

Casus Workshops

De FIA (de organisatie van de racesport) wil een systeem waarin ze bij kunnen houden per raceklasse en seizoen van die raceklasse, welke team hierin gereden hebben. (Bijvoorbeeld: In de formule 1 reden in 2018 Red Bull, Ferrari, Haas en Mercedes, in de Nascar reden in 2018 Joe Gibbs Racing, Penske En Haas). Jij bent gevraagd hiervoor een systeem te ontwerpen en implementeren. Hierbij moet je rekening houden met de volgende eisen:

- Een raceklasse is uniek. Verder heeft een raceklasse o.a. een naam, aantal races, het jaar dat de klasse voor het eerst gereden is en of de klasse nog actief is.*
- Een seizoen is uniek binnen een raceklasse (seizoen 2018 kan bijvoorbeeld maar 1x bestaan binnen Formule 1, maar kan ook binnen Nascar bestaan). Een seizoen bevat een aantal races wat gereden moet worden en (indien het seizoen geëindigd is) een kampioen.*
- Een raceteam heeft een unieke naam en kan in meerdere klassen deelnemen. Verder heeft een team een standplaats (stad, land), hoofdsponsor, oprichtingsjaar en een directeur*
- Het moet mogelijk zijn nieuwe raceklassen, seizoenen en teams toe te voegen, te verwijderen en te updaten. Verder moet het mogelijk zijn seizoenen aan raceklassen toe te voegen en verwijderen en teams aan seizoenen toe te voegen en te verwijderen.*

Opdracht: Maak een conceptueel model, bijv. een EER en een domeinmodel, bijv. een klassendiagram voor bovenstaande casus. Implementeer de casus en laat zien dat je oplossing werkt door middel van unit tests. Het is niet de bedoeling de data op te slaan of op een UI te tonen, het gaat hierbij alleen om de logica.

Stappenplan

Het stappenplan kan gebruikt worden voor elk model (conceptueel model, domeinmodel, database ontwerp, architectuur) wat gemaakt wordt, waar sommige stappen niet in elk model relevant zijn

1. Bepaal welke entiteiten er zijn
2. Bepaal welke entiteiten gerelateerd zijn (bepaal nog niet hoe de relaties er uit zien, maar alleen dat ze gerelateerd zijn)
3. Ontwerp de relaties verder:
 - a. Bepaal hoe de relatie van entiteit 1 naar entiteit 2 is:
 - i. Wat voor relatie is het?
 - ii. Is het een verplichte of optionele relatie?
 - iii. Wat is de multiplicititeit?
 - b. Bepaal hoe de relatie van entiteit 2 naar entiteit 1 is:
 - i. Wat voor relatie is het?
 - ii. Is het een verplichte of optionele relatie?
 - iii. Wat is de multiplicititeit?
4. Bepaal de attributen
5. Bepaal de operaties – in geval van een domeinmodel
6. Verantwoording van het diagram:
 - a. Wat houdt het diagram in (wat is het verhaal)
 - i. Bijvoorbeeld: “Dit is een schematische weergave van hoe het systeem er uit ziet”
 - ii. Als je maar een deel van het systeem ontwerpt, geef je dat aan: (bijvoorbeeld “In onderstaand domeinmodel is te zien hoe de use cases terugkomen in de logicalaag van de applicatie”).
 - b. Bespreek aannames en keuzes, waarbij aannames uiteindelijk weggewerkt zouden moeten worden (bijvoorbeeld door ze te controleren bij de opdrachtgever)
 - c. Enkele extra aandachtspunten:
 - i. Overduidelijke / triviale zaken neem je niet op: Dat een entiteit Persoon de attributen Naam, Leeftijd, Adres heeft is logisch en hoeft niet verantwoord te worden.
 - ii. Samengestelde entiteiten kunnen toegelicht worden (denk aan overerving of het samenvoegen van 2 entiteiten in het conceptueel model naar 1 entiteit in het domeinmodel)
 - iii. Als je een aanname of keuze maakt, noteer deze dan meteen, dat voorkomt dat je dit later vergeet

In dit stappenplan kunnen bepaalde stappen in een willekeurige volgorde uitgevoerd worden (bijvoorbeeld eerst attributen bepalen, voordat je de relaties bepaalt). Wel zijn een aantal stappen vast: Bepalen welke entiteiten er zijn is altijd stap 1, bepalen welke entiteiten gerelateerd zijn komt altijd voor de entiteiten verder ontwerpen, en de verantwoording schrijven is altijd de laatste stap

Verder is het slim de eerste versie(s) van een ontwerp altijd ‘offline’ op een whiteboard te doen, zodat je ruzie met de tool voorkomt in het ontwerpproces.