Analyse Document

Jaron Poel

Inhoudsopgave

Inleiding: 2
Kwaliteitseisen en beperkingen 3
UC Diagram 4
Use cases 5
Conceptueel model 6
Flowchart 6

Inleiding:

Dit document gaat het hebben over de requirements en kwaliteitseisen die Circustrein moet bevatten en het ontwerp ervan.
Daarnaast laat ik van de must haves de use-case diagram, use cases, test cases, test matrix en conceptueel model zien. De use cases laten zien hoe een gebruiker met het systeem communiceert, de test cases zijn voorbeelden met test gegevens om use cases te testen en de test matrix laat zien of alle must-have requirement die zijn opgesteld minstens 1 use & test case hebben, de conceptueel model, de architectuur en het klassendiagram.

Kwaliteitseisen en beperkingen

B = Beperking
Een beperking houdt de grenzen in tot waar een functional requirement (FR) werkt. Bijvoorbeeld “Bij het menu hebben we slechts de opties…”

K = Kwaliteit
Een kwaliteit van een functional requirement geeft een positieve toevoeging op het functional requirement aan.

Must haves:

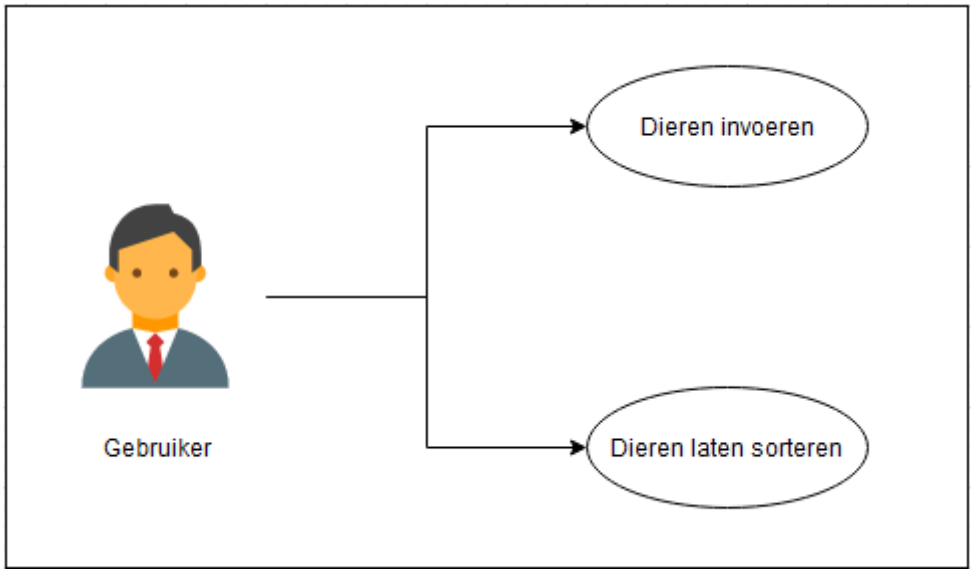
FR-01 Er moeten dieren kunnen worden ingevoerd.
- **B-01.1** Elk dier heeft een aantal punten gebaseerd op zijn/haar formaat
- **B-01.2** Elk dier heeft een voedsel type.

FR-02 Dieren moeten gesorteerd worden in treinwagons.
- **B-02.1** Elke treinwagon kan maximaal 10 punten aan formaat dieren vervoeren.
- **B-02.2** Een carnivoor mag niet met een dier van hetzelfde of een kleiner formaat zijn.
- **B-02.3** Het is niet toegestaan om elk dier in een aparte wagon te plaatsen.
- **K-02.1** Het eindresultaat na het sorteren wordt visueel aangetoond.

FR-03 Bij het invoeren en sorteren wordt rekening gehouden met het dier zijn voeding.
- **B-03.1** Voedsel types zijn: carnivoren & herbivoren.

FR-04 Bij het invoeren en sorteren wordt rekening gehouden met het dier zijn formaat.
- **B-04.1** Grote dier = 5 punten
- **B-04.2** Middel dier = 3 punten
- **B-04.3** Klein dier = 1 punt

UC Diagram

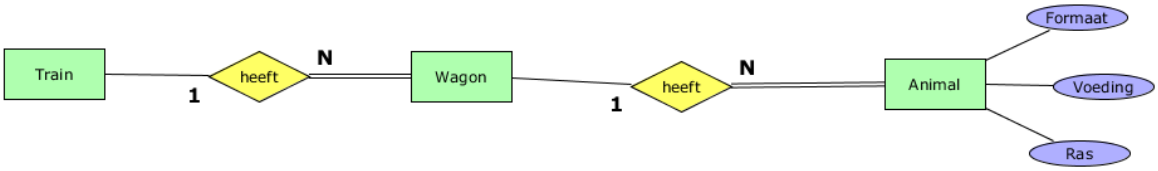


Use cases

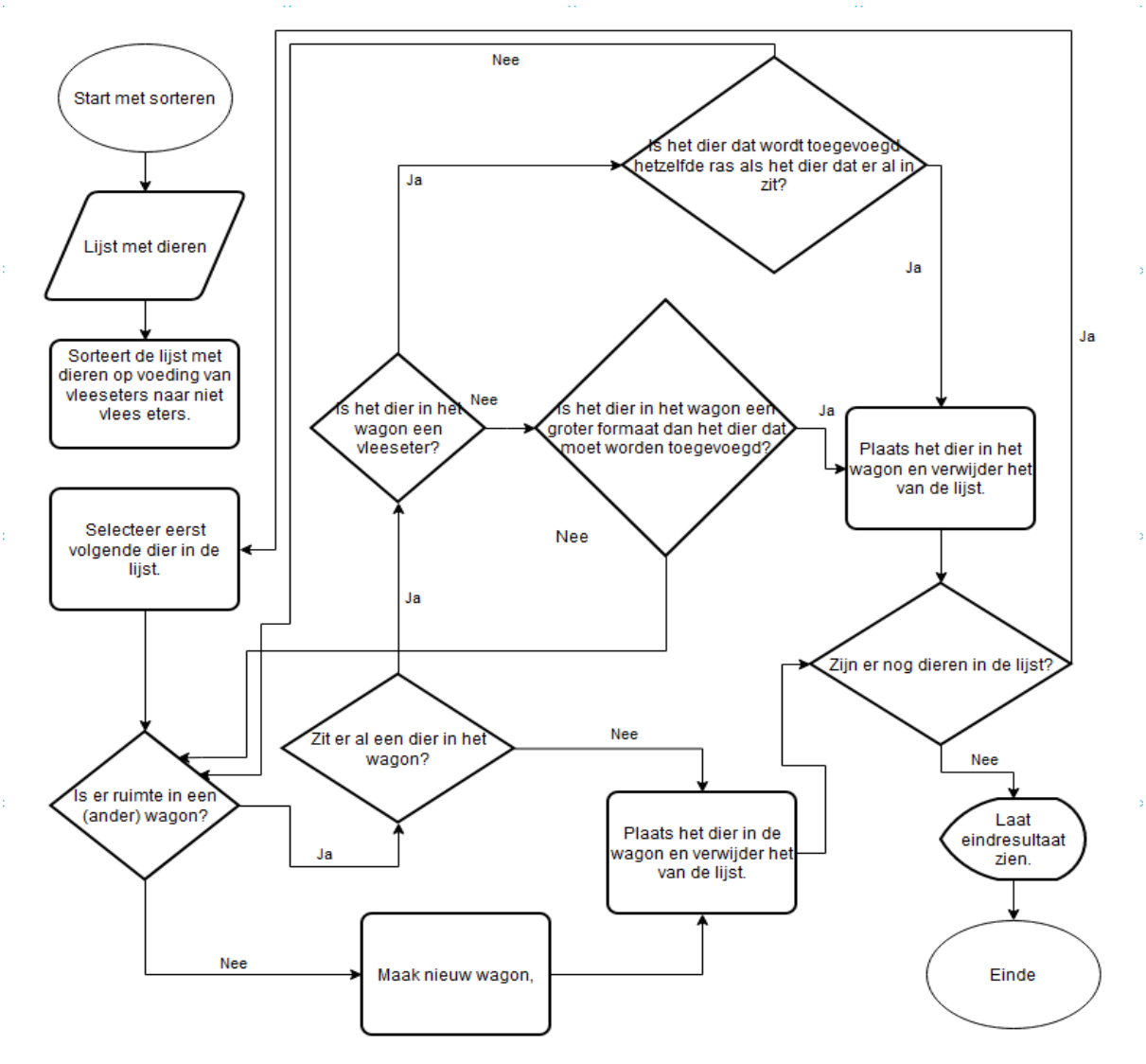
Tabel 1: Use case voor het invoeren van dieren. (UC01)	
Naam	Dieren invoeren.
Samenvatting	De actor kan dieren in het systeem invoeren.
Actors	Gebruiker
Aannamen	De actor heeft het programma opgestart.
Omschrijving	1. De actor geeft aan een dier te willen invoeren. 2. Het systeem toont de invoer pagina. 3. De actor vult de informatie in van het dier. 4. Het systeem controleert de gegevens en voegt het dier toe. [1]
Uitzonderingen	1. Bij een ongeldig formaat of voedsel type zal het systeem een foutmelding weergeven. Ga terug naar stap 3.
Resultaat	De actor heeft succesvol een dier toegevoegd.

Tabel 2: Use case voor het sorteren van dieren in wagons. (UC02)	
Naam	Dieren sorteren.
Samenvatting	De actor kan dieren die in het systeem zitten sorteren.
Actors	Gebruiker
Aannamen	De actor heeft de dieren die die wilt sorteren ingevoerd.
Omschrijving	1. De actor geeft aan de dieren te willen laten sorteren. 2. Het systeem voert de sortering algoritme uit en weergeeft de uitkomst van het algoritme aan de actor. [1]
Uitzonderingen	1. Als er iets fout gaat met het sorteren, zal het systeem een foutmelding weergeven met wat er fout is gegaan in het algoritme.
Resultaat	De actor heeft succesvol de dieren gesorteerd in wagons.

Conceptueel model



Flowchart



Klopt niet het ras gedeelte moet er uit.