

Functioneel ontwerp

- [Use cases](#)
- [Use Case 10 - Activeer Pas](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 8 - geluid aan poot toevoegen](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 4 - geluid toevoegen](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 11 - geluiden beluisteren](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 7 - Lijst van geluiden inzien](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 6 - Lijst van weetjes inzien](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 1 - Pas scannen](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 2 - Resultaten inzien](#)

- [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 11 - weetje aan poot toevoegen](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 5 - geluid toevoegen](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Use Case 3 - Weetje beluisteren](#)
 - [Preconditie](#)
 - [Postconditie](#)
 - [Success scenario](#)
 - [Alternatieve flow\(s\)](#)
- [Architectuur](#)
- [deployment](#)
 - [Beide versies](#)
 - [Development versie](#)
 - [Productie versie](#)
- [Mockups beschrijving](#)
 - [Ranger app](#)
 - [Administrator app](#)
- [domeinmodel](#)
- [Sequence diagrams](#)
 - [Nieuwe poot aanmelden](#)
 - [Online komen poot](#)
 - [Poot versturen logdata](#)
 - [Ranger bezoekt poot](#)
- [Requirements](#)
 - [Functionele Requirements](#)
 - [Niet-functionele Requirements](#)
- [deployment](#)
 - [Beide versies](#)
 - [Development versie](#)
 - [Productie versie](#)

Use cases

Use Case 10 - Activeer Pas

Primaire Actor: Ranger

Scope: Ranger applicatie in de diertuin

Een ranger wil zijn pas activeren door zijn bestaande ranger account er op te laden. Dit account is overgebleven van een vorig bezoek.

Preconditie

De ranger heeft een pas ontvangen bij de kassa.

Postconditie

Het account van de ranger is op de pas geladen.

Success scenario

Gebbruiker	Systeem
1. De ranger scant zijn pas bij de zuil.	2. Herkent de ranger pas en geeft de mogelijkheid om een ranger code in te voeren.
3. De ranger `voert zijn code in.	4. Het systeem herkent de code en laad de ranger gegevens op de pas.

Alternatieve flow(s)

4. Het systeem herkent de code niet en geeft de ranger de optie om opnieuw de code in te voeren.

Use Case 8 - geluid aan poot toevoegen

Primaire Actor: Administrator

Scope: binnen het administrator paneel

Een administrator wil een dierengeluid aan een poot toevoegen.

Preconditie

De administrator heeft toegang tot de unieke identifier van de poot en de poot waar de administrator het geluid van wil aanpassen is online.

Postconditie

Geluid op de poot is toegevoegd of gewijzigd.

Success scenario

Gebruiker	Systeem
1. Navigeert naar de "Poot aanpassen" pagina	
	2. Toont de "Poot aanpassen" pagina
3. Kiest ervoor om het dierengeluid aan te passen	
	4. Toont beschikbare dierengeluiden
5. Kiest een nieuw dierengeluid en klikt op "Aanpassen"	
	6. Slaat nieuw dierengeluid op bij de desbetreffende paal
	7. Update de poot met het nieuwe geluid

Alternatieve flow(s)

7. Poot valt onder het updaten weg.
1. Poot zal geüpdate worden zodra deze online komt.

Use Case 4 - geluid toevoegen

Primaire Actor: Administrator

Scope: Admin paneel

De administrator wil een nieuw dierengeluid toevoegen.

Preconditie

De administrator heeft het admin paneel open.

Postconditie

Het geluid is toegevoegd.

Success scenario

Gebruiker	Systeem
1. Navigeer naar de "nieuw geluid	

toevoegen"

2. Toont de pagina waar nieuwe geluiden geupload kunnen worden

3. Kiest een of meerdere geluiden om te uploaden.

4. Kiest om de bestanden te versturen

5. Upload de nieuwe bestanden en stuurt een confirmatie

Alternatieve flow(s)

5. Bestanden zijn niet valide, het systeem toont welke bestanden **niet** geupload zijn.

Use Case 11 - geluiden beluisteren

Primaire Actor: Administrator

Scope: Admin paneel

De administrator wil een geluid beluisteren

Preconditie

De administrator heeft het admin paneel open en de lijst van weetjes/geluidjes open.

Postconditie

Het geluid wordt afgespeeld.

Success scenario

Gebruiker

Systeem

1. Kiest ervoor om het geluidje te beluisteren

2. Geluid wordt gespeeld

Alternatieve flow(s)

Use Case 7 - Lijst van geluiden inzien

Primaire Actor: Administrator

Scope: Admin paneel

De administrator wil een lijst van geluiden zien.

Preconditie

De administrator heeft het admin paneel open.

Postconditie

Lijst van geluiden getoond

Success scenario

Gebruiker	Systeem
1. Navigeert naar de geluiden pagina.	2. Vraagt geluiden op 3. Toont beschikbare geluiden op de geluiden pagina

Alternatieve flow(s)

Use Case 6 - Lijst van weetjes inzien

Primaire Actor: Administrator

Scope: Admin paneel

De administrator wil een lijst van weetjes zien.

Preconditie

De administrator heeft het admin paneel open.

Postconditie

Lijst van weetjes getoond

Success scenario

Gebruiker	Systeem
1. Navigeert naar de weetjes pagina.	2. Vraagt weetjes op 3. Toont beschikbare weetjes op de weetjes pagina

Alternatieve flow(s)

Use Case 1 - Pas scannen

Primaire Actor: Ranger

Scope: Het scannen van de pas door een gebruiker

De ranger moet de pas op verschillende momenten scannen om een actie te voltooien.

Preconditie

De pas welke de ranger gebruikt is geactiveerd en gebruiksklaar.

Postconditie

De pas is gescand.

Success scenario

Gebruiker	Systeem
1. De ranger houdt de pas tegen de poot	2. De pas wordt gescant en de relevante info beschikbaar gemaakt

Alternatieve flow(s)

2. De pas is niet geactiveerd, het systeem laat een rood lampje branden en stopt de actieve usecase

Use Case 2 - Resultaten inzien

Primaire Actor: Ranger

Scope:

De ranger wil bij thuiskomst zijn resultaten kunnen bekijken. Hierbij kan de Ranger bijv. zien welke dieren de ranger bezocht heeft.

Preconditie

De ranger heeft zijn unieke rangercode bemachtigt (danwel via een print van een diploma of op een andere manier).

Postconditie

De ranger kan zijn unieke gegevens bekijken

Success scenario

Gebruiker

1. De ranger gaat thuis naar de resultaten website.

3. De ranger voert zijn unieke rangercode in.

Systeem

2. De resultatenwebsite wordt getoond

4. Het systeem haalt de unieke rangergegevens op en toont deze

Alternatieve flow(s)

3. De ranger voert een verkeerde code in, de website zal een foutmelding geven en opnieuw de mogelijkheid geven om de code in te voeren.

Use Case 11 - weetje aan poot toevoegen

Primaire Actor: Administrator

Scope: binnen het administrator paneel

Een administrator wil een weetje aan een poot toevoegen.

Preconditie

De administrator heeft toegang tot de unieke identifier van de poot en de poot waar de administrator het weetje van wil aanpassen is online.

Postconditie

Weetje op de poot is toegevoegd of gewijzigd.

Success scenario

Gebruiker

1. Navegeert naar de "Poot aanpassen" pagina

3. Kiest ervoor om het weetje aan te passen

5. Kiest een nieuw weetje en klikt op "Aanpassen"

Systeem

2. Toont de "Poot aanpassen" pagina

4. Toont beschikbare weetjes

6. Slaat nieuw weetje op bij de desbetreffende paal

7. Update de poot met het nieuwe weetje

Alternatieve flow(s)

7. Poot valt onder het updaten weg.

1. Poot zal geüpdate worden zodra deze online komt.

Use Case 5 - geluid toevoegen

Primaire Actor: Administrator

Scope: Admin paneel

De administrator wil een nieuw weetje toevoegen.

Preconditie

De administrator heeft het admin paneel open.

Postconditie

Het weetje is toegevoegd.

Success scenario

Gebruiker	Systeem
1. Navigeer naar de "nieuw weetje toevoegen"	
	2. Toont de pagina waar nieuwe weetjes geupload kunnen worden
3. Kiest een of meerdere weetjes om te uploaden.	
4. Kiest om de bestanden te versturen	
	5. Upload de nieuwe weetjes en stuurt een confirmatie

Alternatieve flow(s)

5. Bestanden zijn niet valide, het systeem toont welke weetjes **niet** geupload zijn.

Use Case 3 - Weetje beluisteren

Primaire Actor: Ranger

Scope: Hele systeem

Een ranger staat bij een poot en wil een weetje horen, de ranger scant hiervoor zijn pas en krijgt een weetje te horen.

Preconditie

De pas welke de ranger gebruikt is geactiveerd en gebruiksklaar.

Postconditie

De ranger heeft het weetje gehoord

Success scenario

Gebruiker	Systeem
1. De ranger scant de pas	2. Het systeem speelt een weetje
	3. Het systeem verwerkt de informatie

Alternatieve flow(s)

1. ..

Architectuur

Hier moet iets geschreven worden over de architectuur



deployment

Voor dit project zal er gewerkt worden met twee deployment strategieën, dit wordt gedaan omdat de deployment tijdens het testen minder secure hoeft te zijn dan tijdens echte productie. Het doel is om de productie build online te hebben bij de field-tests.

Beide versies

Beide versies draaien op een black-box server waar [Docker \(https://www.docker.com/\)](https://www.docker.com/) containers gedraaid kunnen worden (al dan niet met [docker-compose \(https://docs.docker.com/compose/\)](https://docs.docker.com/compose/)). Voor zowel de development deployments als de productie deployments wordt docker gebruikt.

Development versie

In de development deployment worden de softwareproducten gedraaid met hun built-in development servers. (bijv. npm start -> dev react app)

Dit staat ons toe om snel te schakelen tussen de verschillende producten aangezien er niks gebuild hoeft te worden.



Productie versie



Mockups beschrijving

Ranger app

De ranger app laat de trips zien die een ranger heeft afgelegd, ook geeft deze onderin (en via het menu) de optie om de gehoorde geluidjes nogmaals te horen.

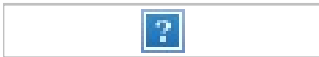


Administrator app

Als de administrator een poot wil aanpassen komt hij/zij eerst op de pagina waar alle poten weergegeven worden. Dat is de volgende pagina:



Als de administrator ervoor kiest om een poot aan te passen komt hij/zij op het volgende scherm (popup). Hier kan hij/zij onder andere de naam, het geluid, de weetjes en de positie van de poot aanpassen.



De volgende twee schermen tonen de "geluiden beheren" en "weetjes beheren" paginas. Op deze pagina's kan de administrator de weetjes/geluidjes zijn en deze afspelen.



Als de administrator op het + knopje rechtsboven drukt krijgt hij/zij het volgende scherm (popup) te zien om een geluid / weetje toe te voegen.



domeinmodel



Wanneer een kind voor het eerst een pas krijgt wordt de pas gekoppeld aan een nieuwe Ranger. Vanaf dat moment is het kind een ranger. Bij een ranger hoort een rangercode die het kind kan gebruiken om de volgende keer dat het kind bij burgers zoo komt verder te gaan als de zelfde ranger.

Een ranger gaat speuren naar poten. Op een saai plek in het park staan groepjes poten waar rangers naar op zoek kunnen gaan. Een groepje van poten die dicht bij elkaar staan en allemaal het zelfde weetje afspelen heet een speurpunt.

Een ranger bezoekt dus speurpunten. De ranger kan zijn bezoeken aan de speurpunten en het moment dat het speurpunt is bezocht terugvinden in de webapplicatie.

Wanneer een ranger hun pas scant bij de poot dan spreekt de poot een weetje uit. Vervolgens laat de poot een dierengeluid horen van een dichtbijzijnd dier.

Een speurtocht staat niet in het domeinmodel omdat er geen vaste volgorde is waarin een ranger de speurpunten moet bezoeken. Vandaar dat een 'speurtocht' concept niet bestaat.

Sequence diagrams

Nieuwe poot aanmelden

Wanneer een totaal nieuwe poot in het systeem komt, moet deze aangemeld worden binnen het systeem. Dit sequence diagram beschrijft de gebeurtenissen bij het aanmelden van een nieuwe poot. Deze poot heeft nog geen configuraties. De bedoeling is dat de poot een ID krijgt van de backend waarmee de poot identificeerbaar is.

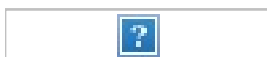


Online komen poot

Wanneer een poot uit heeft gestaan en weer online komt (bijvoorbeeld voor onderhoud of bij stroomuitval), zal deze een opstart sequence doorlopen. Wanneer een poot voor het eerst wordt opgestart zal de sequence diagram voor "*Nieuwe poot aanmelden*" doorlopen worden.

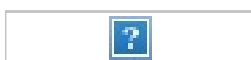
Na het aanmelden bij de gateway zal de gateway aan de backend vragen naar de configuratie van de poot. De backend zal de configuratie terug sturen naar de gateway. Wanneer de configuratie van de poot gelijk is gebleven zal de gateway geen actie ondernemen. Anders zal de gateway de audio bestanden downloaden en lokaal opslaan zodat deze later niet gedownload hoeft te worden.

De audio bestanden die nog niet op de poot voorkomen zullen dan verzonden worden naar de poot en op de gateway wordt bijgewerkt welke audio files op de poot staan. Gedurende het verzenden van de audio files zal de gateway ook updates naar de backend sturen om de voortgang te melden.



Poot versturen logdata

De poot verstuurt periodiek logdata naar de gateway. De gateway zal een timestamp toevoegen en een JSON object opbouwen. De timestamp wordt bij de gateway toegevoegd sinds de poot geen besef van tijd heeft. Dit object wordt doorgestuurd naar de backend. Onder logdata valt ook informatie over dat een ranger de poot heeft bezocht.



Ranger bezoekt poot

Link naar use case: [Link naar use case \(https://github.com/HANICA-MinorMulti/nj2017-iot-dwa-BurgersZoo1/blob/docs/documentatie/use%20cases/pas%20scannen/pas%20scannen.md\)](https://github.com/HANICA-MinorMulti/nj2017-iot-dwa-BurgersZoo1/blob/docs/documentatie/use%20cases/pas%20scannen/pas%20scannen.md)

Wanneer een ranger een poot bezoekt scant de ranger de NFC kaart. De poot verstuurt het id dat op de pas staat door naar de gateway. De gateway voegt een timestamp toe aan het het scannen van de kaart en stuurt dit door naar de backend.



Requirements

Functionele Requirements

- Een ranger kan een rangerpas verkrijgen aan de kassa.
- Een ranger kan deelnemen aan een speurtocht met een rangerpas.
- Door de rangerpas te scannen aan een poot, maakt de ranger vorderingen in de speurtocht.
- De ranger kan luisteren naar een weetje en een dierengeluid door zijn rangerpas te scannen.
- De ranger kan behaalde resultaten inzien door de website te bezoeken met zijn rangercode.
- De ranger krijgt een rangercode voor thuisgebruik na afloop van zijn bezoek.
- De administrator kan poten configureren door de webinterface te gebruiken.
- De administrator kan nieuwe weetjes en geluiden toevoegen aan het systeem.
- De administrator kan weetjes en geluiden aan een poot toevoegen door de webinterface te gebruiken.
- De administrator kan een lijst van alle weetjes en geluiden zien in de webinterface.
- De administrator kan luisteren naar bestaande weetjes en geluiden in het systeem door de webinterface te gebruiken.

Niet-functionele Requirements

- Het systeem moet meerdere requests tegelijk aankunnen.
- De poot moet in een vochtige omgeving kunnen staan.
- Het systeem moet kunnen communiceren in een omgeving zonder wifi en gps.

deployment

Voor dit project zal er gewerkt worden met twee deployment strategieën, dit wordt gedaan omdat de deployment tijdens het testen minder secure hoeft te zijn dan tijdens echte productie. Het doel is om de productie build online te hebben bij de field-tests.

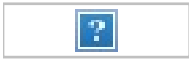
Beide versies

Beide versies draaien op een black-box server waar [Docker \(https://www.docker.com/\)](https://www.docker.com/) containers gedraaid kunnen worden (al dan niet met [docker-compose \(https://docs.docker.com/compose/\)](https://docs.docker.com/compose/)). Voor zowel de development deployments als de productie deployments wordt docker gebruikt.

Development versie

In de development deployment worden de softwareproducten gedraaid met hun built-in development servers. (bijv. npm start -> dev react app)

Dit staat ons toe om snel te schakelen tussen de verschillende producten aangezien er niks gebuild hoeft te worden.



Productie versie

