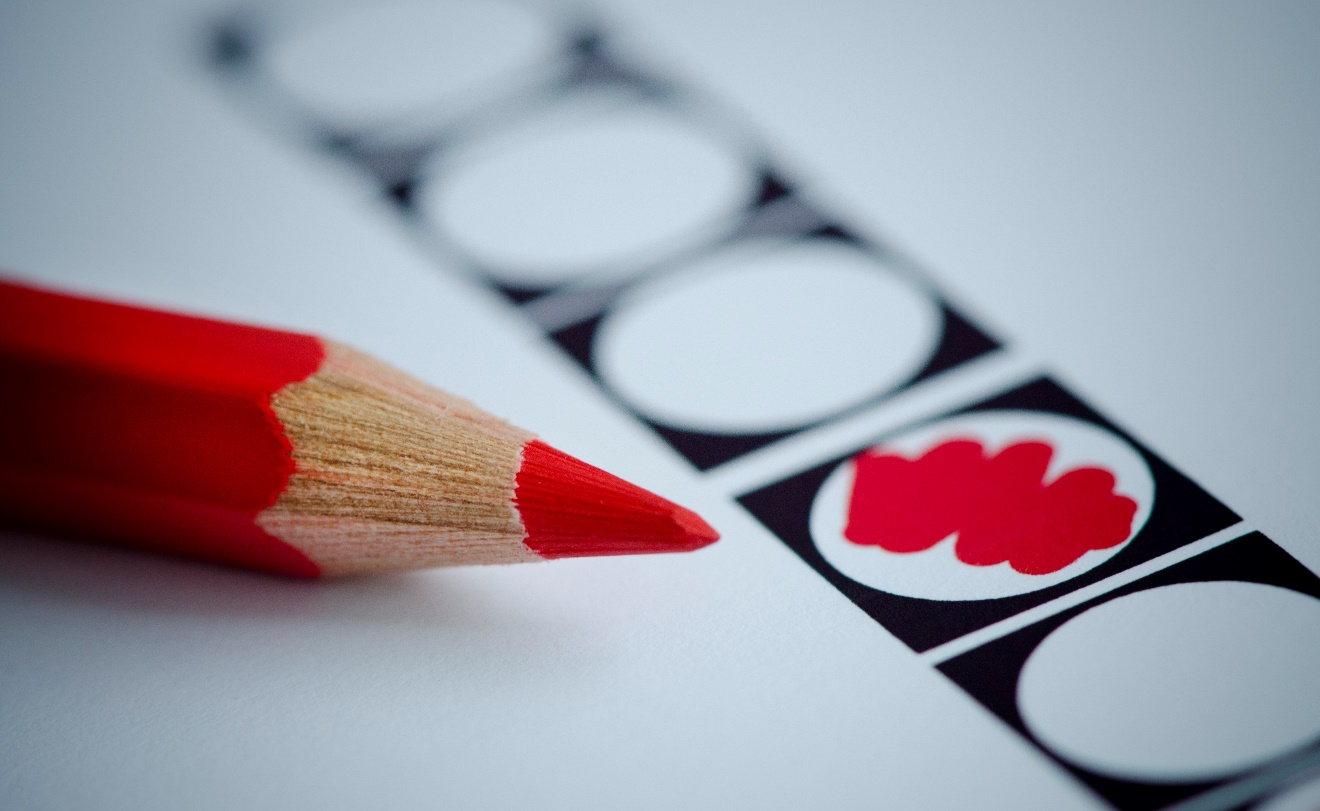
ontwerpdocument

Live performance

Michelle Broens | S21M | 21-06-2017

Inhoudsopgave

[Inleiding 2](#_Toc485825206)

[Begrippen 3](#_Toc485825207)

[Architectuur 4](#_Toc485825208)

[Klassendiagram 5](#_Toc485825209)

[Klassendiagram bijlage 6](#_Toc485825210)

[Database ontwerp 7](#_Toc485825211)

[Database ontwerp bijlage 8](#_Toc485825212)

[Tabellen 9](#_Toc485825213)

[Uitslag 9](#_Toc485825214)

[Verkiezing 9](#_Toc485825215)

# Inleiding

In dit document wordt er dieper ingegaan op de applicatie die in het document beschreven staat. Hier komen de meer gecompliceerde uitwerkingen die bijna meteen geïmplementeerd kunnen worden. Deze onderdelen zijn:

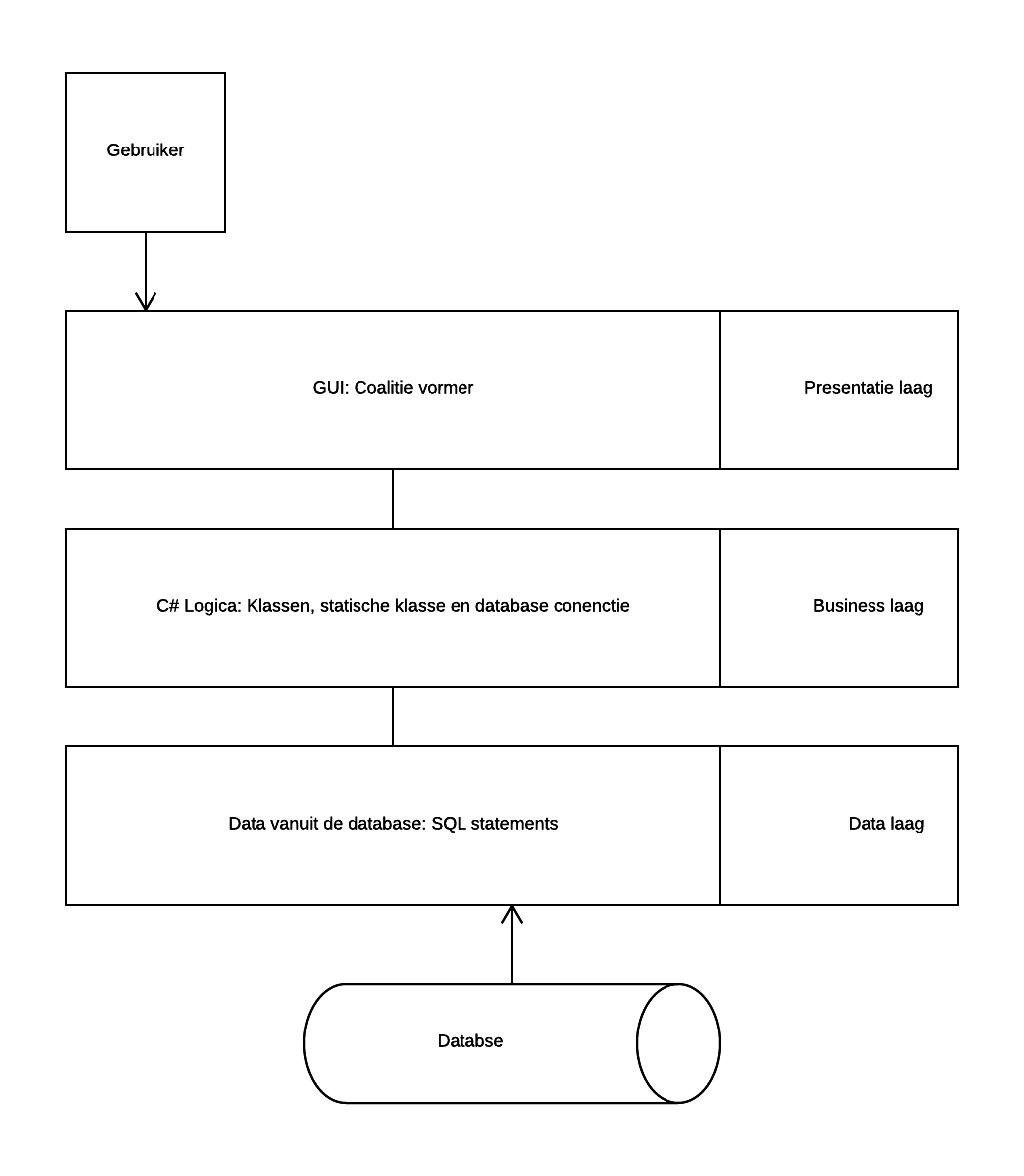
* Begrippen: dit is een lijst met begrippen hier staan woorden die de lezer misschien niet begrijpt of misschien niet helemaal begrijpt.
* Architectuur: dit geeft de structuur en communicatie van de applicatie weer in de vorm van een diagram
* Database ontwerp: hierin komt een diagram van de tabellen en attributen die in de database gezet worden, samen met een bijlage van dit diagram.
* Klassendiagram: dit geeft aan welke klassen, properties en methodes er in de applicatie gebruikt gaan worden.
* Infrastructuur: hier wordt in een diagram beschreven hoe de infrastructuur in elkaar zit, dus de server waar eventueel de database opstaat, de clients waarop de applicatie zou kunnen draaien.

# Begrippen

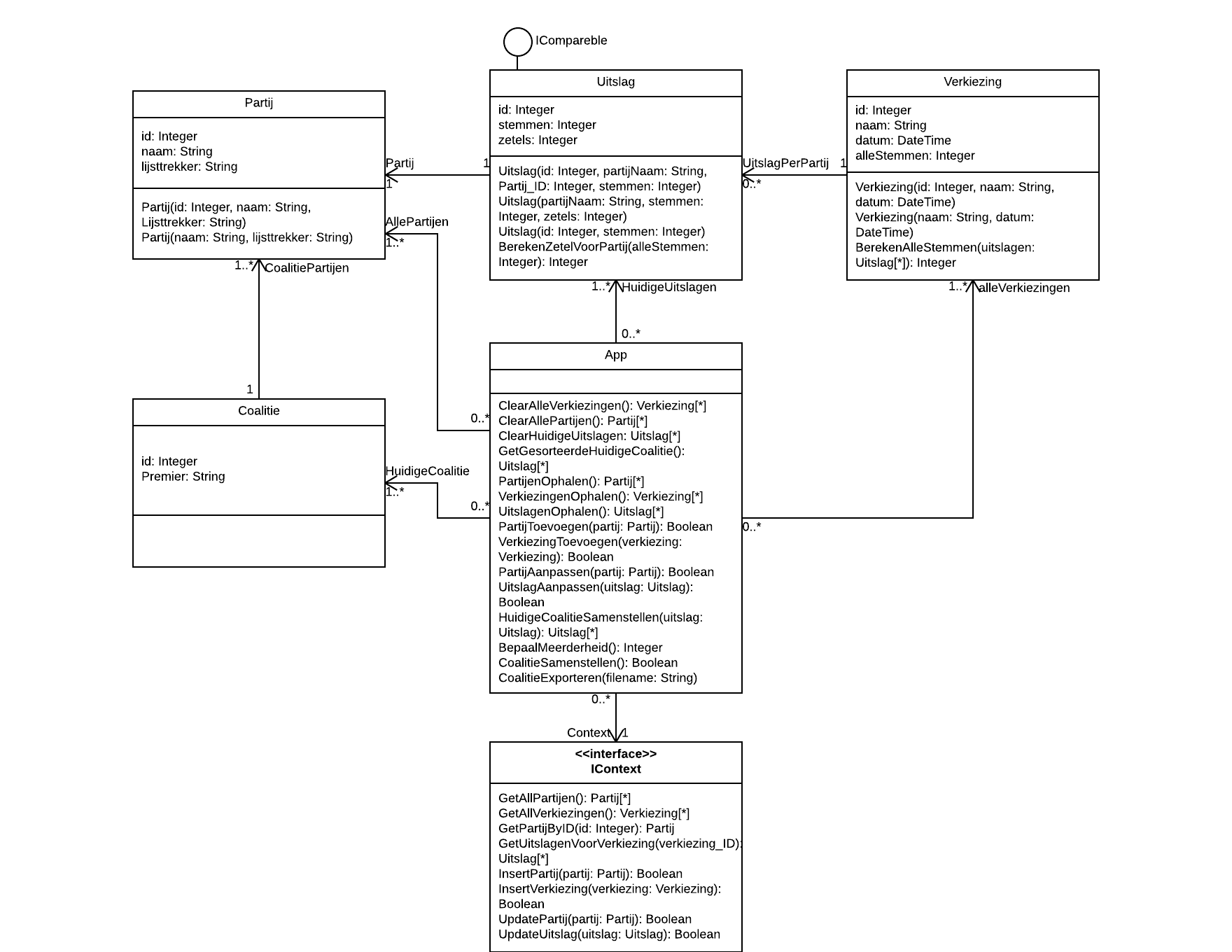
|  |  |
| --- | --- |
| **Begrip** | **Uitleg** |
| Property | In een property worden waardes opgeslagen om een privéveld buiten de klasse te kunnen gebruiken. |
| Methode | Een handeling die meerdere keren aangeroepen kan worden. |
| Infrastructuur | Een verzameling van voorzieningen die nodig is voor het transport van data. |
| Clients | Een client is een werkstation dus een pc of een laptop, die verbonden zit met de rest van de infrastructuur. |
| Query | Een query is een verzoek voor data/informatie van een tabel (of meerdere tabellen) aan de database. |

# Architectuur

De gebruiker logt in op het inlogscherm (presentatie laag) de gegevens worden doorgestuurd naar de business laag. Deze gegevens worden door middel van een query (data laag) naar de database gestuurd.



# Klassendiagram

Hier komen alle klassen die geïmplementeerd worden in de applicatie met onder het diagram nog een bijlage met een verdere uitleg.

## Klassendiagram bijlage

Ik ben ervan uit gegaan dat alle attributen een private setter hebben, dus read only zijn, ik heb dit niet in dit diagram gezet omdat ik dit alleen belanrijk vindt bij attributen zoals salaris om aan te geven dat deze echt nóóit een public setter mag hebben.

Verder heb ik besloten dat ik géén gebruik ga maken van het repository pattern, maar een kleine “hack” ga gebruiken vergelijkbaar met dit pattern, omdat het mij te veel tijd kost om het pattern goed te implementeren en hier heb ik nu geen tijd voor. De interface die ik hiervoor gebruik is de **IContext**.

De **App** klasse is de klassen die de informatie van de logic laag doorgeeft naar de UI laag. Deze heeft ook methodes die door de data laag uitgevoerd worden door middel van de **IContext** interface. Deze methodes dienen ervoor dat de input van de user doorgestuurd kan worden naar de logic laag en de data laag.

Een Uitslag heeft één Partij, want de uitslagen zijn gegeven per Partij. De **Uitslag** klasse heeft de stemmen en het aantal zetels voor één **Partij**, een **Verkiezing** heeft een lijst van **Uitslagen** die samen alle **Uitslagen** voor een verkiezing vormen.

# C:\Users\miche\Downloads\DBO - Page 1 (2).pngDatabase ontwerp

## Database ontwerp bijlage

Ik heb gebruik gemaakt van identity voor het automatisch optellen van de ID’s. Dit staat niet in het model omdat ik in het FUN2 dictaat niets kon vinden over hoe dit genoteerd moet worden.

Ik ben ervan uit gegaan dat de **Naam** van de partij uniek is en dat een **Lijsttrekker** niet een lijsttrekker kan zijn voor meerdere partijen, daarom zijn deze waardes uniek.

Ik heb ervoor gekozen om een **Coalitie** geen naam te geven, maar de naam van de premier te gebruiken als naam voor de coalitie.

**Zetels** is nullable omdat er bij gemeenteraadverkiezingen is van zetels, deze wordt daar niet meegenomen.

Er is een koppeltabel tussen **Partij** en **Coalitie** gekomen, dit is gedaan om de meer op meer relatie die daar in het EER zit op te lossen. Om een coalitie te vormen zijn meerdere partijen nodig, maar er zijn ook partijen die niet opgenomen worden in een coalitie. Om een coalitie te vormen worden er dus een aantal partijen uitgekozen en in deze koppeltabel gezet.

De andere tabel die ik toegevoegd heb die niet in het EER staat is **Uitslag**, dit lost de meer op meer relatie op tussen **Partij** en **Verkiezing**. Deze tabel laat per partij zien hoe veel stemmen en zetels de desbetreffende partij heeft. Een verkiezing heeft dus meerdere uitslagen omdat de uitslag per partij opgeslagen wordt in plaats van één uitslag voor de gehele verkiezing. Ik heb ervoor gekozen om dit zo op te lossen, want er was een tabel nodig die het aantal stemmen en aantal zetels op nam, dus de echte uitslag van de verkiezing voor de partij.

Er kan een **Verkiezing** zijn zonder **Uitslag**, bijvoorbeeld als de **Uitslag** nog niet berekend is. Een uitslag behoort daarentegen altijd tot een verkiezing omdat er nooit een uitslag zal zijn als er geen verkiezing is.

## Tabellen

Dit zijn niet alle tabellen die ik heb in mijn database, maar ik heb in het kader van tijd besloten om er twee uit te werken om aan te tonen dat ik weet hoe dit moet.

### Uitslag

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribuut** | **Type** | **Beschrijving** | **Verplicht** |
| ID | Int | De primary key voor deze tabel | Ja |
| Verkiezing\_ID | Int | Een foreign key die verwijst naar het ID van de verkiezing tabel | Ja |
| Partij\_ID | Int | Een foreign key die verwijst naar het ID van de partij tabel | Ja |
| Stemmen | Int | Het aantal stemmen dat een partij heeft | Ja |
| Zetels | int | Het aantal zetels dat een partij heeft | Nee |

### Verkiezing

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribuut** | **Type** | **Beschrijving** | **Verplicht** |
| ID | Int | De primary key voor deze tabel | Ja |
| Naam | varchar(40) | De naam van de verkiezing, deze is uniek, dus er zijn geen verkiezingen met twee keer dezelfde naam | Ja |
| Datum | date | De datum wanneer de verkiezing gehouden, deze is uniek, er zijn geen twee verkiezingen geweest op dezelfde datum | Ja |