()
{
    _tshirtDataService = new TshirtDataService();
    LoadData();
    LoadCommands();
    Messenger.Default.Register<UpdateListMessage>(this, OnUpdateListMessage);
    _dialogService = new DialogService();
}

private void OnUpdateListMessage(UpdateListMessage obj)
{
    LoadData();
    _dialogService.CloseDialog();
}

private void LoadCommands()
{
    //initialize my command
```

```
            //Action is the code to invoke when the command is invoked
            //CanExecute is the predicate that determines if the command can be executed.
            EditCommand = new CustomCommand(EditTshirt, CanEditTshirt);
        }

        private bool CanEditTshirt(object obj)
        {
            if (SelectedTshirt != null)
                return true;
            return false;
        }

        private void EditTshirt(object obj)
        {
            Messenger.Default.Send<Tshirt>(SelectedTshirt);
            _dialogService.ShowDialog();
        }

        private void LoadData()
        {
            Tshirt = _tshirtDataService.GetAllTshirts().ToObservableCollection();
        }

    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using TshirtStockMgtApp.App.ViewModel;

namespace TshirtStockMgtApp.App
{
    public class ViewModelLocator
    {

        private static TshirtOverviewViewModel _tshirtOverviewViewModel = new
TshirtOverviewViewModel();
        private static TshirtDetailViewModel _tshirtDetailViewModel = new
TshirtDetailViewModel();

        public static TshirtOverviewViewModel TshirtOverviewViewModelInstance
        {
            get
            {
                return _tshirtOverviewViewModel;
            }
        }

        public static TshirtDetailViewModel TshirtDetailViewModel
        {
            get
            {
                return _tshirtDetailViewModel;
            }
        }
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using TshirtStockMgtApp;
using TshirtStockMgtApp.Model;

namespace TshirtStockMgtApp.DAL
{
    public interface ITshirtRepository
    {
        //define interface for tshirt
        void DeleteTshirt(Tshirt tshirt);
        Tshirt GetATshirt();
        Tshirt GetByTshirtID(string tshirtID);
        List<Tshirt> GetTshirts();
        void UpdateTshirts(Tshirt tshirt);
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using TshirtStockMgtApp.Model;

namespace TshirtStockMgtApp.DAL
{
    public class TshirtRepository : ITshirtRepository
```

```
{
    private static List<Tshirt> _lstTshirts;

    public TshirtRepository()
    {

    }
    public void DeleteTshirt(Tshirt tshirt)
    {
        _lstTshirts.Remove(tshirt);
        using (connTshirtsEntities ctx = new connTshirtsEntities())
        {
            try
            {
                string qry = "DELETE FROM Tshirts WHERE tshirtID={0}";
                var rows = ctx.Database.ExecuteSqlCommand(qry, tshirt.TshirtID);
            }
            catch (System.Exception)
            {


            }
        }
    }

    public Tshirt GetATshirt()
    {
        if (_lstTshirts == null)
         LoadTshirts();
        return _lstTshirts.FirstOrDefault();
    }

    public Tshirt GetByTshirtID(string tshirtID)
    {
        if (_lstTshirts == null)
            LoadTshirts();
        return _lstTshirts.Where(t => t.TshirtID == tshirtID).FirstOrDefault();
    }

    public List<Tshirt> GetTshirts()
    {
        if (_lstTshirts == null)
            LoadTshirts();
        return _lstTshirts;
    }
```

```
        private void LoadTshirts()
        {
            using (connTshirtsEntities ctx = new connTshirtsEntities())
            {
                _lstTshirts = ctx.Tshirts.ToList();
            }
        }

        public void UpdateTshirts(Tshirt tshirt)
        {
            Tshirt tshirtToUpdate = _lstTshirts.Where(p => p.TshirtID==
tshirt.TshirtID).FirstOrDefault();
            tshirtToUpdate = tshirt;

            using (connTshirtsEntities ctx = new connTshirtsEntities())
            {
                try
                {
                    string qry = "UPDATE Tshirts SET TshirtID={0}, ShortDescription={1},
Longdescription={2}, Image={3}, Size={4}, Color={5}, UnitPrice={6}, InStock={7} ";
                    var rows = ctx.Database.ExecuteSqlCommand(qry, tshirt.TshirtID,
tshirt.ShortDescription, tshirt.LongDescription, tshirt.Image, tshirt.Size, tshirt.Color,
tshirt.UnitPrice, tshirt.InStock);
                }
                catch (System.Exception)
                {


                }
            }
        }

    }
}
```

```
<Window x:Class="TshirtStockMgtApp.App.View.TshirtDetailView"
        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:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:TshirtStockMgtApp.App.View"
         xmlns:i="http://schemas.microsoft.com/expression/2010/interactivity"
        mc:Ignorable="d"
        DataContext="{Binding Source={StaticResource mainViewModelLocator},
Path=TshirtDetailViewModel}"
        Title="TshirtDetailView" Height="431.2" Width="1042.15">
```

```
    <ScrollViewer>
        <Grid Margin="0" Height="403" Background="Lavender">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="150"></ColumnDefinition>
                <ColumnDefinition Width="2*"></ColumnDefinition>
                <ColumnDefinition Width="*"></ColumnDefinition>
            </Grid.ColumnDefinitions>
            <Grid.RowDefinitions>
                <RowDefinition Height="65"></RowDefinition>
                <RowDefinition Height="30"></RowDefinition>
                <RowDefinition Height="30"></RowDefinition>
                <RowDefinition Height="80"></RowDefinition>
                <RowDefinition Height="30"></RowDefinition>
                <RowDefinition Height="30"></RowDefinition>
                <RowDefinition Height="30"></RowDefinition>
                <RowDefinition Height="30"></RowDefinition>
                <RowDefinition Height="50"></RowDefinition>
            </Grid.RowDefinitions>
            <Label Grid.Row="0" Name="lblTshirtName" Content="{Binding ShortDescription}"
Grid.ColumnSpan="2" FontSize="40" FontWeight="Bold"></Label>

            <Image Name="imgTshirt" Grid.Column="2" Source="{Binding SelectedTshirt.Image,
Converter={StaticResource localImageConverter}}"
                   Grid.RowSpan ="10" MaxHeight="150" MaxWidth="150"></Image>
            <Label Grid.Row="1" Content="Tshirt Id" Style="{StaticResource styleLabel}"/>
            <Label Grid.Row="2" Content="Name" Style="{StaticResource styleLabel}"
VerticalAlignment="Center"/>
            <Label Grid.Row="3" Content="Long Description" Style="{StaticResource
styleLabel}"/>
            <Label Grid.Row="4" Content="Size" Style="{StaticResource styleLabel}"/>
            <Label Grid.Row="5" Content="Color" Style="{StaticResource styleLabel}"/>
            <Label Grid.Row="6" Content="Unit Price" Style="{StaticResource styleLabel}"
Grid.RowSpan="2"/>
            <Label Grid.Row="7" Content="Stock on Hand" Style="{StaticResource styleLabel}"/>

            <TextBox Background="Bisque" Grid.Row="1" Grid.Column="1" Text="{Binding
SelectedTshirt.TshirtID}" Name="txtTshirtID"></TextBox>
            <TextBox Background="Bisque" Grid.Row="2" Grid.Column="1" Text="{Binding
SelectedTshirt.ShortDescription}" Name="txtName" AcceptsReturn="True" Height="75"
Style="{StaticResource styleTextBox}"/>
            <TextBox Background="Bisque" Grid.Row="3" Grid.Column="1" Text="{Binding
SelectedTshirt.LongDescription}" TextWrapping="Wrap" Name="txtLongDesc"
Style="{StaticResource styleTextBox}"/>
            <TextBox Background="Bisque" Grid.Row="4" Grid.Column="1" Text="{Binding
SelectedTshirt.Size}" TextWrapping="Wrap" Name="txtSize" Style="{StaticResource
styleTextBox}"/>
```

```
            <TextBox Background="Bisque" Grid.Row="5" Grid.Column="1" Text="{Binding
SelectedTshirt.Color}" TextWrapping="Wrap" Name="txtColor" Style="{StaticResource
styleTextBox}"/>
            <TextBox Background="Bisque" Grid.Row="6" Grid.Column="1" Name="txtUnitPrice"
Text="{Binding  SelectedTshirt.UnitPrice, Converter={StaticResource localPriceConverter}}"
Style="{StaticResource styleTextBox}"></TextBox>
            <TextBox Background="Bisque" Grid.Row="7" Grid.Column="1" Name="txtInStock"
Text="{Binding SelectedTshirt.InStock}"/>

            <StackPanel Grid.Row="10" Orientation="Horizontal" Grid.ColumnSpan="2">
                <Button x:Name="btnSave" Margin="15,15,15,15.2" Content="Save Tshirt"
FontSize="14">
                     <i:Interaction.Triggers>
                        <i:EventTrigger EventName="Click">
                            <!-- an action on the view model-->
                            <i:InvokeCommandAction Command="{Binding SaveCommand}"/>
                        </i:EventTrigger>
                    </i:Interaction.Triggers>
                </Button>
                <Button x:Name="btnDelete" Margin="15,15,15,15.2" Content="Delete Tshirt"
FontSize="14">
                    <i:Interaction.Triggers>
                        <i:EventTrigger EventName="Click">
                            <i:InvokeCommandAction Command="{Binding DeleteTshirt}"/>
                        </i:EventTrigger>
                    </i:Interaction.Triggers>
                </Button>
            </StackPanel>

        </Grid>

    </ScrollViewer>
</Window>


<Window x:Class="TshirtStockMgtApp.App.View.TshirtOverviewView"
        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:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:TshirtStockMgtApp.App.View"
        mc:Ignorable="d"
         DataContext="{Binding Source={StaticResource mainViewModelLocator},
Path=TshirtOverviewViewModelInstance}"
        Title="TshirtOverviewView" Height="650" Width="750">
    <Grid>
        <Grid.ColumnDefinitions>
```

```
      <ColumnDefinition Width="380"></ColumnDefinition>
      <ColumnDefinition Width="*"></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <ListView Grid.Column="0" Margin="5" ItemsSource="{Binding Tshirt}"
SelectedItem="{Binding SelectedTshirt}"
         Name="lstTshirt" FontSize="16" Background="Beige">
      <ListView.ItemTemplate>
        <DataTemplate>
          <Border BorderBrush="DarkOrange" BorderThickness="1">
            <Grid>
              <Grid.ColumnDefinitions>
                <ColumnDefinition Width="100"/>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="90"/>
              </Grid.ColumnDefinitions>
              <Grid.RowDefinitions>
                <RowDefinition Height="30"/>
                <RowDefinition Height="30"/>
                <RowDefinition Height="15"/>
              </Grid.RowDefinitions>
              <Image Source="{Binding Image,Converter={StaticResource
localImageConverter}}" Grid.RowSpan="2" VerticalAlignment="Center"/>
              <Label Content="{Binding ShortDescription}" Grid.ColumnSpan="2"
Grid.Column="1" FontSize="14" FontWeight="Bold"/>
              <StackPanel Grid.Row="1" Orientation="Horizontal" Grid.Column="1">
                <Label Content="Units on Hand:"/>
                <Label Content="{Binding OnHand,Converter={StaticResource
localDoNothingConverter}}"/>

              </StackPanel>
              <Label Content="{Binding UnitPrice,Converter={StaticResource
localPriceConverter}}" HorizontalAlignment="Right" Grid.Row="1" Grid.Column="2"/>
            </Grid>
          </Border>
        </DataTemplate>
      </ListView.ItemTemplate>

    </ListView>
    <Grid Grid.Column="1" Margin="5" Name="gridSelectedItem" Background="Bisque">
      <Grid.RowDefinitions>
        <RowDefinition Height="250"></RowDefinition>
        <RowDefinition Height="30"></RowDefinition>
        <RowDefinition Height="30"></RowDefinition>
        <RowDefinition Height="30"></RowDefinition>
        <RowDefinition Height="30"></RowDefinition>
        <RowDefinition Height="30"></RowDefinition>
```

```
                <RowDefinition Height="50"></RowDefinition>
                <RowDefinition Height="30"></RowDefinition>
            </Grid.RowDefinitions>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="150"></ColumnDefinition>
                <ColumnDefinition Width="*"></ColumnDefinition>
            </Grid.ColumnDefinitions>
            <Image Name="imgProduct" Grid.Row="0" Source="{Binding SelectedTshirt.Image,
Converter={StaticResource localImageConverter}}"
                   Grid.ColumnSpan="2" MaxHeight="250" MaxWidth="150"></Image>
            <Label Grid.Row="1" Content="Id" Style="{StaticResource styleLabel}"/>
            <Label Grid.Row="2" Content="Name" Style="{StaticResource styleLabel}"/>
            <Label Grid.Row="3" Content="Description" Style="{StaticResource styleLabel}"/>
            <Label Grid.Row="4" Content="Unit Price" Style="{StaticResource styleLabel}"/>
            <Label Grid.Row="5" Content="On Hand" Style="{StaticResource styleLabel}"/>

            <Label Grid.Row="1" Grid.Column="1" Name="lblID" Content="{Binding
SelectedTshirt.TshirtID, Mode=OneWay}"></Label>
            <Label Grid.Row="2" Grid.Column="1" Name="lblName" Content="{Binding
SelectedTshirt.ShortDescription}"></Label>
            <Label Grid.Row="3" Grid.Column="1" Name="lblDescription"  Content="{Binding
SelectedTshirt.LongDescription}"></Label>
            <Label Grid.Row="4" Grid.Column="1" Name="lblPrice"  Content="{Binding
SelectedTshirt.UnitPrice, Converter={StaticResource localPriceConverter}}"></Label>
            <Label Grid.Row="5" Grid.Column="1" Name="lblOnHand"  Content="{Binding
SelectedTshirt.InStock}"></Label>

            <StackPanel Grid.Row="7" VerticalAlignment="Center" >
                <Button  Name="btnEdit" Content="Edit Product" Command="{Binding
EditCommand}" FontSize="14"/>
            </StackPanel>
        </Grid>
    </Grid>
</Window>
```