

VRAAG 1: COCKTAILBAR

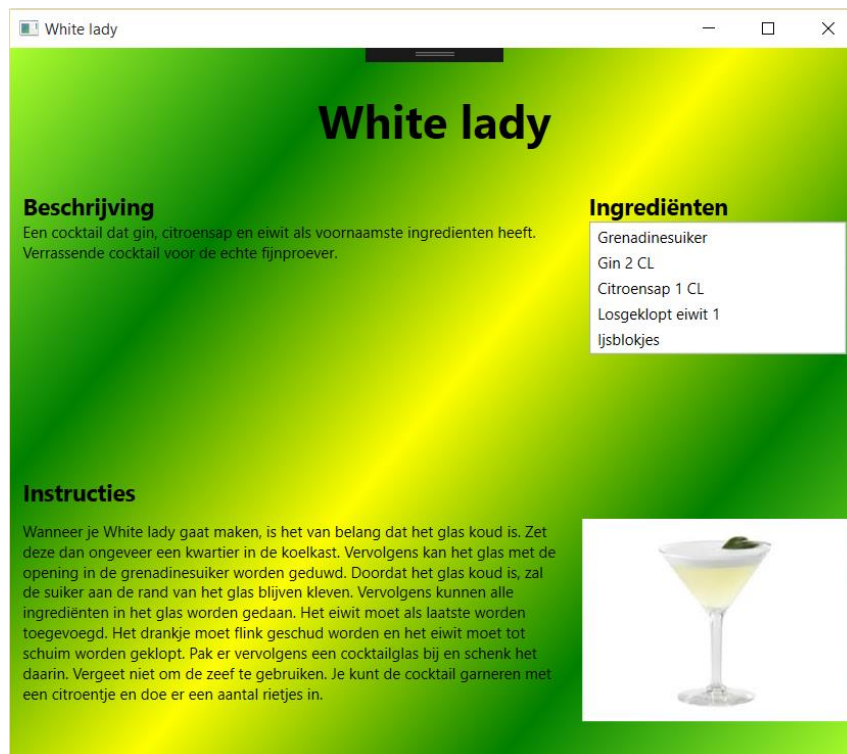
/25

Open de solution 'Cocktailbar - Skelet'. Daarin vind je 2 verschillende projecten. Je schrijft een WPF applicatie die cocktails ophaalt uit een database met behulp van **ADO .NET**. Elke cocktail heeft een id, een naam, een beschrijving, instructies en ingrediënten.



Cocktail overzicht

Als je dubbelklikt op een cocktail in het overzicht, worden alle gegevens van de cocktail getoond alsook de afbeelding van die cocktail:

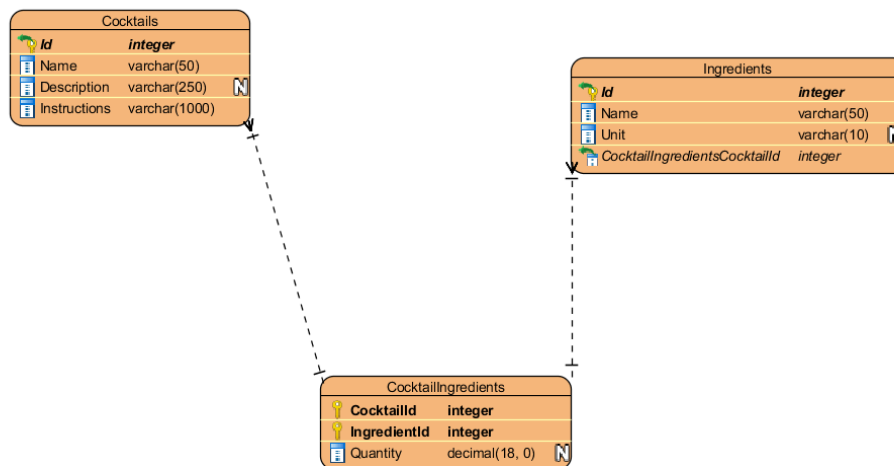


Details cocktail

Vul het project aan als volgt (TIP: Lees de comments in het skelet, laat deze comments ook staan!):

1. Je hebt een sql script gekregen om de database aan te maken met de dummy data: **CreateCocktails.sql**. Voer dit script uit om de database aan te maken en te vullen met data.

Het ERD van de database ziet er als volgt uit :



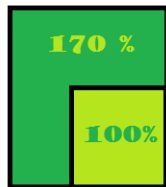
2. Gegeven zijn de verschillende domeinklassen in het CocktailBarData project: **Cocktail**, **CocktailIngredient** en **Ingredient**. Deze zijn gebaseerd op de database om de verschillende properties voor deze klassen te schrijven.
3. Vervolledig de klasse **ConnectionFactory**. Lees de connectionstring uit je configuratiefile zodat je connectie kan maken met de database.
4. Schrijf de nodige repository klassen met de juiste methodes om je SQL queries uit te voeren op je database. Zo moet je bijvoorbeeld een methode voorzien om alle cocktails op te halen uit de database: CocktailsRepository, CocktailsIngredientsRepository en IngredientsRepository. Tip: om te controleren of een waarde uit de database 'null' is, kan je de *IsDBNull* methode van *SqlDataReader* gebruiken.
5. Vervolledig de XAML code

MainWindow:

- Zorg dat je MainWindow er uit ziet zoals in bovenstaande afbeelding. Je gebruikt **databinding** om de verschillende elementen van content te voorzien! De afbeelding voor de knop vind je in de Images folder.
- Gebruik een *ListView* om een overzicht van de cocktails uit de database te tonen. Tip: stop een *GridView* object in de *View* property van de *ListView*.
- Wanneer je dubbelklikt op een cocktail, ga je naar het CocktailDetailsWindow.

CocktailDetailsWindow:

- Een gedeelte van de XAML code is reeds gegeven.
- Vul de databinding statements aan.
- Voorzie de achtergrond zoals op de screenshot. De gebruikte kleuren zijn 'GreenYellow', 'Green' en 'Yellow'.
- Gebruik een Converter om de image te tonen. De naam van de image is de naam van de cocktail zonder spatie(s) met ".jpg" als extensie. Alle images bevinden zich in Images folder.
- Wanneer er met de muis over de image wordt bewogen, vergroot de image met 70%. Dit moet geschaald worden vanuit de rechter onderhoek.

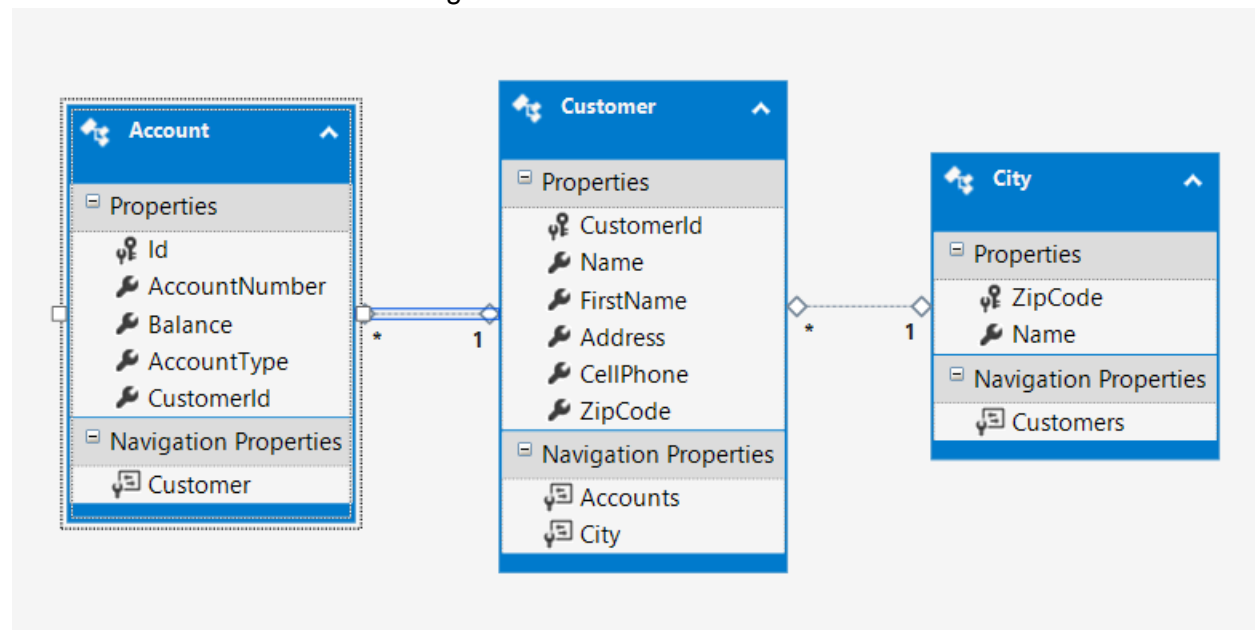


VRAAG 2: BANKAPPLICATION

/20

Open de solution 'BankApplication - Skelet'. Daarin vind je 5 verschillende projecten. Je schrijft een WPF applicatie die Customers van een bank kan ophalen uit een database. Als je een Customer selecteert worden de rekeningen (accounts) van deze klant weergegeven. Een klant kan geld overzetten van een van zijn rekeningen naar een andere (van zijn eigen rekeningen). Hiervoor maak je gebruik van **Entity Framework**. De XAML heb je reeds cadeau gekregen. Vul de solution aan als volgt:

1. Je schrijft 3 verschillende domeinklassen: **Customer**, **Account** en **City**.
Het type rekening is een property van het type **AccountType**. Een enum die reeds gedefinieerd is in de startoplossing.
Gebruik het type **decimal** voor de balans van een rekening.
Je vindt hier het klassediagram



2. De naam van deze database moet **BankDB** zijn. Maak gebruik van de **FluentAPI** om de volgende zaken in te stellen:
 - De primary key van City: **ZipCode**
 - De relatie tussen Account en Customer
 - De relatie tussen Customer en CityJe mag **geen data-annotaties** gebruiken!
Gebruik de nodige tools om een migratie aan te maken en de database te genereren.
3. Het **WPF project** maakt gebruik van de **CustomerRepository**, **AccountRepository** en de **CityRepository** klassen om objecten op te halen of weg te schrijven naar de database. In die klasse zijn de verschillende methodes al geschreven. Vul deze methodes aan.

4. Vul je database met dummy data. Als je datamodel overeenkomt met de modeloplossing, kan je hiervoor gebruik maken van het script *Bank.sql*. Indien het script fouten geeft kan je best rechtstreeks in de database dummy data ingeven.

Wanneer de applicatie start, wordt het **CustomerWindow** getoond. In dit venster krijg je een overzicht van alle klanten.

Name	Firstname	Address	City	Cellphone
Willems	Marijke	Beemdstraat 15	Riemst	0499/12.32.54
Hendriks	Wesley	Vaartstraat 31	Hasselt	0494/16.54.87
Hermans	Kris	Overweg 115	Tongeren	0476/12.82.15

Show Accounts Save Add Customer

Wanneer een waarde wordt aangepast van een klant, wordt dit in de database opgeslagen zodra er op de 'Save' knop wordt geklikt.

Wanneer er op de 'Add Customer' knop wordt geklikt, wordt er een extra lege lijn toegevoegd, waarin de gegevens van de nieuwe klant kunnen worden ingegeven.

TIP: gebruik de CanUserAddRows property van de datagrid.

Na het klikken op de 'Save' knop, wordt deze nieuwe klant aan de database toegevoegd en kan je geen nieuwe gegevens meer toevoegen tot je opnieuw op de 'Add Customer' knop klikt.

Name	Firstname	Address	City	Cellphone
Willems	Marijke	Beemdstraat 15	Riemst	0499/12.32.54
Hendriks	Wesley	Vaartstraat 31	Hasselt	0494/16.54.87
Hermans	Kris	Overweg 115	Tongeren	0476/12.82.15
Lambrechts	Rita			

Show Accounts

Save

Add Customer

Wanneer een klant is geselecteerd en er wordt op de 'Show Accounts' button geklikt, wordt er naar het **AccountsWindow** genavigeerd. Alle rekeningen van de geselecteerde klant worden getoond.

Account Number	AccountType	Balance
123-654651-156	PaymentAccount	78000.00
351-854321-123	YouthAccount	1600.00
546-565435-653	PremiumAccount	30000.00

Add Account

Save

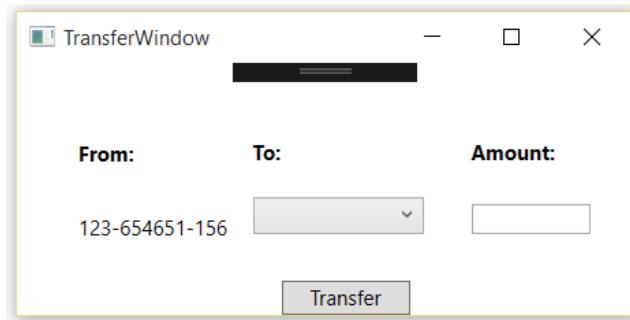
Transfer Amount

Bij het klikken op de 'Add Account' button, wordt er opnieuw een lege regel voorzien waar de accountgegevens ingevuld kunnen worden. Bij het klikken op de 'Save' knop wordt deze nieuwe rekening aan de database toegevoegd en kunnen er geen nieuwe gegevens meer worden toegevoegd tot er opnieuw op de 'Add Account' button wordt geklikt.

Het AccountType is een Enum. De code om deze enum te tonen in een combobox is reeds aanwezig.

Wanneer je een rekening selecteert, kan je op de 'Transfer Amount' knop klikken om een bedrag over te zetten naar een andere (eigen) rekening.

Je navigeert dan naar het **TransferWindow**:

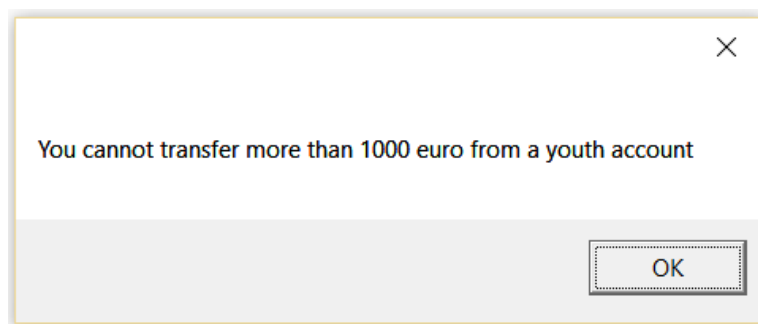
The screenshot shows a window titled "TransferWindow" with standard Windows window controls (minimize, maximize, close). Inside the window, there are three labels: "From:", "To:", and "Amount:". Under "From:", the account number "123-654651-156" is displayed. Under "To:", there is a dropdown menu with a downward arrow. Under "Amount:", there is an empty text input field. At the bottom center of the window is a button labeled "Transfer".

De combobox bevat alle rekeningen van de geselecteerde klant (behalve de rekening vanwaar de transfer gebeurt, dus '123-654651-156' mag niet meer voorkomen in de combobox. Maak gebruik van Linq om dit te bewerkstelligen). De rekeningen van de klant worden meegegeven in de constructor. Je hebt dus **geen repositories** nodig in het TransferWindow.

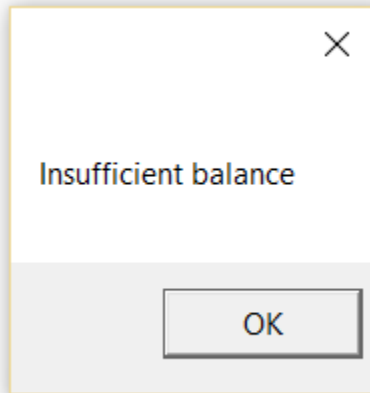
Wanneer er op de Transfer button wordt geklikt, wordt het ingevulde bedrag overgeschreven naar de gekozen rekening uit de combobox.

Na het sluiten van dit venster, zijn de bedragen in het **AccountsWindow** aangepast.

5. Bij het klikken op de 'Transfer' knop maak je gebruik van de **TransferMoney** methode van de **AccountManager** class die je kan vinden in het project **Bank.Business**. In deze methode stop je de volgende logica:
 - a. Er mag geen bedrag hoger dan 1000 euro worden overgeschreven vanaf een YouthAccount in het **TransferWindow**. Je kan dus na het selecteren van een YouthAccount geen bedrag hoger dan 1000 overschrijven naar een andere rekening. De methode gooit dan een **InvalidTransferException** en in de UI krijg je de volgende foutmelding:



- b. Er kan geen bedrag worden overgeschreven dat hoger is dan het bedrag wat op de rekening staat. Zoniet gooit de methode een **InvalidTransferException** en in de UI krijg je de volgende foutmelding:



6. Schrijf testen in het testProject **Bank.Test** voor bovenstaande business methode.

De testmethodes zijn reeds aangemaakt. Je dient enkel de testcode aan te vullen. Zorg dat alle testen slagen.

VEEL SUCCES !!