LAPORAN RESPONSI

PEMROQAMAN LANJUTAN

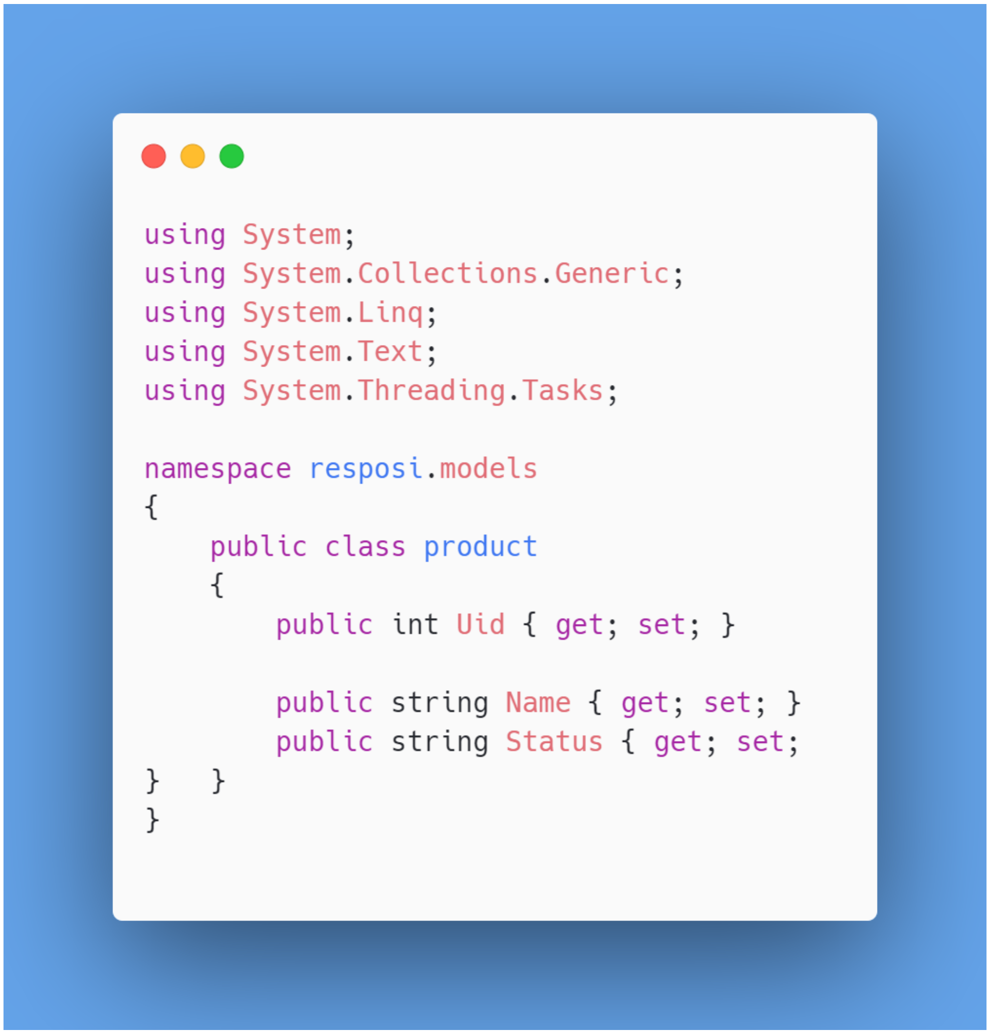
DI SUSUN OLEH :

MUHAMMAD ABDUL RAAFI

21.11.3867

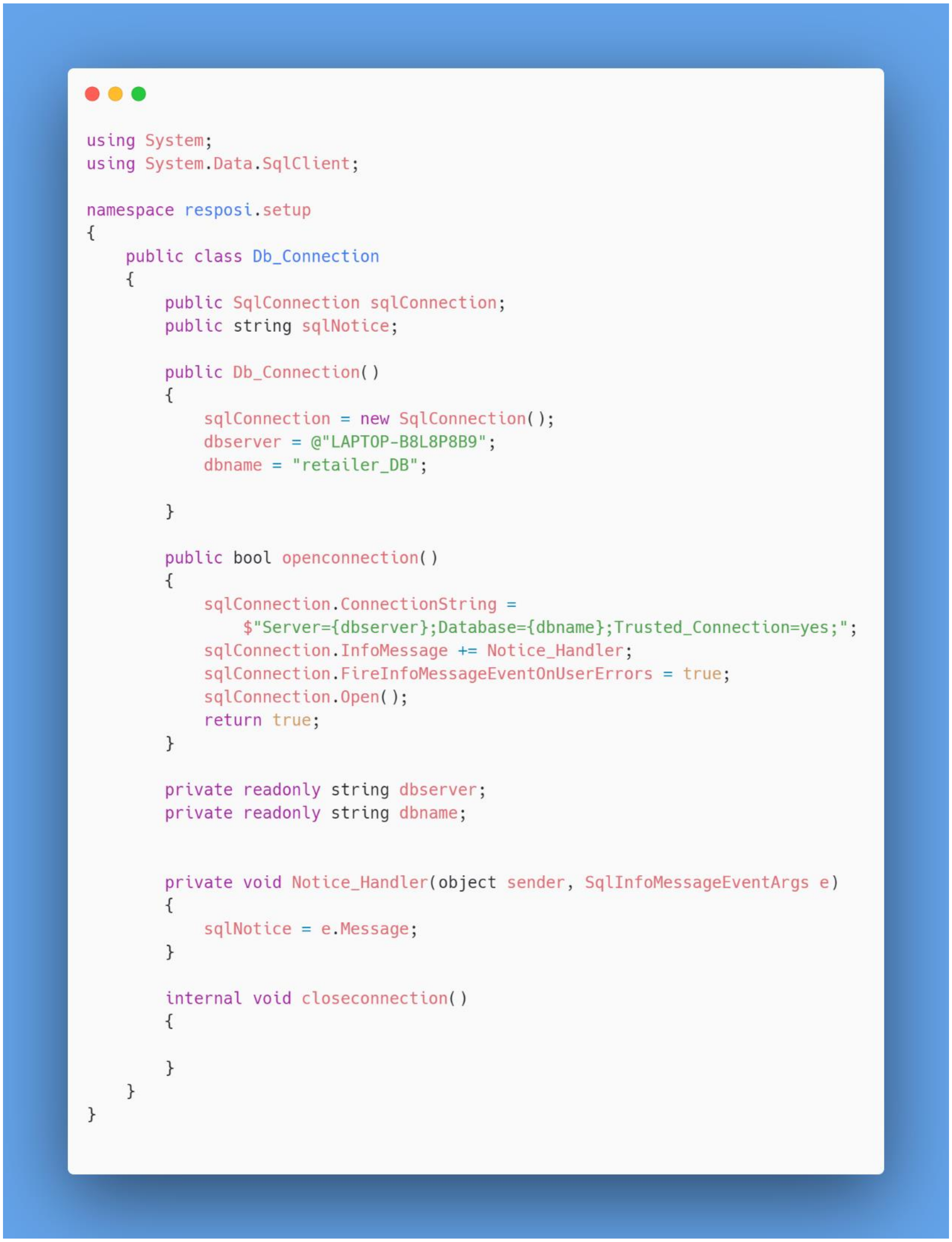
IF 01

- **Product.cs**



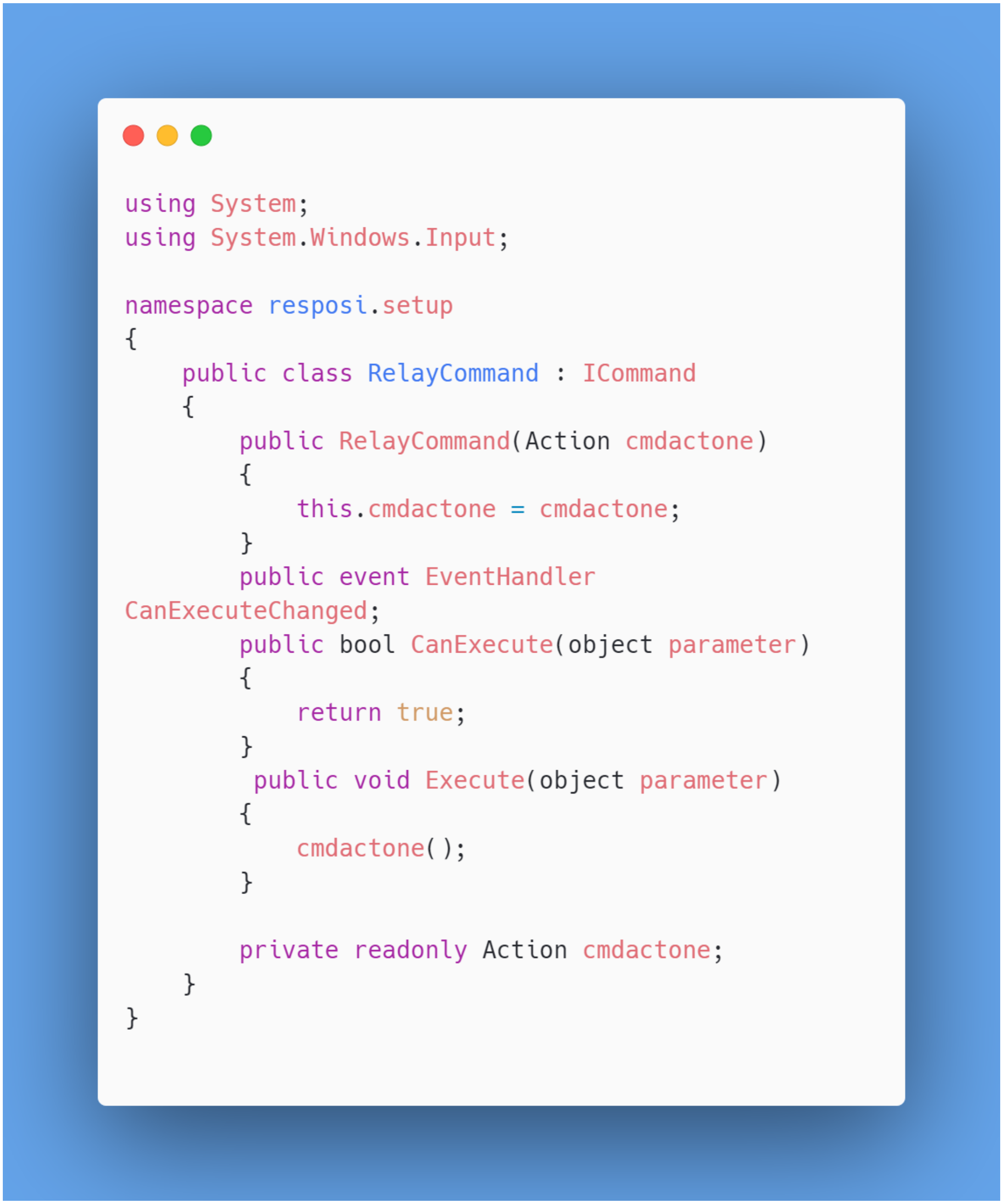
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace resposi.models
{
    public class product
    {
        public int Uid { get; set; }

        public string Name { get; set; }
        public string Status { get; set;
}   }
}
```

- **Db_Connection.cs**



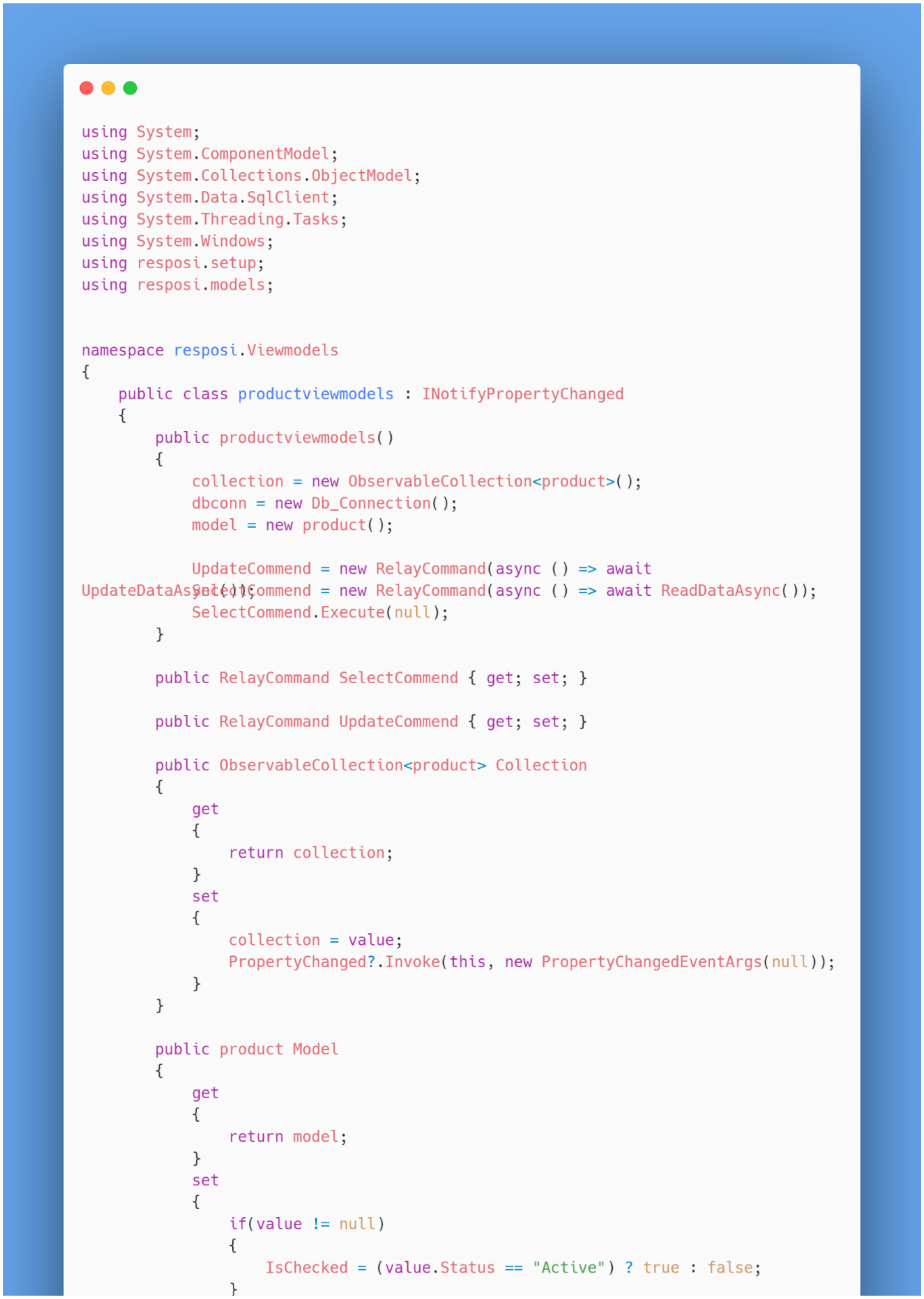
```
using System;
using System.Data.SqlClient;

namespace resposi.setup
{
    public class Db_Connection
    {
        public SqlConnection sqlConnection;
        public string sqlNotice;

        public Db_Connection()
        {
            sqlConnection = new SqlConnection();
            dbserver = @"LAPTOP-B8L8P8B9";
            dbname = "retailer_DB";

        }

        public bool openconnection()
        {
            sqlConnection.ConnectionString =
                $"Server={dbserver};Database={dbname};Trusted_Connection=yes;";
            sqlConnection.InfoMessage += Notice_Handler;
            sqlConnection.FireInfoMessageEventOnUserErrors = true;
            sqlConnection.Open();
            return true;
        }

        private readonly string dbserver;
        private readonly string dbname;


        private void Notice_Handler(object sender, SqlInfoMessageEventArgs e)
        {
            sqlNotice = e.Message;
        }

        internal void closeconnection()
        {

        }
    }
}
```

- **Relay Command.cs**

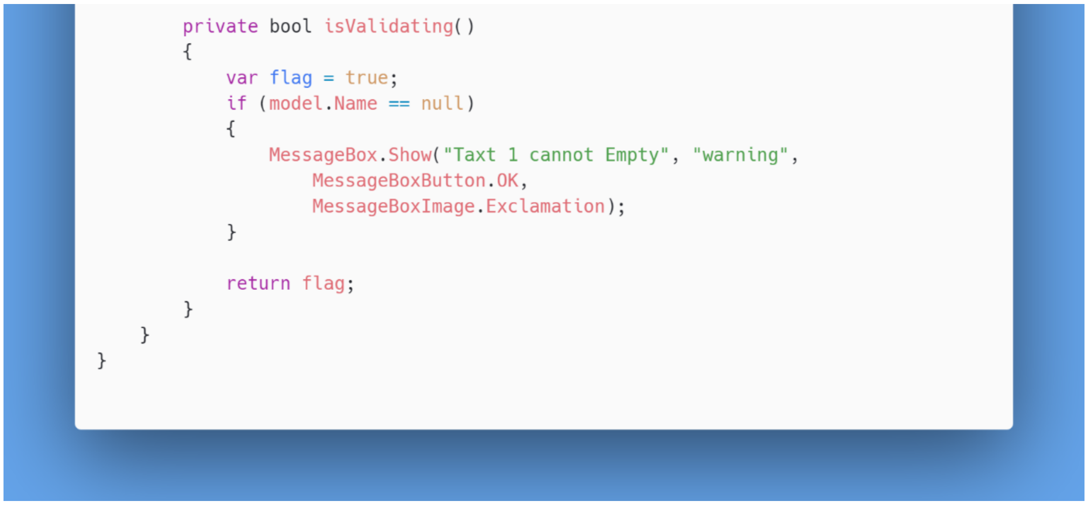


```
using System;
using System.Windows.Input;

namespace resposi.setup
{
    public class RelayCommand : ICommand
    {
        public RelayCommand(Action cmdactone)
        {
            this.cmdactone = cmdactone;
        }
        public event EventHandler
CanExecuteChanged;
        public bool CanExecute(object parameter)
        {
            return true;
        }
         public void Execute(object parameter)
        {
            cmdactone( );
        }

        private readonly Action cmdactone;
    }
}
```

- **Productviewmodels.cs**

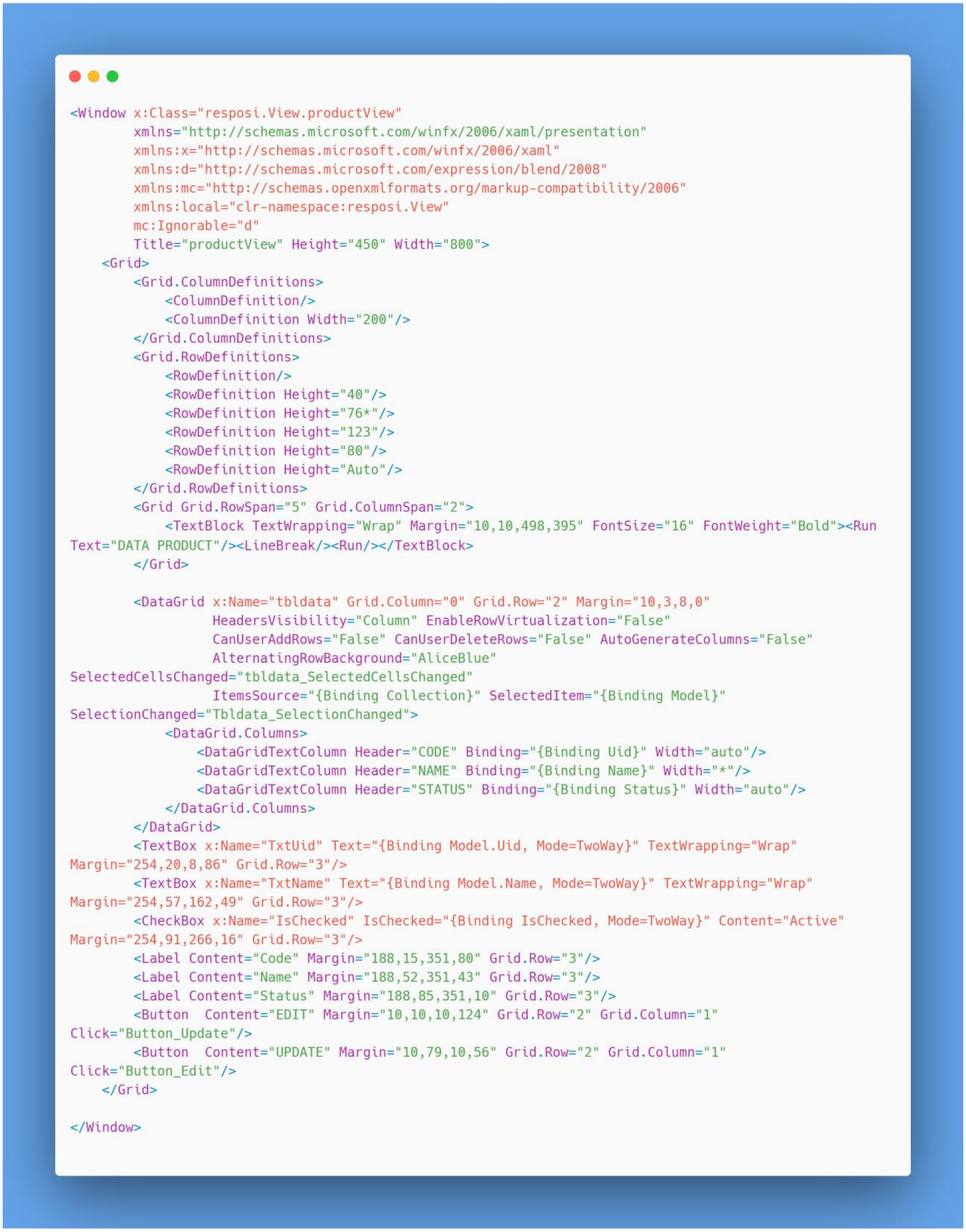


```
using System;
using System.ComponentModel;
using System.Collections.ObjectModel;
using System.Data.SqlClient;
using System.Threading.Tasks;
using System.Windows;
using resposi.setup;
using resposi.models;


namespace resposi.Viewmodels
{
    public class productviewmodels : INotifyPropertyChanged
    {
        public productviewmodels()
        {
            collection = new ObservableCollection<product>();
            dbconn = new Db_Connection();
            model = new product();

            UpdateCommend = new RelayCommand(async () => await
UpdateDataAsync());
            SelectCommend = new RelayCommand(async () => await ReadDataAsync());
            SelectCommend.Execute(null);
        }

        public RelayCommand SelectCommend { get; set; }

        public RelayCommand UpdateCommend { get; set; }

        public ObservableCollection<product> Collection
        {
            get
            {
                return collection;
            }
            set
            {
                collection = value;
                PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(null));
            }
        }

        public product Model
        {
            get
            {
                return model;
            }
            set
            {
                if(value != null)
                {
                    IsChecked = (value.Status == "Active") ? true : false;
                }
                model = value;
                PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(null));
            }
        }

        public bool IsChecked
        {
            get
            {
                return check;
            }
            set
            {
                check = value;
                PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(null));
            }
        }

        public event PropertyChangedEventHandler PropertyChanged;
        public event Action OnCallBack;

        private readonly Db_Connection dbconn;
        private ObservableCollection<product> collection;
        private product model;
        private bool check;

        private async Task ReadDataAsync()
        {
            dbconn.openconnection();
            await Task.Delay(0);
            var query = "SELECT * FROM products";
            var sqlcmd = new SqlCommand(query, dbconn.sqlConnection);
            var sqlresult = sqlcmd.ExecuteReader();

            if (sqlresult.HasRows)
            {
                Collection.Clear();
                while (sqlresult.Read())
                {
                    Collection.Add(new product
                    {
                        Uid = sqlresult[0] as int? ?? 0,
                        Name = sqlresult[1].ToString(),
                        Status = sqlresult[2].ToString() == "1" ?
                        "Active" : "Not Active",
                    });
                }
            }
            dbconn.closeconnection();
            OnCallBack?.Invoke();
        }

        private async Task UpdateDataAsync()
        {
            if(isValidating())
            {
                dbconn.openconnection();
                var state = check ? "1" : "0";
                var Query = $"Update products SET" +
                            $"Name = '{model.Name}'," +
                            $"Status = '{model.Status}'," +
                            $"WHERE Uid = '{model.Uid}',";
                var sqlcmd = new SqlCommand(Query, dbconn.sqlConnection);
                sqlcmd.ExecuteReader();
                dbconn.closeconnection();
                await ReadDataAsync();
            }
        }
```



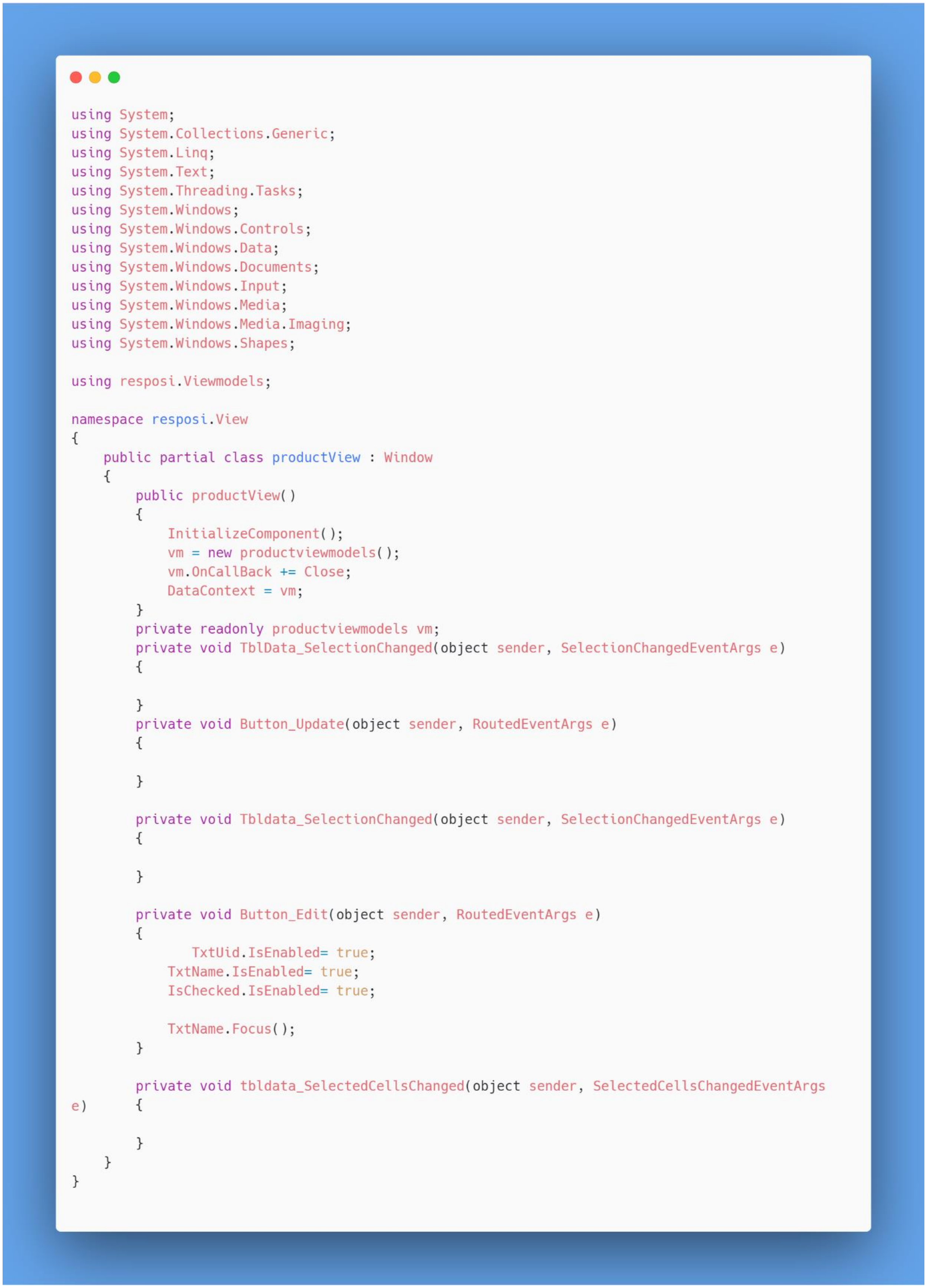
```
        private bool isValidating()
        {
            var flag = true;
            if (model.Name == null)
            {
                MessageBox.Show("Taxt 1 cannot Empty", "warning",
                    MessageBoxButton.OK,
                    MessageBoxImage.Exclamation);
            }

            return flag;
        }
    }
}
```

- **productView.xaml**



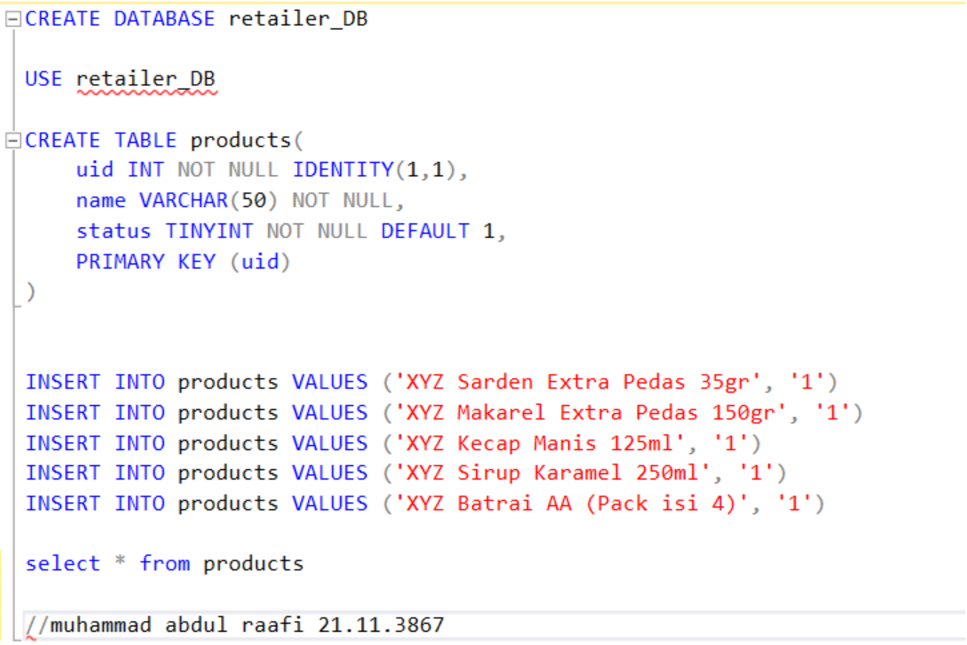
```
<Window x:Class="resposi.View.productView"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:resposi.View"
        mc:Ignorable="d"
        Title="productView" Height="450" Width="800">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition Width="200"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition Height="40"/>
            <RowDefinition Height="76*"/>
            <RowDefinition Height="123"/>
            <RowDefinition Height="80"/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>
        <Grid Grid.RowSpan="5" Grid.ColumnSpan="2">
            <TextBlock TextWrapping="Wrap" Margin="10,10,498,395" FontSize="16" FontWeight="Bold"><Run
Text="DATA PRODUCT"/><LineBreak/><Run/></TextBlock>
        </Grid>

        <DataGrid x:Name="tbldata" Grid.Column="0" Grid.Row="2" Margin="10,3,8,0"
                  HeadersVisibility="Column" EnableRowVirtualization="False"
                  CanUserAddRows="False" CanUserDeleteRows="False" AutoGenerateColumns="False"
                  AlternatingRowBackground="AliceBlue"
SelectedCellsChanged="tbldata_SelectedCellsChanged"
                  ItemsSource="{Binding Collection}" SelectedItem="{Binding Model}"
SelectionChanged="Tbldata_SelectionChanged">
            <DataGrid.Columns>
                <DataGridTextColumn Header="CODE" Binding="{Binding Uid}" Width="auto"/>
                <DataGridTextColumn Header="NAME" Binding="{Binding Name}" Width="*"/>
                <DataGridTextColumn Header="STATUS" Binding="{Binding Status}" Width="auto"/>
            </DataGrid.Columns>
        </DataGrid>
        <TextBox x:Name="TxtUid" Text="{Binding Model.Uid, Mode=TwoWay}" TextWrapping="Wrap"
Margin="254,20,8,86" Grid.Row="3"/>
        <TextBox x:Name="TxtName" Text="{Binding Model.Name, Mode=TwoWay}" TextWrapping="Wrap"
Margin="254,57,162,49" Grid.Row="3"/>
        <CheckBox x:Name="IsChecked" IsChecked="{Binding IsChecked, Mode=TwoWay}" Content="Active"
Margin="254,91,266,16" Grid.Row="3"/>
        <Label Content="Code" Margin="188,15,351,80" Grid.Row="3"/>
        <Label Content="Name" Margin="188,52,351,43" Grid.Row="3"/>
        <Label Content="Status" Margin="188,85,351,10" Grid.Row="3"/>
        <Button  Content="EDIT" Margin="10,10,10,124" Grid.Row="2" Grid.Column="1"
Click="Button_Update"/>
        <Button  Content="UPDATE" Margin="10,79,10,56" Grid.Row="2" Grid.Column="1"
Click="Button_Edit"/>
    </Grid>

</Window>
```

- **productView.xaml.cs**



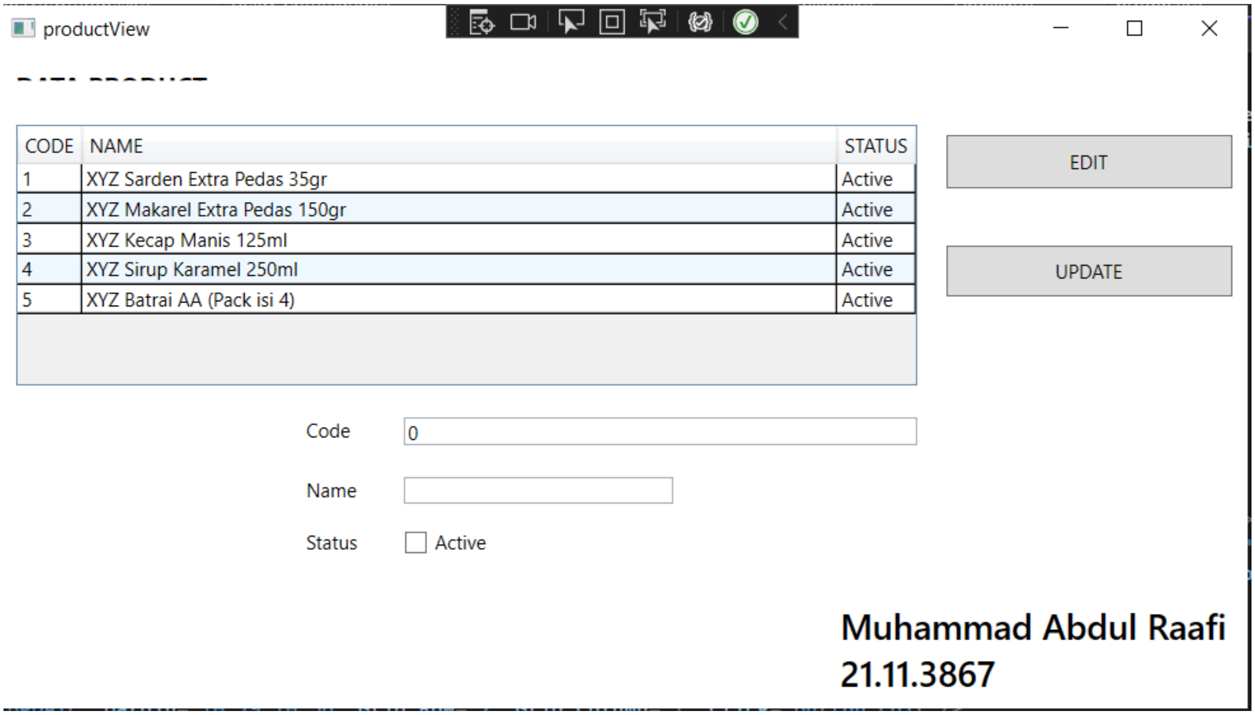
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

using resposi.Viewmodels;

namespace resposi.View
{
    public partial class productView : Window
    {
        public productView()
        {
            InitializeComponent();
            vm = new productviewmodels();
            vm.OnCallBack += Close;
            DataContext = vm;
        }
        private readonly productviewmodels vm;
        private void TblData_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {

        }
        private void Button_Update(object sender, RoutedEventArgs e)
        {

        }

        private void Tbldata_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {

        }

        private void Button_Edit(object sender, RoutedEventArgs e)
        {
               TxtUid.IsEnabled= true;
            TxtName.IsEnabled= true;
            IsChecked.IsEnabled= true;

            TxtName.Focus();
        }

        private void tbldata_SelectedCellsChanged(object sender, SelectedCellsChangedEventArgs
e)      {

        }
    }
}
```

- **Sql**



```
CREATE DATABASE retailer_DB

USE retailer_DB

CREATE TABLE products(
    uid INT NOT NULL IDENTITY(1,1),
    name VARCHAR(50) NOT NULL,
    status TINYINT NOT NULL DEFAULT 1,
    PRIMARY KEY (uid)
)


INSERT INTO products VALUES ('XYZ Sarden Extra Pedas 35gr', '1')
INSERT INTO products VALUES ('XYZ Makarel Extra Pedas 150gr', '1')
INSERT INTO products VALUES ('XYZ Kecap Manis 125ml', '1')
INSERT INTO products VALUES ('XYZ Sirup Karamel 250ml', '1')
INSERT INTO products VALUES ('XYZ Batrai AA (Pack isi 4)', '1')

select * from products

//muhammad abdul raafi 21.11.3867
```

- **Hasil**



CODE	NAME	STATUS
1	XYZ Sarden Extra Pedas 35gr	Active
2	XYZ Makarel Extra Pedas 150gr	Active
3	XYZ Kecap Manis 125ml	Active
4	XYZ Sirup Karamel 250ml	Active
5	XYZ Batrai AA (Pack isi 4)	Active

Code 0

Name

Status □ Active

Muhammad Abdul Raafi
21.11.3867