

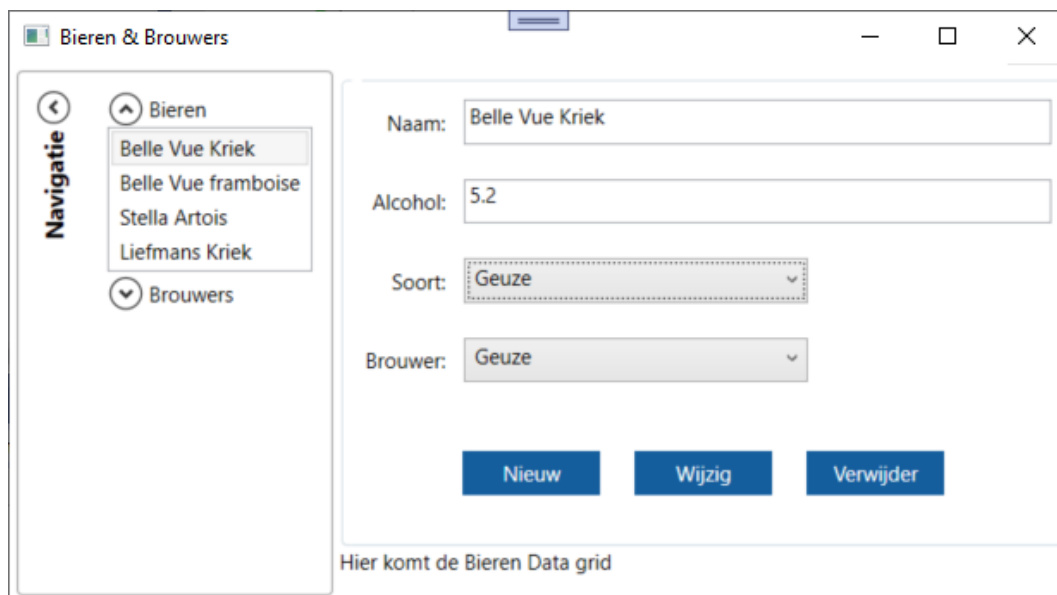
HOW TO –

C. WPF UI maken voor Bieren

(Voorbeeldapplicatie van bieren, Brouwers en soorten bieren)

<https://github.com/CSharpSyntraWest/HOWTO-WPF-MVVM>

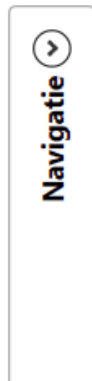
1. In dit hoofdstuk wordt een User interface gemaakt voor de Bieren, bestaande uit een bierenlijst (BierenView) en een formulier met details van een (geselecteerd) bier :



2. Eerst passen we de MainWindow.xaml aan om een grid met 2 kolommen te tonen. In de linkerkolom plaatsen we de lijst van bieren (later doen we hetzelfde met de lijst van Brouwers). Deze lijst wordt in een expander control geplaatst, zodat we met de pijltjes-icoontjes de lijst kunnen uitklappen (expand) of inklappen (collapse)
Er is eveneens een expander met header Navigatie, die alle elementen in de linkse kolom kan inklappen of uitklappen met een expand orientatie naar rechts toe.
3. In de 2de kolom wordt een details view van een geselecteerd bier in de lijst getoond. Via deze view gaan we later de CRUD (Create/Read/Update/Delete)-functionaliteit implementeren om een nieuw bier toe te voegen, te wijzigen of te verwijderen uit de lijst van bieren via de buttons.
4. Open MainWindow.xaml en wijzig de titel; `Title="Bieren & Brouwers"`
5. maak een Grid met 2 kolommen en 1 rij:

```
<Grid DataContext="{StaticResource MainViewModel}">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="auto" />
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>
</Grid>
```


6. In de linkerkolom voegen we een expansion control toe die naar rechts uitklapt (`ExpandDirection="Right"`):

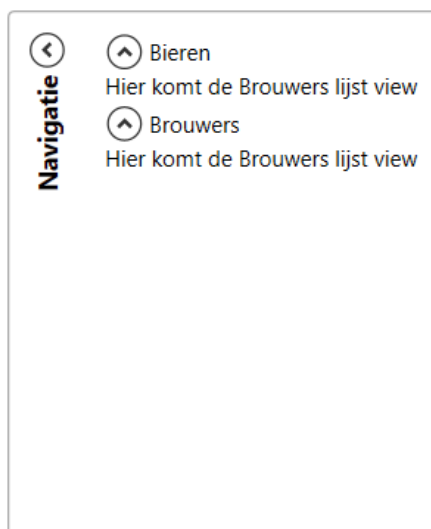


Om de header text vertikaal te plaatsen, doen we een rotatie van 270° van de header Textblock

```
<Expander Padding="10"
          Margin="10"
          BorderBrush="DarkGray"
          ExpandDirection="Right"
          Grid.Row="0"
          Grid.Column="0"
          IsExpanded="True">
  <Expander.Header>
    <TextBlock Text="Navigatie"
              FontSize="14"
              FontWeight="Bold">
      <TextBlock.LayoutTransform>
        <RotateTransform Angle="270" />
      </TextBlock.LayoutTransform>
    </TextBlock>
  </Expander.Header>
</Expander>
```

7. In de Expander (dus tussen de tags `<Expander..` en `</Expander>`) Expander plaatsen we een Stackpanel met 2 expanders die naar onder uitklappen (default gedrag van Expander):

 Bieren & Brouwers




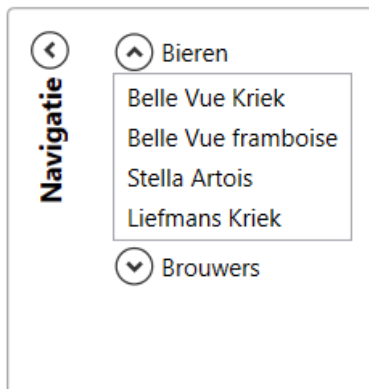
```

<Expander Padding="10"
          Margin="10"
          BorderBrush="DarkGray"
          ExpandDirection="Right"
          Grid.Row="0"
          Grid.Column="0"
          IsExpanded="True">
  <Expander.Header>
    <TextBlock Text="Navigatie"
              FontSize="14"
              FontWeight="Bold">
      <TextBlock.LayoutTransform>
        <RotateTransform Angle="270" />
      </TextBlock.LayoutTransform>
    </TextBlock>
  </Expander.Header>
  <StackPanel>
    <Expander Header="Bieren">
      <TextBlock Text="Hier komt de Brouwers lijst view"/>
    </Expander>
    <Expander Header="Brouwers">
      <TextBlock Text="Hier komt de Brouwers lijst view"/>
    </Expander>
  </StackPanel>
</Expander>

```

8. In de expander met Header tekst “Bieren” plaatsen we de BierenView usercontrol (deze usercontrol hebben we reeds gemaakt in oefening B) (verwijder de TextBlock). Wanneer je de app runt, en de Bieren Expander openklikt, zie je nu de lijst van biernamen verschijnen :

 Bieren & Brouwers



```

<Expander Padding="10"
          Margin="10"
          BorderBrush="DarkGray"
          ExpandDirection="Right"
          Grid.Row="0"
          Grid.Column="0"
          IsExpanded="True">
  <Expander.Header>
    <TextBlock Text="Navigatie"
              FontSize="14"
              FontWeight="Bold">

```

```

        <TextBlock.LayoutTransform>
            <RotateTransform Angle="270" />
        </TextBlock.LayoutTransform>
    </TextBlock>
</Expander.Header>
</StackPanel>
    <Expander Header="Bieren">
        <views:BierenView DataContext="{Binding BierenVM}" />
    </Expander>
    <Expander Header="Brouwers">
        <TextBlock Text="Hier komt de Brouwers lijst view"/>
    </Expander>
</StackPanel>
</Expander>

```

9. Later vullen we de Brouwers expander met de lijst van Brouwers namen. Voorlopig werken we nu verder met de Bieren. Plaats een GroupBox zonder header text in de rechter kolom van de Grid met daarin een voorlopige TextBlock die we later vervangen door een nieuwe usercontrol:

```

<GroupBox Margin="10" Grid.Column="1" >
    <TextBlock Text="Hier komt BierDetailsView usercontrol"/>
</GroupBox>

```

10. De volledige Xaml code van MainWindow.xaml ziet er nu zo uit :

```

<Window x:Class="B_MockDataServiceWPFMVVM.MainWindow"
    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:B_MockDataServiceWPFMVVM"
    xmlns:viewmodels="clr-namespace:B_MockDataServiceWPFMVVM.ViewModels"
    xmlns:views="clr-namespace:B_MockDataServiceWPFMVVM.Views"
    mc:Ignorable="d"
    Title="Bieren & Brouwers" MinHeight="350" MinWidth="525">
    <Window.Resources>
        <viewmodels:MainViewModel x:Key="MainViewModel"/>
    </Window.Resources>
    <Grid DataContext="{StaticResource MainViewModel}">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="auto" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <Expander Padding="10"
            Margin="10"
            BorderBrush="DarkGray"
            ExpandDirection="Right"
            Grid.Row="0"
            Grid.Column="0"
            IsExpanded="True">
            <Expander.Header>
                <TextBlock Text="Navigatie"
                    FontSize="14"
                    FontWeight="Bold">
                    <TextBlock.LayoutTransform>
                        <RotateTransform Angle="270" />
                    </TextBlock.LayoutTransform>
                </TextBlock>
            </Expander.Header>

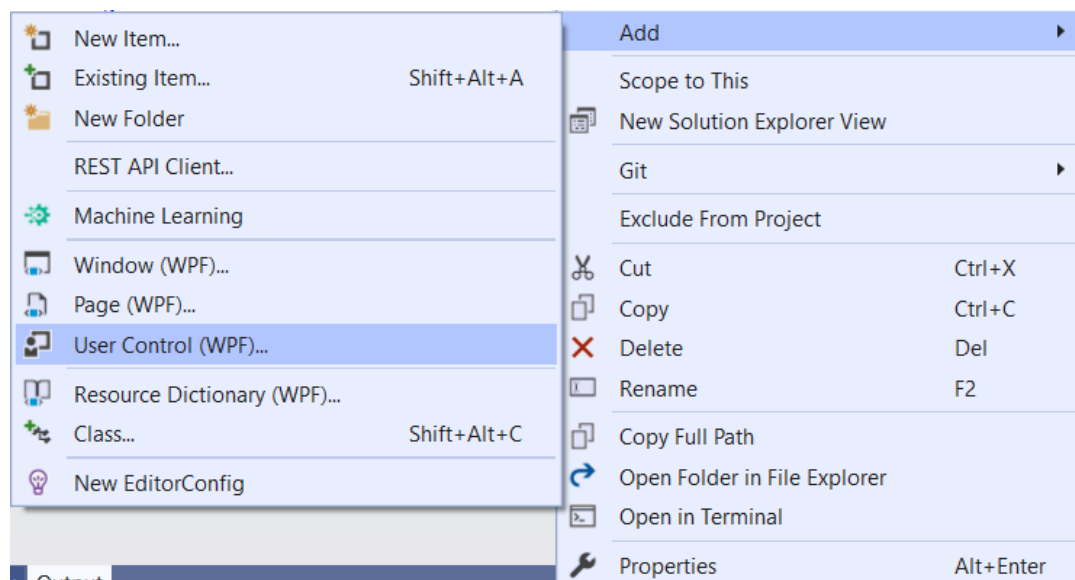
```

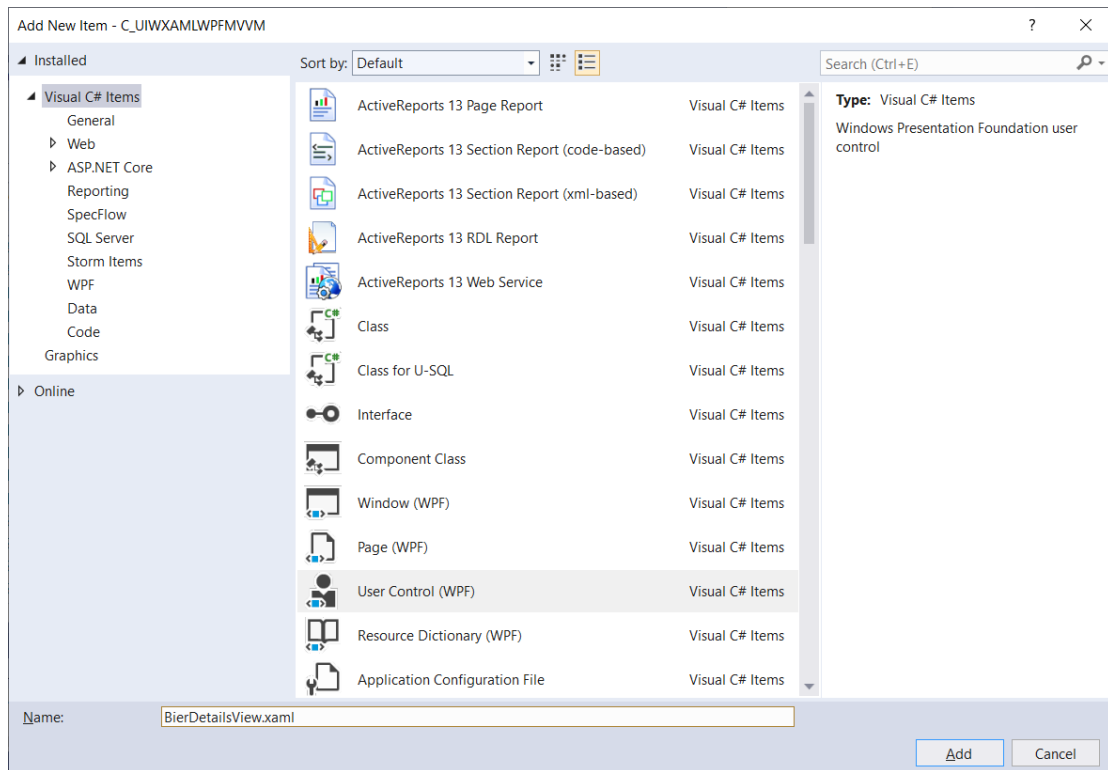
```

        </TextBlock>
    </Expander.Header>
</StackPanel>
    <Expander Header="Bieren">
        <views:BierenView DataContext="{Binding BierenVM}" />
    </Expander>
    <Expander Header="Brouwers">
        <TextBlock Text="Hier komt de Brouwers lijst view"/>
    </Expander>
</StackPanel>
</Expander>
    <GroupBox Margin="10" Grid.Column="1" >
        <TextBlock Text="Hier komt BierDetailsView usercontrol"/>
    </GroupBox>
</Grid>
</Window>

```

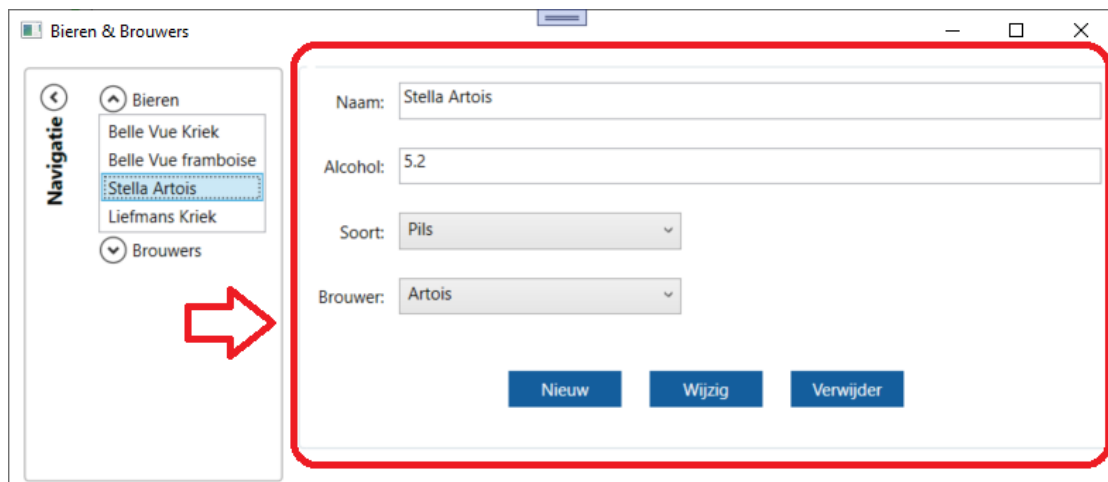
11. Maak een nieuwe usercontrol met naam BierDetailsView onder de folder Views:





12. In de BierDetailsView willen we de details tonen van het geselecteerde bier in de lijst van bieren:

13.



14. Open de BierDetailsView.xaml en voeg een Stackpanel toe in plaats van de de <Grid> en </Grid> tags, en daarbinnen een Grid met 3 kolommen en 9 rijen:

```
<UserControl x:Class="B_MockDataServiceWPFMVVM.Views.BierDetailsView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:B_MockDataServiceWPFMVVM.Views"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800">

    <StackPanel Margin="10">
        <Grid Margin="10">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="Auto" />
                <ColumnDefinition Width="6" />
                <ColumnDefinition Width="*" />
            </Grid.ColumnDefinitions>

            <Grid.RowDefinitions>
                <RowDefinition Height="Auto" />
                <RowDefinition Height="20" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="20" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="20" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="20" />
                <RowDefinition Height="Auto" />
            </Grid.RowDefinitions>

        </Grid>
    </StackPanel>
</UserControl>
```

15. In de eerste kolom plaatsen we nu de Labels en in de 3de kolom de TextBox velden waaraan we de Properties van een Bier object binden:

De rijen met hoogte=20 worden gebruikt om wat lege ruimte tussen de labels en textboxes te zetten. Op

```
<UserControl x:Class="B_MockDataServiceWPFMVVM.Views.BierDetailsView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:B_MockDataServiceWPFMVVM.Views"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800">

    <StackPanel Margin="10">
        <Grid Margin="10">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="Auto" />
                <ColumnDefinition Width="6" />
                <ColumnDefinition Width="*" />
            </Grid.ColumnDefinitions>

            <Grid.RowDefinitions>
                <RowDefinition Height="Auto" />
                <RowDefinition Height="20" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="20" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="20" />
                <RowDefinition Height="Auto" />
                <RowDefinition Height="20" />
                <RowDefinition Height="Auto" />
            </Grid.RowDefinitions>

        </Grid>
    </StackPanel>
</UserControl>
```

```

        <RowDefinition Height="Auto" />
        <RowDefinition Height="20" />
        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>
    <Label Grid.Row="0" Content="Naam:"
        HorizontalAlignment="Right" />
    <TextBox Text="{Binding SelectedBier.Naam,
UpdateSourceTrigger=PropertyChanged}" Grid.Column="2" />
    <Label Content="Alcohol:"
        Grid.Row="2"
        HorizontalAlignment="Right" />
    <TextBox Text="{Binding SelectedBier.Alcohol,
UpdateSourceTrigger=PropertyChanged}" Grid.Row="2"
        Grid.Column="2" />
    <Label Content="Soort:" Grid.Row="4"
        HorizontalAlignment="Right" />
    <ComboBox Grid.Row="4" Grid.Column="2" Width="200" HorizontalAlignment="Left"
ItemsSource="{Binding BierSoorten}" SelectedItem="{Binding SelectedBier.BierSoort,
UpdateSourceTrigger=PropertyChanged}" DisplayMemberPath="SoortNaam"/>
    <Label Content="Brouwer:" Grid.Row="6"
        HorizontalAlignment="Right" />
    <ComboBox Grid.Row="6" Grid.Column="2" Width="200" HorizontalAlignment="Left"
ItemsSource="{Binding Brouwers}" SelectedItem="{Binding SelectedBier.Brouwer,
UpdateSourceTrigger=PropertyChanged}" DisplayMemberPath="BrNaam"/>

    <StackPanel Grid.Row="8" Grid.Column="0" Grid.ColumnSpan="3"
HorizontalAlignment="Center" Orientation="Horizontal" Margin="10">
        <Button Margin="10"
            Grid.Row="8"
            Content="Nieuw" />

        <Button Margin="10"
            Grid.Row="8"
            Content="Wijzig" />
        <Button Margin="10"
            Grid.Row="8"
            Content="Verwijder" />
    </StackPanel>

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

```

De Text properties van de textboxes zijn gebonden aan het SelectedBier object (deze selected bier verwijst naar het geselecteerde item in de lijst van bieren). De bovenste textbox is gebonden aan de public property Naam van de geselecteerde bier (<TextBox

Text="{Binding SelectedBier.Naam ...}),

- De Text Property van de 2de textbox is gebonden aan de public property Alcohol (<TextBox Text="{Binding SelectedBier.Alcohol ...}).

- De eerste Combobox is gebonden aan de lijst van biersoorten (ItemsSource="{Binding BierSoorten}") en het geselecteerde item in de combobox is gebonden aan de public property BierSoort van het geselecteerde bier (SelectedItem="{Binding SelectedBier.BierSoort}). DisplayMemberPath="SoortNaam is de informatie die getoond wordt in de combobox, dus de soortnamen van de soorten bieren worden getoond.

-De onderste combobox is gebonden aan de lijst van Brouwers (ItemsSource="{Binding Brouwers}") . Het geselecteerde item is gebonden aan de Brouwer property van het geselecteerde Bier (SelectedItem="{Binding SelectedBier.Brouwer}) en in deze combobox worden de namen van Brouwers getoond (DisplayMemberPath="BrNaam")

UpdateSourceTrigger=PropertyChanged verwijst naar de trigger die zorgt dat er een notificatie wordt gestuurd (naar de gebonden ViewModel klasse) wanneer het event PropertyChanged zich voordoet (dit gebeurt bij een wijziging van Text in een Textbox en wijziging van selectie bij Combobox en ListBox)

16. Open **BierenViewModel.cs** en voeg de **Properties** die gebonden worden met de **TextBoxes** en **Comboboxes** met bijhorende **private fields** toe. We initialiseren eveneens de **biersoorten** en **brouwers** collecties in de constructor, zoals de **bieren**. Om het juiste element in de comboboxes te selecteren, zetten we de **SelectedBierSoort** en **SelecterBrouwer** in de constructor op de **SelectedBier.BierSoort** en **SelectedBier.Brouwer** :

```
public class BierenViewModel:ObservableObject
{
    private IDataService _dataService;
    private ObservableCollection<Bier> _bieren;
    private ObservableCollection<BierSoort> _biersoorten;
    private ObservableCollection<Brouwer> _brouwers;
    private BierSoort _selectedBierSoort;
    private Brouwer _selectedBrouwer;
    private Bier _selectedBier;
    public BierenViewModel(IDataService dataService)
    {
        _dataService = dataService;
        Bieren = new ObservableCollection<Bier>(dataService.GeefAlleBieren());
        BierSoorten = new ObservableCollection<BierSoort>(dataService.GeefAlleBierSoorten());
        if(Brouwers == null)
        {
            Brouwers = new ObservableCollection<Brouwer>(dataService.GeefAlleBrouwers());
        }
        if (SelectedBier != null)
        {
            SelectedBierSoort = SelectedBier.BierSoort;
            SelectedBrouwer = SelectedBier.Brouwer;
        }
    }

    public ObservableCollection<Brouwer> Brouwers {
        get { return _brouwers; }
        set { OnPropertyChanged(ref _brouwers, value); }
    }
    public ObservableCollection<Bier> Bieren {
        get { return _bieren; }
        set { OnPropertyChanged(ref _bieren, value); }
    }
    public Brouwer SelectedBrouwer
    {
        get { return _selectedBrouwer; }
        set { OnPropertyChanged(ref _selectedBrouwer, value); }
    }
    public Bier SelectedBier
    {
        get { return _selectedBier; }
        set { OnPropertyChanged(ref _selectedBier, value); }
    }
    public ObservableCollection<BierSoort> BierSoorten
    {
        get { return _biersoorten; }
        set { OnPropertyChanged(ref _biersoorten, value); }
    }
    public BierSoort SelectedBierSoort
    {
        get { return _selectedBierSoort; }
        set { OnPropertyChanged(ref _selectedBierSoort, value); }
    }
}
```

17. Nu gaan we de **BierDetailsView** usercontrol op de **MainWindow** plaatsen. Open hiervoor **MainWindow.xaml**, verwijder de **TextBlock** uit de groupbox onderaan en plaats hier de **BierDetailsView** :

....

```
</Expander>
<GroupBox Margin="10" Grid.Column="1" >
    <views:BierDetailsView DataContext="{Binding BierenVM}" />
</GroupBox>
```

```

    </Grid>
</Window>

```

18. De volledige xaml code van MainWindow is nu :

```

<Window x:Class="B_MockDataServiceWPFMVVM.MainWindow"
    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:B_MockDataServiceWPFMVVM"
    xmlns:viewmodels="clr-namespace:B_MockDataServiceWPFMVVM.ViewModels"
    xmlns:views="clr-namespace:B_MockDataServiceWPFMVVM.Views"
    mc:Ignorable="d"
    Title="Bieren & Brouwers" MinHeight="350" MinWidth="525">
    <Window.Resources>
        <viewmodels:MainViewModel x:Key="MainViewModel"/>
    </Window.Resources>
    <Grid DataContext="{StaticResource MainViewModel}">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="auto" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <Expander Padding="10"
            Margin="10"
            BorderBrush="DarkGray"
            ExpandDirection="Right"
            Grid.Row="0"
            Grid.Column="0"
            IsExpanded="True">
            <Expander.Header>
                <TextBlock Text="Navigatie"
                    FontSize="14"
                    FontWeight="Bold">
                <TextBlock.LayoutTransform>
                    <RotateTransform Angle="270" />
                </TextBlock.LayoutTransform>
            </TextBlock>
            </Expander.Header>
            <StackPanel>
                <Expander Header="Bieren">
                    <views:BierenView DataContext="{Binding BierenVM}" />
                </Expander>
                <Expander Header="Brouwers">
                    <TextBlock Text="Hier komt de Brouwers lijst view"/>
                </Expander>
            </StackPanel>
        </Expander>
        <GroupBox Margin="10" Grid.Column="1" >
            <views:BierDetailsView DataContext="{Binding BierenVM}" />
        </GroupBox>
    </Grid>
</Window>

```

19. Via de Binding van de DataContext van de BierDetailsView kunnen we op dezelfde manier als bij de BierenView de BierenViewModel binden aan de BierDetailsView (DataContext="{Binding BierenVM}"). BierenVM is namelijk een public property van de MainViewModel en verwijst naar een object van de klasse BierenViewModel.

20. Start de app op en je krijgt het volgende scherm te zien :

The screenshot shows a Windows application titled "Bieren & Brouwers". On the left, a "Navigatie" sidebar contains a tree view with "Bieren" selected, showing a list of beers: "Belle Vue Kriek", "Belle Vue framboise", "Stella Artois" (highlighted), and "Liefmans Kriek". The main content area on the right displays the details for "Stella Artois": "Naam: Stella Artois", "Alcohol: 5.2", "Soort: Pils" (dropdown), and "Brouwer: Artois" (dropdown). At the bottom of the main area are three buttons: "Nieuw", "Wijzig", and "Verwijder".

21. Wanneer je een ander bier selecteert in de lijst van bieren aan de linkerkant, zie je aan de rechterkant de waarden in de gebonden controls (TextBoxes en Comboboxes) mee veranderen. Dit gebeurt d.m.v. de Binding van ViewModel properties aan properties van de controls.

Run Nu de app, je ziet de listbox met bieren :

