

Functioneel Ontwerp

1 Inleiding

In dit bestand wordt het functioneel ontwerp van het formule 1 geschreven. Hierbij zal worden gekeken naar de functionele en non-functionele eisen die van belang zijn bij dit project. Tijdens het project worden er usecases opgesteld die gerealiseerd moeten worden. Wanneer er aan een usecase word gewerkt word deze fullydressed uitgewerkt.

Verder wordt er in dit document verwoordingen ontleed en geanalyseerd, met deze onledingen word een CDM gemaakt. Dit CDM word in dit document beschreven.

2 Functionele eisen

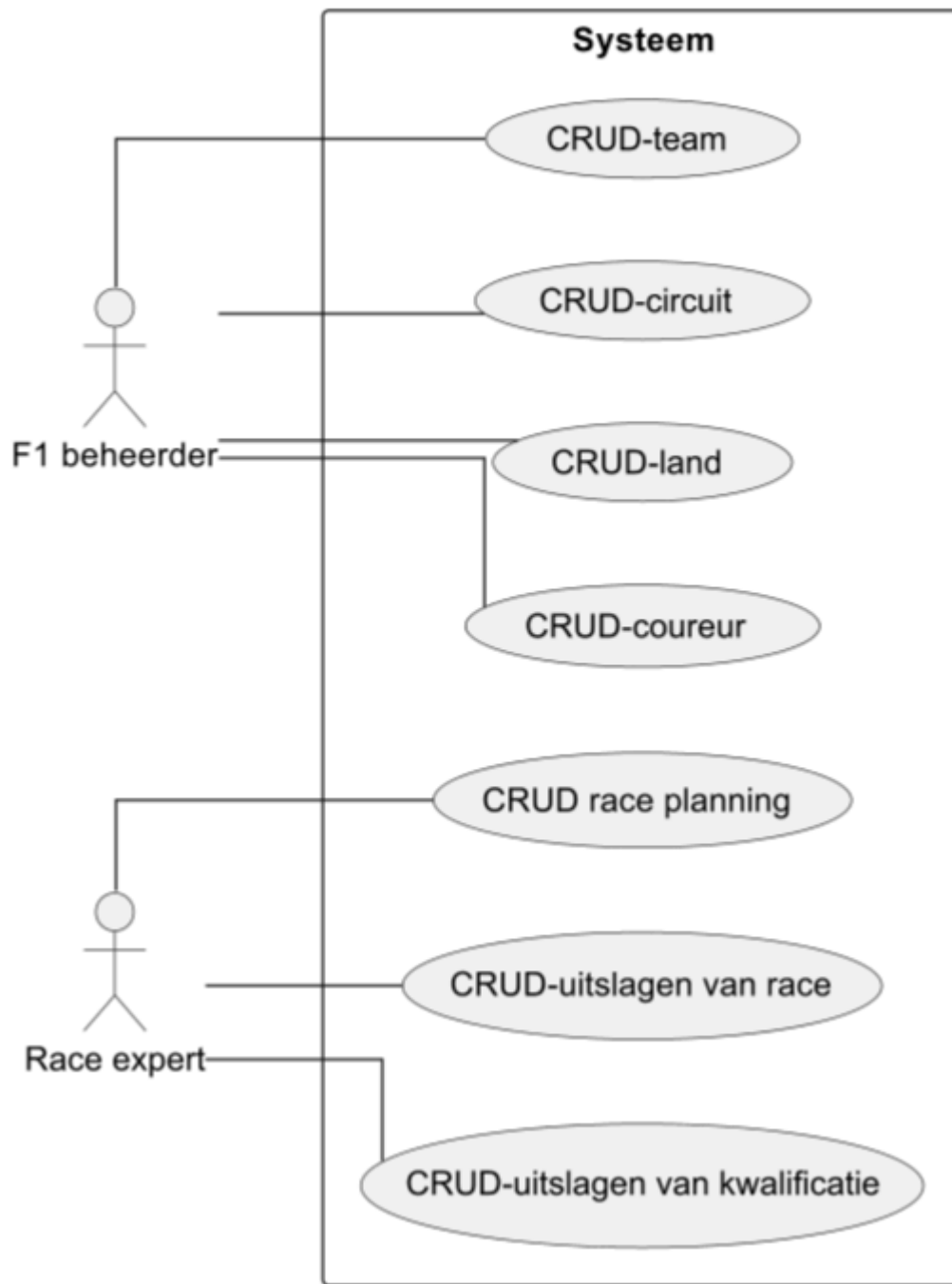
functioneel

- Het systeem moet data van planning van races kunnen aanbieden.
- Het systeem moet data van coureurs kunnen aanbieden.
- Het systeem moet data van uitslagen van races kunnen aanbieden.
- Het systeem moet data van pitstops (tijdens de race) kunnen aanbieden.
- Het systeem moet data van resultaten van kwalificatieraces kunnen aanbieden.
- Het systeem moet opslaan wie een verzoek voor een JSON aanvraagt.
- Het systeem moet data in JSON-formaat aanbieden.
- De database waar alle formule 1 data wordt in opgeslagen moet een relationele database zijn.
- De database waar alle JSON in worden opgeslagen moet een non-relatieve database zijn.
- Klanten mogen geen contact hebben met de database waarin alle formule 1 data direct is in opgeslagen.

3 Fully dressed

Use cases diagrams

In het plaatje hieronder staat een UML diagram van de use cases die in het formule 1 systeem zitten. Deze acties zullen door het systeem moeten kunnen worden uitgevoerd.



Hieronder heb je de fully dressed use cases. Omdat de fully dressed use cases allemaal CRUD zijn hebben wij er, maar twee uitgewerkt aangezien de rest in principe hetzelfde is met een paar kleine aanpassingen

Use case: CRUD-coureur

Primary actor: F1 beheerder

Stakeholders and Interests: Als F1 beheerder wil ik de coureurs kunnen toevoegen, bijwerken en verwijderen, zodat ik de database kan onderhouden.

Requirements:

- De F1 beheerder is ingelogd in de F1beheerder rol.

Brief description:

De F1 beheerder kan data toevoegen, bijwerken of verwijderen. Dit doet die door de bijbehorende procedure aan te roepen.

Preconditions:

De mongoDB staat online en kan verzoeken ontvangen.

Er is een land in de land tabel.

De API kan data versturen en ontvangen.

Postconditions:

De mongoDB is geupdate.

Main Success Scenario (Basic Flow):

Actor Action	System Responsibility
1.De F1 beheerder expert execute de addCoureur stored procedure.	
	2. Voegt de coureur data toe aan de relationele database.

Extensions (Alternative Flows): update coureur data.

Actor Action	System Responsibility
1b. De F1 beheerder expert execute de updateCoureur stored procedure.	
	2b. Update de coureur data in de relationele database.

Extensions (Alternative Flows): delete coureur data.

Actor Action	System Responsibility
1c. De F1 beheerder expert execute de deleteCoureur stored procedure.	
	2b. delete de coureur data in de relationele database.

Use case: CRUD race planning

Primary actor: Race expert

Stakeholders and Interests: Als Race expert wil ik de coureurs kunnen toevoegen, bijwerken en verwijderen, zodat ik de database kan onderhouden.	
Requirements: <ul style="list-style-type: none"> De Race expert is ingelogd in de RaceExpert rol. De bijbehorende Race expert is ingelogd in de RaceExpert rol. 	
Brief description: <p>De F1 beheerder kan data toevoegen, bijwerken of verwijderen. Dit doet die door de bijbehorende procedure aan te roepen.</p>	
Preconditions: <p>De mongoDB staat online en kan verzoeken ontvangen. Er zit data in de land tabel. Er zit data in de circuit tabel. De API kan data versturen en ontvangen.</p>	
Postconditions: <p>De mongoDB is geupdate.</p>	
Main Success Scenario (Basic Flow):	
Actor Action	System Responsibility
1.De Race expert execute de SP_addRacePlanning stored procedure.	
	2. De procedure voegt de race planning data toe aan de relationele database.
	3. Stopt de stored procedure addCoureur in de tabel executieSchema.
	4. Job runt de stored procedure en stuurt de gemaakte JSON door naar de API.
	5. API stuurt de JSON door naar de MongoDB.
	6. De mongoDB voegt de nieuwe race planning toe.

4 Entity relationship diagram (erd)

4.1 Feittypes

Bijlage D

	<p>ID: MATCH</p> <p>Predicaat: De grand prix op <circuitNaam> van <startdatum> heeft een totaal <eindatum>.</p>
8	<p>De coureur met id 2 zit in de grand prix Abu Dhabi van 2021-01-01 in het team Mercedes.</p> <p>De <i>coureur met id 1 zit in grand prix Abu Dhabi van 2021-01-01</i> in het <i>team Red Bull Racing Honda</i>.</p> <p>ET: deelnemendeCoureur ET: team</p> <p>ID: ET: race + ET: coureur ID: att: teamNaam</p> <p>ID: MATCH + MATCH</p> <p>RT: deelnemendeCoureurInRace tussen deelnemendeCoureur (dependent) en race.</p> <p>RT: deelnemendeCoureurBevatCoureur tussen deelnemendeCoureur (dependent) en coureur.</p> <p>RT: deelnemendeCoureurInTeam tussen deelnemendeCoureur en team</p> <p>Predicaat: De coureur met <coureurID> zit in de grand prix <circuitNaam> van <startdatum> in het <teamNaam>.</p>
8	<p>De coureur met id 2 in de grand prix Abu Dhabi van 2021-01-01 heeft volgnummer 2.</p> <p>De <i>coureur met id 1 in de grand prix Abu Dhabi van 2021-01-01</i> heeft <i>volgnummer 1</i>.</p> <p>ET: deelnemendeCoureur att: volgnummer</p> <p>ID: MATCH</p> <p>Predicaat: De coureur met <coureurID> in de grand prix <circuitNaam> van <startdatum> heeft <volgnummer>.</p>
10	<p>Het team Red Bull komt uit Oostenrijk.</p> <p>Het <i>team Force India</i> komt uit <i>India</i>.</p> <p>ET: team ET: land</p> <p>ID: MATCH ID: Match</p> <p>RT: teamThuisLand tussen team en land.</p> <p>Predicaat: Het <team> komt uit <land>.</p>
11	<p>De coureur met id 2 in de grand prix Abu Dhabi van 2021-01-01 heeft op de 57e lap een lap tijd van 98109 ms.</p> <p>De <i>coureur met id 2 in de grand prix Abu Dhabi van 2021-01-01 heeft op de 58e lap een lap tijd van 98109 ms</i>.</p> <p>ET: lap ; att: lapTijd</p> <p>ID: ET: deelnemendeCoureur + att: huidigeLap</p> <p>ID: ET: MATCH</p> <p>RT: deelnemendeCoureurZitOpLap tussen lap(dependent) en deelnemendeCoureur.</p> <p>Predicaat: De coureur met id <coureurID> zat in de grand prix <circuitNaam> van <startdatum> heeft op de <lap> lap een lap tijd van <lapTijd>.</p>
12	<p>De coureur met id 2 heeft in de grand prix Abu Dhabi van 2021-01-01 een tijd van +2.256s.</p> <p>De coureur met id 3 heeft in de grand prix Abu Dhabi van 2021-01-01 een tijd + 1 lap.</p> <p>De coureur met id 4 heeft in de grand prix Abu Dhabi van 2021-01-01 een tijd van DNF.</p> <p>De coureur met id 5 heeft in de grand prix Abu Dhabi van 2021-01-01 een tijd van DNS.</p> <p>De <i>coureur met id 1 heeft in de grand prix Abu Dhabi van 2021-01-01 een tijd van 1:30.17.345</i>.</p> <p>ET: deelnemendeCoureur Att: finishtijd</p> <p>ID: ET: race + ET: coureur</p> <p>ID: MATCH + MATCH</p> <p>RT: deelnemendeCoureurInRace tussen deelnemendeCoureur (dependent) en race.</p>

	ET: coureur Att: achternaam ID: MATCH Predicaat: De <coureurID> heeft als <achternaam>
20	De <i>coureur met id 1</i> heeft als <i>voornaam Max</i> . Coureur met id 2 heeft als voornaam Lewis. ET: coureur Att: voornaam ID: MATCH Predicaat: De coureur met id <coureurID> heeft als voornaam <voornaam>

4.1.2 Besluiten van feittypes

In **feittype 4** hebben wij ervoor gekozen om land een aparte ET te maken om datavervuiling tegen te gaan zoals dubbele data en foute data. Hierdoor is de data in de tabel meer consistent.

Kolom coureurID:

De kolom coureurID bevat de ID van de coureur. Wij hebben ervoor gekozen om voor elk coureur een ID op te nemen. Dit hebben wij gedaan omdat een coureur dezelfde voor en of achternaam als een andere coureur kan hebben. Door een ID toe te voegen aan de coureur kunnen wij de coureur uniek identificeren.

Team tabel

Team tabel hebben we los staan omdat een coureur voor andere teams kan uitkomen. Dit kan veranderen midden in het seizoen. Zo kunnen krijgen we minder duplicated records in de database ook minder foutieve data.

deelnemendeCoureur tijd

Als finishtijd 0 is, betekent het een DNS (did not start). Hiervoor kan je ook geen punten krijgen.

race startdatum

We hebben op tabel race startdatum en circuit gekozen als primary omdat het is voor gekomen dat ze op dezelfde circuit vaker in een jaar hebben geracet.

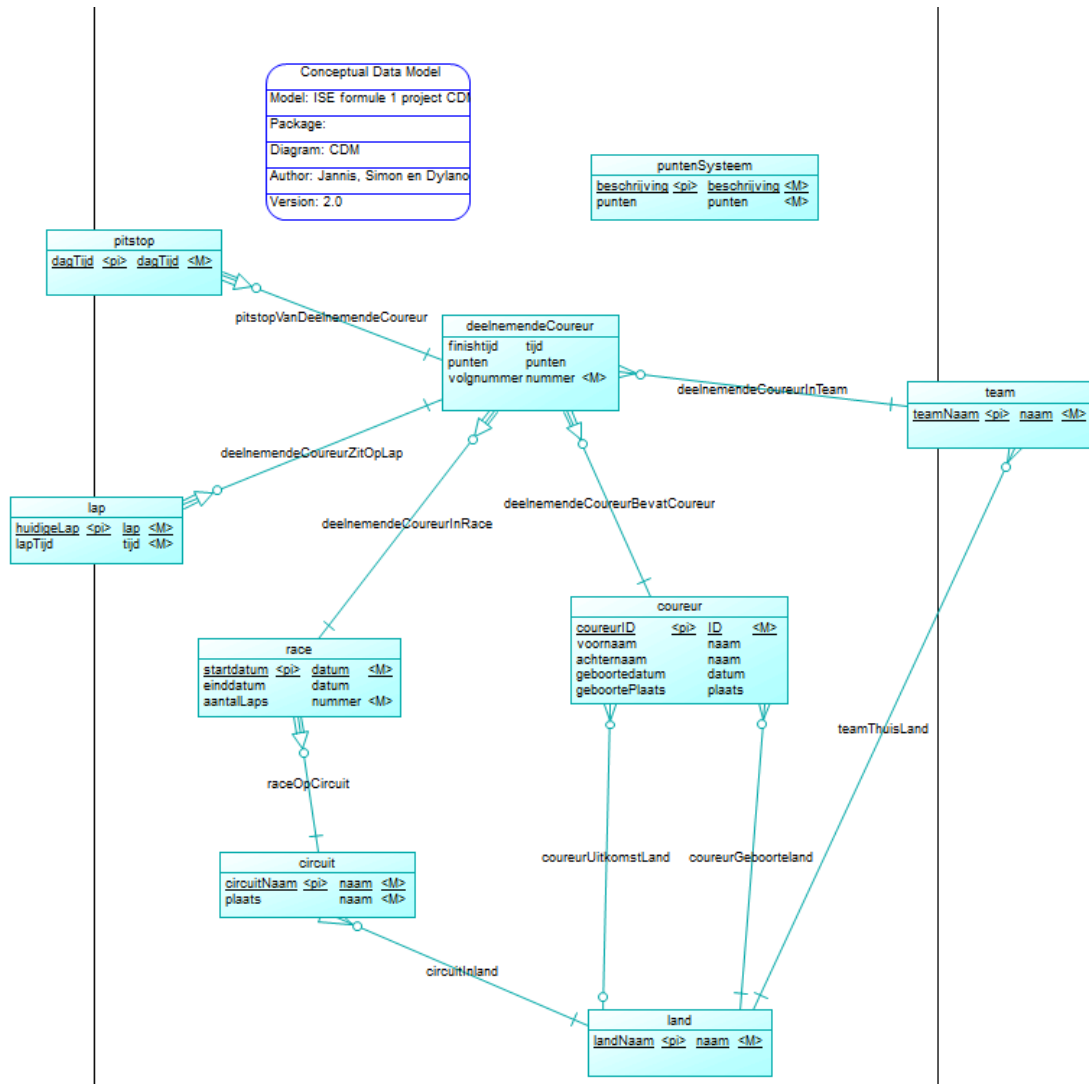
Punten DeelnemendeCoureurs

Punten worden nu per race opgeslagen zodat er een makkelijker gekeken kan worden hoeveel punten een coureur heeft per jaar.

4.2 Conceptueel Data Model(CDM)

Hieronder staat het CDM. In het CDM staat het ontwerp van de database na de ontleding van de verwoordingen. Hierin is te zien welke tabellen er in de database komen te staan met welke attributen en welke relaties er tussen de tabellen

zijn.



4.2.1 Beschrijving van de tabellen en attributen

Land:

De tabel land bevat alle landnamen van alle landen die hebben meegedaan aan de formule 1.

puntenSysteem:

De tabel puntenSysteem bestaat uit een beschrijving van waar iemand moet voldoen om de punten te krijgen en de bijbehorende punten bij die beschrijving. Ook heeft het een ID om makkelijk mee te kunnen rekenen.

circuit:

De tabel circuit bestaat uit de circuit namen van alle circuit van de formule 1 en in welke plaats het circuit ligt.

coureur:

De tabel coureur bestaat uit ID van de coureur en de voor en achternaam van de coureur.

race:

De tabel race bevat het startdatum wanneer de race is gestart, einddatum.

deelnemendeCoureur:

De tabel bevat Lap voor hoeveel rondes er zijn gereden. Finishtijd hoelang er gereden is.

Team:

De tabel bevat teamnaam.

Lap:

De tabel bevat de lap tijd van de coureur.

Pitstop:

Houd de tijd bij dat een pitstop heeft plaats gevonden.

4.3 Domains

Domain	Datatype	Beperkingen lengte	Keuze van datatype
datum	Date		We hoeven niet meer te hebben dan een datum
beschrijving	Variable characters	255	Hiervoor hebben we een limiet gekozen van 255 omdat er anders teveel opslag vrijgehouden wordt
ID	Integer		hebben we als integer omdat we de grootte niet goed kunnen inschatten dat we het kleiner kunnen maken.
lap	Integer		Hebben we als integer omdat we de grootte niet goed kunnen inschatten dat we het kleiner kunnen maken.
naam	Variable characters	60	We kiezen voor maximaal aantal characters van 60 omdat dat de standaard lengte is voor een naam in de database
punten	Integer		Hebben we als integer omdat we de grootte niet goed kunnen inschatten dat we het kleiner kunnen maken.
tijd	Long integer		Hier hebben we gekozen voor een long integer omdat de ronde tijd in millisecondes zijn dus er kunnen hele grote getallen uit komen
plaats	Variable characters	255	We kozen voor 255 characters omdat er plaatsnamen zijn die heel erg lang zijn
dagTijd	Time		Omdat we alleen de tijd moeten hebben wanneer iemand naar binnen komt voor een pitstop

4.4 Rules

4.4.1 Business rules

BR1: Een race heeft 20 coureurs. Dat mogen er niet meer of minder zijn.
BR2: Een startdatum van de race moet eerder zijn dan een einddatum.
BR3: Het volgnummer in de deelnemendeCoureur moet unique zijn per race.
BR4: Er mogen pas pitstops voor een race bestaand als er minstens één lap is geweest van die race.
BR5: De coureur moet minstens 16 zijn voordat hij deelneemt aan zijn eerste race.
BR6: Datums van een race kunnen niet overlappen met elkaar.

4.4.2 Constrains

BR1

ET coureur, Att: coureurID

ET race

Een race moet 20 coureurID's bevatten.

BR2

ET Race, ATT: startdatum en einddatum.

Voor elke race valt startdatum voor einddatum.

BR3

ET deelnemendeCoureur, Att: startdatum, circuit, volgnummer

BR4

ET

BR5

ET: coureur, Att: geboortedatum, coureurID

ET: deelnemendeCoureur, Att: coureurID, startdatum

BR6

ET: Race, Att: startdatum, einddatum

5. Executie rechten en rollen

In het systeem worden de volgende rollen onderscheiden: Klant, F1 Beheerder en F1 Expert. Deze zijn terug te vinden in het UseCaseDiagram. Om het systeem te beveiligen tegen misbruik kunnen de rollen niet direct Query's uitvoeren op de database. Dit moeten zij doen doormiddel van voorgedefinieerde stored procedures. Ook deze stored procedures worden beveiligd door middel van rechten. Hieronder is een rechten matrix te zien die aangeeft welke stored procedures welke rol mag uitvoeren. "X" betekent dat de rol de stored procedure mag uitvoeren, "Indirect" betekent dat de rol de stored procedure mag uitvoeren via een andere stored procedure maar niet direct.

	F1 Beheerder	F1 Expert
SP_addCountry	x	
SP_addTeam	x	
SP_addCoureur	x	
SP_addCircuit	x	
SP_addLap		x
SP_addRacePlanning		x
SP_deleteThisCoureur	x	
SP_deleteThisDeelnemendeCoureur	x	
SP_deleteLapForThisCoureur	x	
SP_deletePitStopForThisCoureur	x	
SP_updateDeelnemendeCoureur		x
SP_addDeelnemendeCoureur		x
SP_updateLap		x
SP_updateRacePlanning		x