Copilot

Certainly! In .NET MAUI, you can use the `ObservableCollection` along with the CommunityToolkit.Mvvm library to handle data binding and UI updates. Here are the steps:

1. **Model (Student)**:
   - Define your model class (e.g., `Student`) with properties that you want to display in the UI. Make sure to decorate the relevant property with `[ObservableProperty]`.

   ```
   public partial class Student : ObservableObject
   {
       public int Id { get; set; }
       public string Name { get; set; }
       [ObservableProperty] bool IsRegistered;
   }
   ```

2. **ViewModel (MyViewModel)**:
   - Create your view model class (e.g., `MyViewModel`) and use an `ObservableCollection<Student>` to hold your data.
   - Decorate the `ObservableCollection` property with `[ObservableProperty]`.
   - Implement a command (e.g., `UpdateRegistrationsCommand`) that modifies the `IsRegistered` property of each student.

   ```
   public partial class MyViewModel : ObservableObject
   {
       [ObservableProperty] ObservableCollection<Student> Students = new();

       [RelayCommand]
       void UpdateRegistrations()
       {
           foreach (var item in Students)
               item.IsRegistered = true;
       }
   }
   ```

3. **XAML**:
   - In your XAML, bind the `CollectionView` to the `Students` property in your view model.
   - Use a `DataTemplate` to define how each student should be displayed.
   - For example:

   ```
   <ContentPage...>
       <Grid RowDefinitions="*,50">
           <CollectionView Grid.Row="0" ItemsSource="{Binding Students}">
               <CollectionView.ItemTemplate>
                   <DataTemplate x:DataType="model:Student">
                       <Grid ColumnDefinitions="*,30">
                           <Label Grid.Column="0" Text="{Binding Name}" />
                           <Image Grid.Column="1" Source="checkmark.png" IsVisible="{Binding IsRegistered}" />
                       </Grid>
                   </DataTemplate>
               </CollectionView.ItemTemplate>
           </CollectionView>
           <Button Grid.Row="1" Text="Update Data" Command="{Binding UpdateRegistrationsCommand}" />
       </Grid>
   </ContentPage>
   ```

4. **Observations**:
   - Ensure that your `IsRegistered` property changes are correctly propagated to the UI. If not, consider checking other elements (e.g., a `CheckBox`) to verify whether the issue is specific to `IsVisible`.

    - The CommunityToolkit.Mvvm library handles `INotifyPropertyChanged` logic, so you don’t need to handle the change manually .
    - Remember to include the necessary namespaces and references for the CommunityToolkit.Mvvm library.

I hope this helps! Let me know if you have any further questions. 😊