

Tibo Menardo
r0804543



AAD
ImmoWin

odisee
DE CO-HOGESCHOOL

Inhoudstafel

WoningsModule.....	3
Feature 1: Databank initialisatie	4
Feature 2: Filteren en sorteren	5
Feature 3: Invoercontrole	6
Extra Feature: Conversion van euro naar ander valuta via API.....	6

WoningsModule

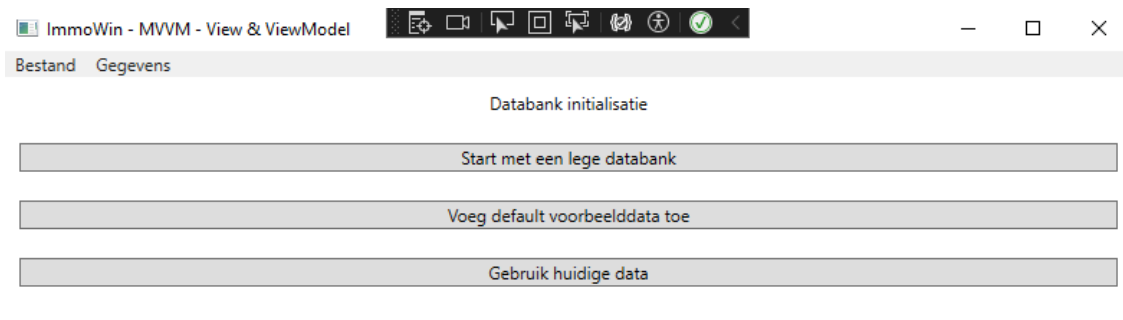
De woningsmodule bestaat uit twee views, een WoninglijstView en een WoningCommandDetailView. De WoninglijstView bevat een lijst van woningen en de Command buttons. De lijst van woningen bevat alle soorten woningen (de subklassen, Appartement en Huis). WoningCommandDetailView bevat de Command knoppen en is gemeenschappelijk voor beide subklassen. Voor het lezen en schrijven van data van een specifieke klasse wordt er gebruik gemaakt van de IDataObject interface. Binnen de WoningCommandDetailView is er een WoningDetailView die de data bevat van de geselecteerde woning. Deze is wel afhankelijk van de type woning.

The screenshot shows the ImmoWin application with two main panels. The left panel, titled 'Lijst woningen', contains a dropdown menu 'Kies een woningstype...', a 'Filter op eigenaar' section with a dropdown and a 'Reset' button, and a 'Sorteren op' section with buttons for 'Adres', 'Waarde', and 'Bouwdatum'. Below these are several property listings, including 'Huis Rijkhuis' and 'Appartement verd. 34'. The right panel, titled 'Appartement detailgegevens', shows details for a selected apartment, including 'Waarde: € 33.00', 'Bouwdatum: 12/21/2023', 'Verdieping: 34', and address fields for 'Straat', 'Nummer', 'Postnummer', and 'Gemeente'. At the bottom of the right panel are 'Bewaren' and 'Annuleren' buttons.

```
2 references
public interface IDataObject
{
    11 references
    void Lees();
    7 references
    void Schrijf();
    7 references
    Boolean DataObjectHasID();
}
```

Feature 1: Databank initialisatie

Bij het starten van de applicatie krijg je de keuze tussen drie opties voor de databank. Starten met een lege databank, voorbeelddata of gebruik huidige data. Bij start met een lege databank wordt alle data van elke tabel verwijderd. Bij voeg voorbeelddata wordt er eerst de data verwijderd vervolgens wordt de seeding methode van de custom DropCreateImmoWinContextContextIfModelChanges klasse opgeroepen. Bij gebruik huidige data gebeurt er geen actie.

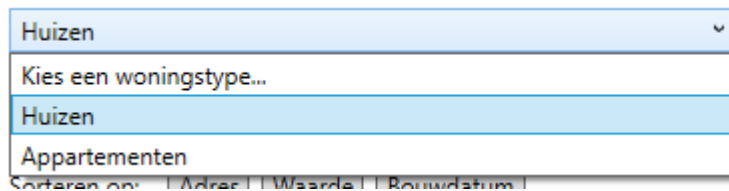


Bij de allereerste keer opstart wordt de custom DropCreateImmoWinContextContextIfModelChanges klasse opgeroepen voor Seeding.

```
17 references
public ImmoWinContext() : base("DbImmoWin")
{
    Database.SetInitializer(new DropCreateImmoWinContextContextIfModelChanges());
}
```

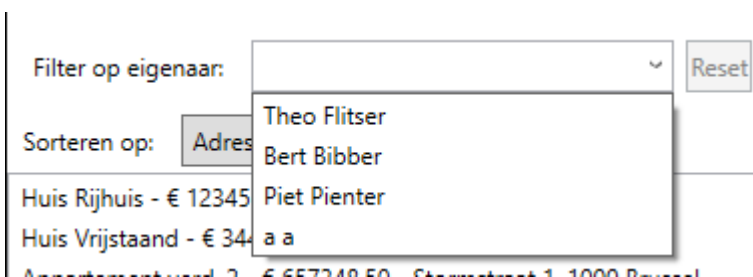
Feature 2: Filteren en sorteren

Aangezien de lijst zowel Huizen als Appartementen bevat kan er gekozen worden tussen niet filteren of alleen Huizen/ Appartementen.




A screenshot of a web application showing a dropdown menu. The menu is open, displaying the following options: 'Huizen' (selected), 'Kies een woningstype...', 'Huizen', and 'Appartementen'. Below the menu, the text 'Sorteren op: Adres | Waarde | Bouwdatum' is visible.

Filteren op eigenaar lukt ook. Er wordt in een ComboBox een lijst van alle klanten getoond. Naast de ComboBox is er een reset knop voor de filter.



A screenshot of a web application showing a filter interface. The 'Filter op eigenaar:' label is followed by a dropdown menu with a list of names: 'Theo Flitser', 'Bert Bibber', 'Piet Pienter', and 'a a'. To the right of the dropdown is a 'Reset' button. Below the filter, the 'Sorteren op:' label is followed by a button labeled 'Adres'. The main content area shows a list of housing types and prices: 'Huis Rijhuis - € 12345' and 'Huis Vrijstaand - € 34567'.

Er kan gesorteerd worden op drie eigenschappen Adres, Waarde en Bouwdatum. Dit wordt gedaan door custom CompareTo methodes in de Woning klasse.



A screenshot of a web application showing a sorting interface. The 'Sorteren op:' label is followed by three buttons: 'Adres', 'Waarde', and 'Bouwdatum'.

Feature 3: Invoercontrole

Voor invoercontrole heeft elke klasse een validate methode die een lijst van exceptions meegeeft. Die lijst wordt wanneer er op de bewaar knop gedrukt wordt opgevangen en de message van elk exception wordt samen getoond in een messagebox. Elke klasse dat een validatie heeft erft over van de IValideerbaar interface.

```
1 reference
private void KlantWijzigingBewarenCommandExecute()
{
    StringBuilder sb = new StringBuilder();
    foreach (Exception ex in Klant.Validate())
    {
        sb.AppendLine(ex.Message);
    }
    if (sb.Length > 0)
    {
        MessageBox.Show(sb.ToString(), "Validation Errors", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    Klant.Schrijf();
}
```

```
3 references
public interface IValideerbaar
{
    11 references
    List<Exception> Validate();
}
```

Extra Feature: Conversion van euro naar ander valuta via API

Om de convertie rate te krijgen van euro t.o.v. andere valuta's wordt er een API gebruikt (currencyapi.com). Er wordt alle mogelijke valuta's ten opzichte van de euro verwacht, vervolgens worden deze bijgehouden in file, "APIResponse.json". Er zijn twee klassen om de data van de API request op te vangen. Aangezien er een gratis Api gebruikt wordt zijn er maar een gelimiteerd aantal requests per maand. Hierdoor wordt er gechecked wanneer de laatste request gedaan is. Als dit meer dan 12 uur geleden was wordt er een nieuwe request gedaan. Deze controle wordt getriggerd elke keer dat een WoningDetailView wordt aangemaakt.

