



Rapport Conceptfase

Verslag

Yarin van Raam	3APP01
Jarne Baelus	3APP02
Jannes Van den Broeck	3APP01
Daan Michielsen	3AI
Thibaud Wagemans	3CCS02
Wim Adriaensen	3CCS01

Project 4.0

Academiejaar 2023-2024

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

VOORWOORD

In dit project komen alle technologietakken samen om een project te realiseren voor een echte klant. Onze klant is VITO, VITO is actief in België en een heleboel andere landen in de wereld om streven naar duurzame ontwikkeling.

Dit document is een blauwdruk voor onze groep om het project in goede banen te leiden door een duidelijke afbakening te maken voor elk onderdeel waaruit onze oplossing bestaat.

Ons team

- Application Development (APP)
 - o Jannes Van den Broeck
 - o Jarne Baelus
 - o Yarin van Raam
- Cloud & Cyber Security (CCS)
 - o Thibaud Wagemans
 - o Wim Adriaensen
- Artificial Intelligence (AI)
 - o Daan Michiels

Project beschrijving

Er is een groot probleem onder de boeren in België waarbij er op hun velden een plant staat genaamd "Knolcyperus".

Knolcyperus, wetenschappelijk bekend als *Cyperus esculentus*, is oorspronkelijk afkomstig uit Noord-Afrika, het Midden-Oosten en delen van Europa. Het heeft zich verspreid naar verschillende delen van de wereld, waaronder België, om verschillende redenen.

Knolcyperus wordt vaak aangetroffen in landbouwgebieden en komt via verschillende routes België binnen. Het kan zich verspreiden via zaden, maar ook via knollen (ondergrondse delen van de plant) die gemakkelijk in de grond kunnen overleven en nieuwe planten kunnen produceren.

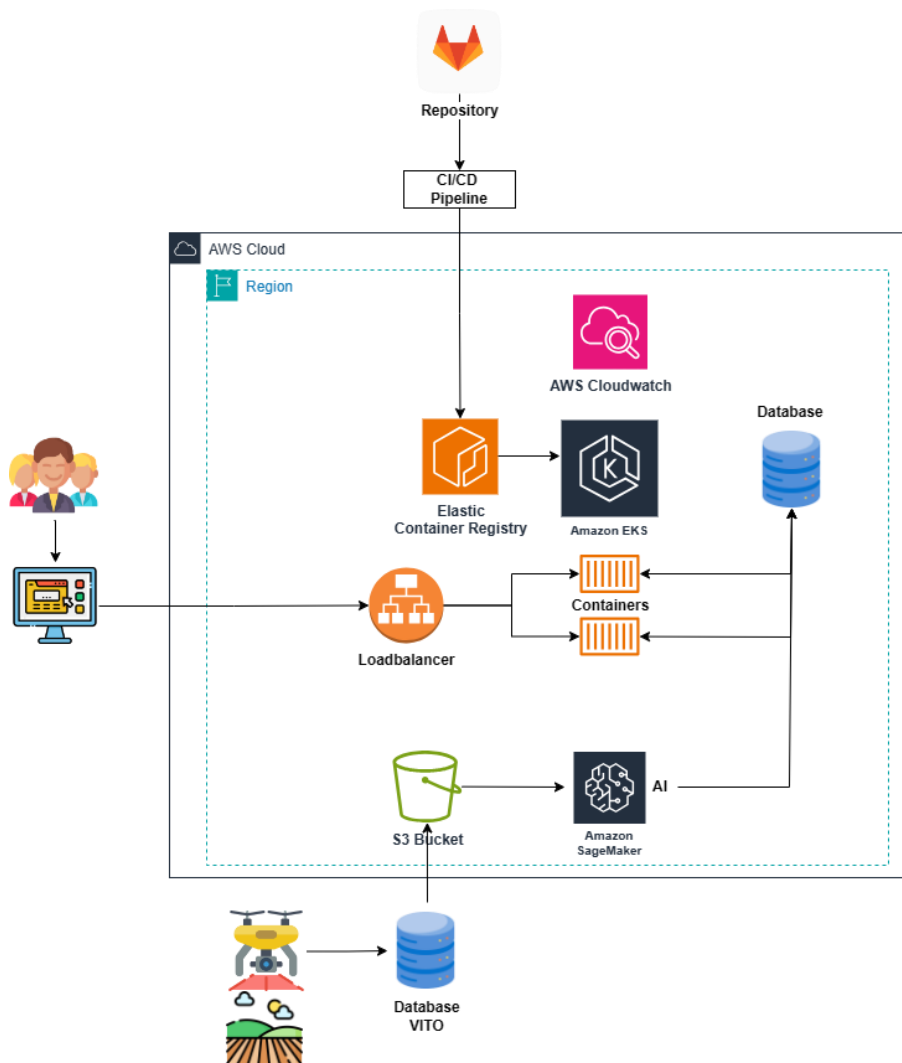
Wij willen het monitoren van knolcyperus op velden bevorderen door het detecteren te automatiseren met onze applicatie.

INHOUDSOPGAVE

VOORWOORD	3
ONS TEAM	3
PROJECT BESCHRIJVING	3
INHOUDSOPGAVE	5
1 ARCHITECTURALE VOORSTELLING	8
2 GEKOZEN TECHNOLOGIEËN	9
2.1 WDM HOSTING PLATFORM	9
2.2 WDM WEBSERVER	10
2.3 WDM DATABASE	10
2.4 WDM AI MODEL	11
2.5 WDM WEB APPLICATIE FRONTEND	11
2.6 WDM MOBIELE APPLICATIE FRONTEND	12
2.7 WDM WEB APPLICATIE BACKEND	12
2.8 DIAGRAM DEVOPS RESULTAAT	13
3 USECASE DIAGRAM	15
3.1 USECASE DIAGRAM WEB APPLICATIE	15
3.1.1 <i>Login</i>	16
3.1.1.1 Login	16
3.1.2 <i>Wachtwoord resetten</i>	17
3.1.2.1 Wachtwoord resetten	17
3.1.3 <i>Registreren</i>	18
3.1.3.1 Medewerker registreren	18
3.1.4 <i>Overzicht van velden</i>	19
3.1.4.1 Overzicht velden	19
3.1.5 <i>Overzicht veld</i>	20
3.1.5.1 Overzicht veld	20
3.1.6 <i>Download SHAPE-file</i>	21
3.1.6.1 Overzicht veld	21
3.1.7 <i>Medewerkers beheren</i>	22
3.1.7.1 Medewerkers beheren	22
3.1.7.2 Medewerker aanmaken	23
3.1.8 <i>Gewassen beheren</i>	23
3.1.8.1 Gewassen beheren	24
3.1.8.2 Gewas aanmaken	24
3.1.9 <i>Upload drone beelden</i>	25
3.1.9.1 Drone beelden uploaden	25
3.1.10 <i>Gebruikers beheren</i>	25
3.1.10.1 Gebruikers beheren	26
3.1.10.2 Gebruiker toevoegen	26
3.1.11 <i>Velden beheren</i>	27
3.1.11.1 Velden beheren	27
3.1.11.2 Veld toevoegen	28
3.2 USECASE DIAGRAM MOBILE APPLICATIE	28
3.2.1 <i>Login</i>	29
3.2.1.1 Login	29
3.2.2 <i>Wachtwoord resetten</i>	30
3.2.2.1 Wachtwoord resetten	30
3.2.3 <i>Overzicht velden</i>	31
3.2.3.1 Overzicht velden	31

3.2.4	<i>Heatmap van een veld</i>	32
3.2.4.1	Overzicht veld	32
3.2.5	<i>Bekijk knolcyperus in AR</i>	33
3.2.5.1	AR	33
3.2.6	<i>Fouten corrigeren</i>	34
3.2.6.1	AR	34
4	ERD	35
5	AI MODEL WERKING	37
5.1	TRAINEN.....	37
5.2	TESTEN.....	37
5.3	VISUELE WEERGAVE VAN DE WERKING.....	37
6	SHAPEFILE TRACTOR	38
7	CONCLUSIE	39

1 ARCHITECTURALE VOORSTELLING



Al onze code voor het project wordt bewaard op onze Gitlab Repository. Bij een nieuwe merge in de main-branch van onze repository zal er een pipeline beginnen lopen die automatisch de nieuwe code build, verpakt in een Docker image en pusht naar de Container Registry op AWS (Hierin worden alle container images bewaard).

EKS (Kubernetes in AWS) zal container images uit ECR halen om hiervan containers te bouwen in een cluster. Kubernetes clusters zorgen ervoor dat onze resources dynamisch worden gealloceerd om aan de huidige trafiek te voldoen. De containers in de clusters halen hun informatie van onze database (Amazon RDS).

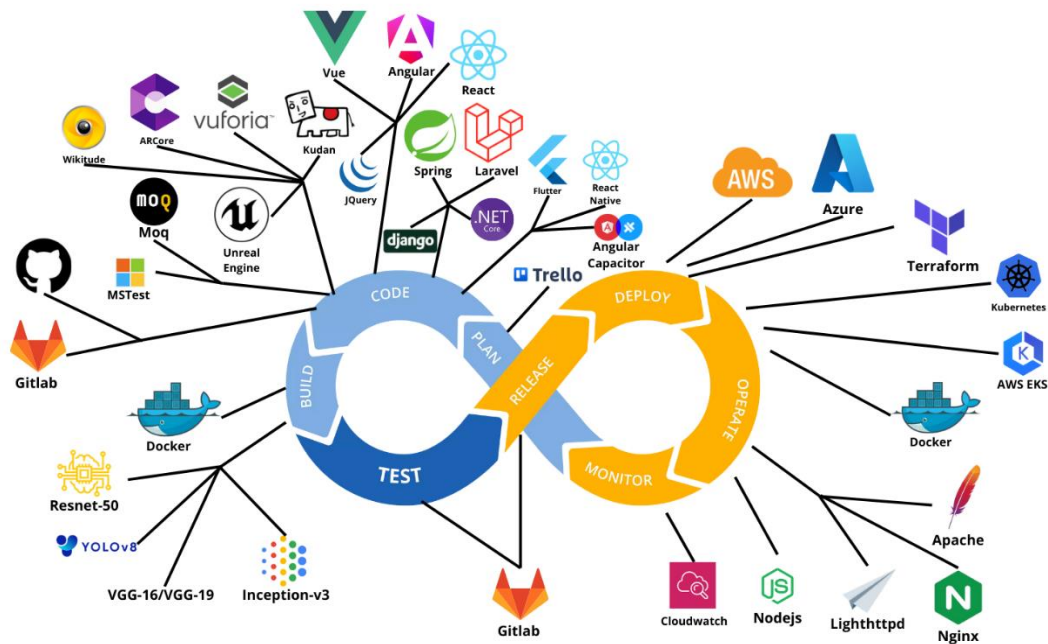
Als een gebruiker nieuwe beelden heeft laten maken zal de applicatie de beelden van de database van VITO halen om deze op te slaan op onze storage op Amazon S3. De API van het AI-model zal dan een call maken naar de S3 buckets om zo de beelden te gebruiken om knolcyperus te detecteren.

De resultaten van het model worden vervolgens opgeslagen in onze RDS-database. Onze applicatie wordt dan aangepast aan de nieuwe resultaten van het model zodat de boer de nieuwe data kan raadplegen.

2 GEKOZEN TECHNOLOGIEËN

In het onderstaand diagram laten we alle overwogen technologieën zien per stap in de DevOps Lifecycle.

Onder het diagram lijsten we alle *Weighted Decision Matrix*'en op om zo van alle technologieën de beste eruit te halen. Onderaan dit hoofdstuk vindt u de geüpdatete versie van de DevOps Lifecycle met de gekozen technologieën.



2.1 WDM Hosting Platform

Hosting Platform				
	In-house Knowledge	Ease of Use	Accessibility (for students)	Final Weight
Weight	2,00	3,00	3,00	
AWS	4	4	5	35
Azure	2	4	4	28

In deze WDM hebben we 2 mogelijke Cloud platformen vergeleken waarop wij ons project kunnen hosten. De 2 platformen waaruit we kiezen zijn AWS en Azure. Deze platformen hebben we vergeleken op vlak van de kennis die we al bezitten over deze Cloud platformen, de gebruiksvriendelijkheid en de toegankelijkheid voor studenten.

De reden dat de toegankelijkheid voor studenten hoger is bij AWS is omdat we voor AWS gemakkelijk een nieuw Lab kunnen aanvragen met opnieuw een budget van \$100. Bij Azure krijgt een student geen Lab omgeving maar wel \$100 aan resources, wanneer dit op is betaal je verder uit eigen zak.

2.2 WDM Webserver

WDM webserver					
	Performance	User Friendly	Compatibility	Knowledge	Final Weight
Weight	4	3	5	2	
Apache	4	4	5	5	63
Nginx	5	4	4	3	58
Lighttpd	3	3	3	2	40
NodeJS	4	4	4	1	50

Als keuzen voor de webserver hebben we Apache, Nginx, Lighttpd en NodeJS vergeleken. Over het algemeen scoren de technologieën ongeveer gelijk over de criteria, buiten Lighttpd. Het grootste verschil zit hem in de '*Knowledge*', hiermee bedoelen we de '*in-house knowledge*', hoeveel wij zelf al over de technologie kennen. Aangezien we tijdens ons studietraject vooral gebruik gemaakt hebben van Apache komt deze er dus als beste uit.

2.3 WDM Database

Database				
	Compatible	User-Friendly	Performance	Final Weight
Weight	5,00	3,00	1,00	
MySQL	5	4	4	41
MongoDB	5	3	5	39
MariaDB	4	3	4	33
PostgreSQL	4	4	4	36

Als Database Management Systeem hebben we de vergelijking gemaakt tussen MySql, MongoDB, MariaDB en Postgresql. De belangrijkste criteria waarmee we ze vergelijken zijn compatibiliteit met andere technologieën en bestandstypen, gebruiksvriendelijkheid en performantie.

Deze technologieën scoren ongeveer gelijk over de criteria, maar MySQL komt er net bovenuit. We hebben eerder gekozen voor AWS als Cloudplatform waarin RDS (Relational Database Service) de service is voor databases. Hierin kunnen we MySQL kiezen als databasesysteem.

2.4 WDM AI model

AI model						
	Accuracy	Ease of use	Familiarity	Speed		Final Weight
Weight	5.00	4.00	3.00	2.00		
Resnet-50	4	5	4	5		62
VGG-19	5	4	3	4		58
Inception-v3	4	4	3	4		53
VGG-16	5	3	4	3		55

Resnet-50 is geschikt voor onze toepassing.

Na wat onderzoek is het resnet-50 model het meest geschikt als de verwerkingssnelheid belangrijk is, zonder nauwkeurigheid op te geven. Voor ons zou ik zeggen dat de nauwkeurigheid wellicht belangrijker is dan de snelheid, omdat de resultaten sowieso enige tijd in beslag zullen nemen, en weten of een veld knolcyperus bevat is belangrijker dan het zo snel mogelijk verkrijgen van de resultaten.

2.5 WDM Web Applicatie Frontend

Frontend Keuze						
	Performance	Support	Ease to use	Knowledge	Features	Final Weight
Weight	1,80	1,20	1,80	1,90	1,70	
Angular	4	5	4	4	5	36,5
Vue	3	3	4	1	4	23
Jquery	3	3	3	2	4	29,2
React	5	5	4	3	4	32

Voor de frontend hebben we een keuze gemaakt uit 4 redelijk bekende frameworks, Angular, Vue, JQuery, en React. Deze frameworks zijn beoordeeld op hun performance, de community support, de ease to use, hoe ervaren we er mee zijn binnen onze groep, en de features die het geeft.

Over het algemeen steken er 2 frameworks bovenuit, dit zijn Angular en React. De meest beslissende factor van het resultaat was de kennis binnen het team, en daar heeft Angular de voorsprong. De reden dat het team meer kennis heeft van Angular is omdat wij een heel semester aan een project gewerkt hebben. React zijn we pas later mee begonnen en hebben we dus minder ervaring mee.

2.6 WDM Mobiele Applicatie Frontend

Frontend app Keuze						
	Performance	Support	Ease to use	Knowledge	Features	Final Weight
Weight	1,80	1,20	1,80	1,90	1,70	
Angular Capacitor	4	5	4	5	5	38,4
React Native	4	4	4	5	4	35,5
Flutter	3	2	3	3	3	33,3

Voor de frontend keuze voor de mobiele applicatie hebben we rekening gehouden met dezelfde factoren als de webapplicatie. We hebben gekeken naar enkele frameworks die we al in de les hebben gezien.

We hebben React Native en Flutter al gezien in de les en Angular Capacitor nog niet maar dit is hetzelfde als een gewone Angular app maken en zou dus niet zo veel verschil zijn.

Dit is dan ook de voornaamste reden waarom we Angular Capacitor erboven uitkomt. We hebben al veel kennis van Angular zoals eerder vermeld en we kunnen hierbij dezelfde codebase gebruiken voor onze mobiele en webapplicatie. Dit maakt het development proces voor beide veel makkelijker.

2.7 WDM Web Applicatie Backend

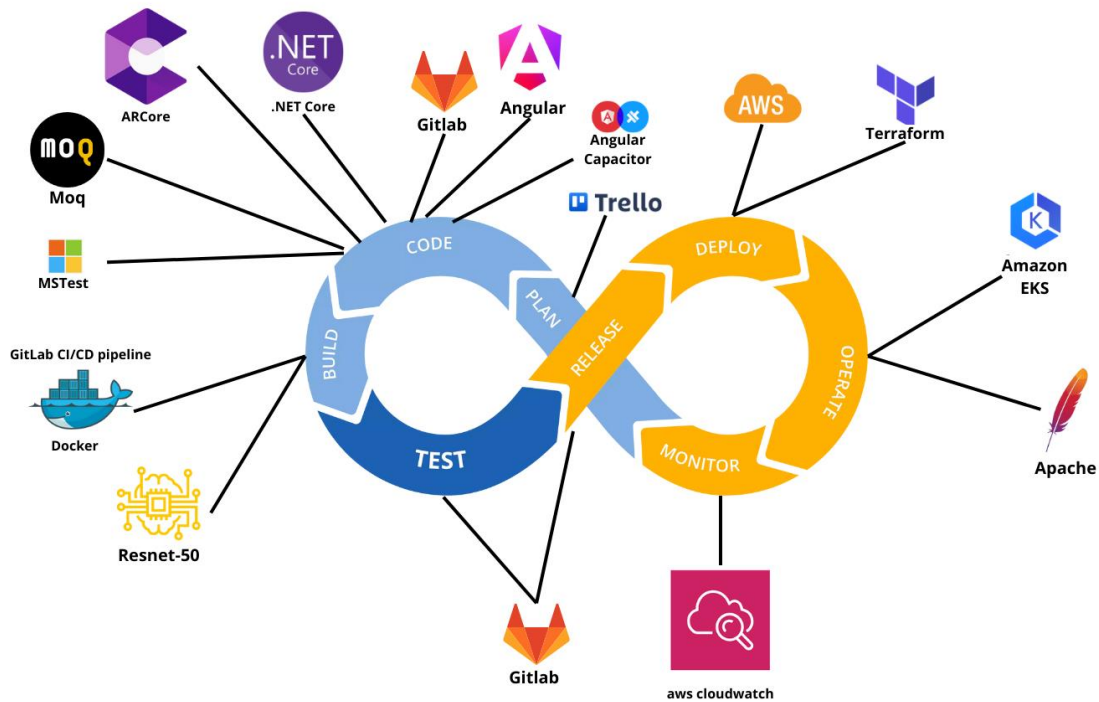
Backend Keuze						
	Performance	Support	Ease to use	Knowledge	Features	Final Weight
Weight	1,80	1,20	1,80	1,90	1,70	
Laravel	3	4	4	3	3	28,2
Spring	4	4	3	4	4	24,2
.NET	5	3	3	5	4	39,8
Django	4	4	3	2	3	30,5

Voor de keuze van het Backend framework, hebben we op dezelfde criteria beoordeeld als de frontend keuze. Performance, community support, hoe makkelijk het is, de kennis ervan binnen het team, en de features. De frameworks die we beoordeeld hebben zijn Laravel, Spring, .NET, en Django.

Voor de Backend was de keuze een stuk duidelijker dan de frontend. .NET schiet er bovenuit. Over het algemeen is .NET een heel geoptimaliseerd framework. Ook is er een grote actieve community, en over het algemeen is dit zeer makkelijk te gebruiken. Ook is ons team er zeer bekend mee, aangezien we deze backend het meest gebruikt hebben voor andere projecten en vakken. .NET heeft ook een zeer uitgebreid aantal features, waardoor we zo goed als alles kunnen doen wat we moeten doen.

2.8 Diagram DevOps resultaat

Dit deel van het document biedt een gedetailleerd overzicht van het project, met de nadruk op de DevOps cycle. Het project maakt gebruik van geavanceerde technologieën om een veilige en schaalbare cloudomgeving te creëren, waarbij de gehele levenscyclus van ontwikkeling tot implementatie wordt afgedekt.



Plan

Bij de planningsfase van de DevOps cycle wordt Trello gebruikt als samenwerkingsplatform. Trello biedt een gestructureerde en intuïtieve omgeving voor het beheren van taken en het plannen van projecten.

Code

In de codefase wordt een gevarieerde set van technologieën gebruikt om robuuste en efficiënte software te ontwikkelen. De belangrijkste technologieën zijn:

Angular: Voor de ontwikkeling van de gebruikersinterface.

GitLab: Als versiebeheersysteem en samenwerkingsplatform.

FastAPI: Een modern, snel webframework voor het bouwen van API's.

ARCore: Voor het integreren van augmented reality in de applicatie.

Moq en MSTest: Frameworks voor het uitvoeren van unit tests.

Build

De buildfase maakt gebruik van de GitLab CI/CD pipeline om Docker-images te bouwen. Daarnaast wordt ResNet-50 gebruikt voor het ontwikkelen van het AI-model dat in de applicatie wordt geïntegreerd.

Test

GitLab wordt opnieuw ingezet in de testfase om geautomatiseerde tests uit te voeren in de CI/CD-pipeline. Dit stelt het team in staat om vroegtijdig en continu kwaliteitscontroles uit te voeren. De geteste code wordt vervolgens ingezet in een specifieke testomgeving.

Release

GitLab wordt gebruikt voor de releasefase, waarin de goedgekeurde code in productie wordt ingezet. Dit proces maakt het mogelijk om op een gecontroleerde manier wijzigingen door te voeren in de live-omgeving.

Deploy

Terraform, geïntegreerd in de GitLab CI/CD-pipeline, wordt gebruikt voor het geautomatiseerd implementeren van infrastructuur op AWS. Dit waarborgt een consistente en beheersbare implementatie.

Operate

In de operationele fase draait de applicatie op AWS EKS (Kubernetes) met behulp van Apache voor het hosten van de webapplicatie. Dit zorgt voor schaalbaarheid en betrouwbaarheid in de productieomgeving.

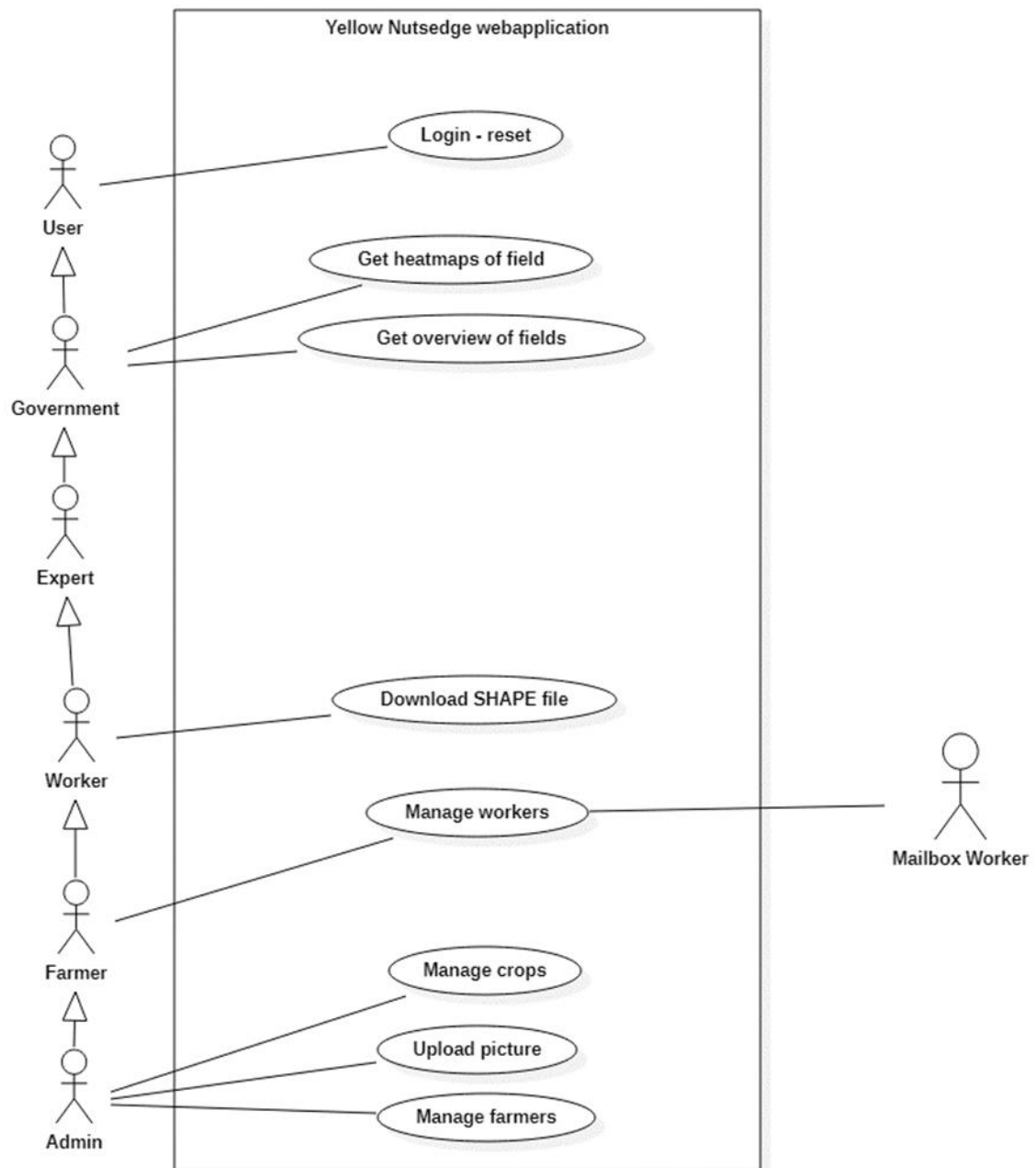
Monitor

AWS CloudWatch wordt gebruikt om de gehele infrastructuur en applicatie te monitoren. Dit omvat het bijhouden van prestaties, logboeken en waarschuwingen, waardoor het team proactief kan reageren op eventuele problemen.

3 USECASE DIAGRAM

In dit deel van ons rapport gaan we kijken naar de Usecase diagrammen voor zowel de webapplicatie als de mobiele applicatie.

3.1 Usecase Diagram Web Applicatie



3.1.1 Login

Functionaliteit: als gebruiker, kan ik me inloggen

Normale werking: het systeem laat het login scherm zien. De gebruiker vult zijn inloggegevens in en klikt op login.

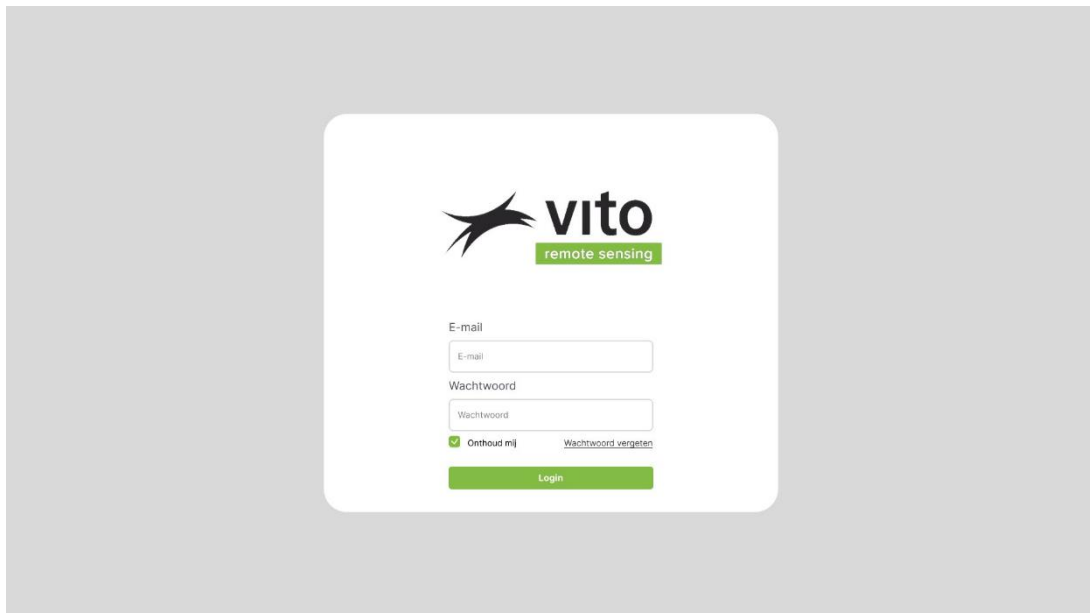
Alternatieven:

Foute inloggegevens: het systeem laat zien dat de inloggegevens fout zijn.

Wachtwoord resetten: het systeem gaat naar de pagina "Wachtwoord resetten".

Schermen:

3.1.1.1 Login



Actie	Uitleg
Login	Link naar pagina "3.1.4.1 Overzicht velden".

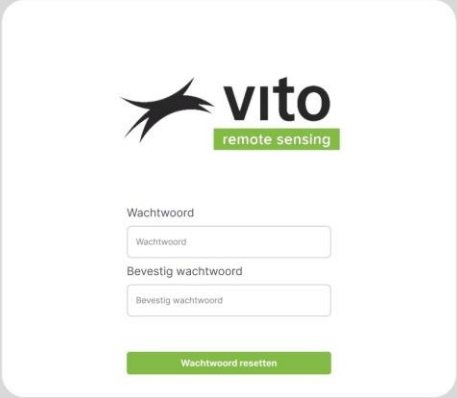
3.1.2 Wachtwoord resetten

Functionaliteit: als gebruiker, kan ik mijn wachtwoord resetten.

Normale werking: het systeem laat de pagina om het wachtwoord te resetten zien. De actor geeft het nieuwe wachtwoord in en klikt op wachtwoord resetten.

Schermen:

3.1.2.1 Wachtwoord resetten



The screenshot shows a web interface for resetting a password. At the top is the 'vito remote sensing' logo. Below it, there are two text input fields. The first is labeled 'Wachtwoord' and the second is labeled 'Bevestig wachtwoord'. At the bottom of the form is a green button with the text 'Wachtwoord resetten'.

Actie	Uitleg
Wachtwoord resetten	Link naar pagina "3.1.1.1 Login".

3.1.3 Registreren

Functionaliteit: als uitgenodigde werker, kan ik me registreren.

Normale werking: het systeem laat een formulier zien. De actor vult zijn gegevens en klikt op registreer.

Schermen:

3.1.3.1 Medewerker registreren

Actie	Uitleg
Registreren	Link naar pagina "3.1.1.1 Login".

3.1.4 Overzicht van velden

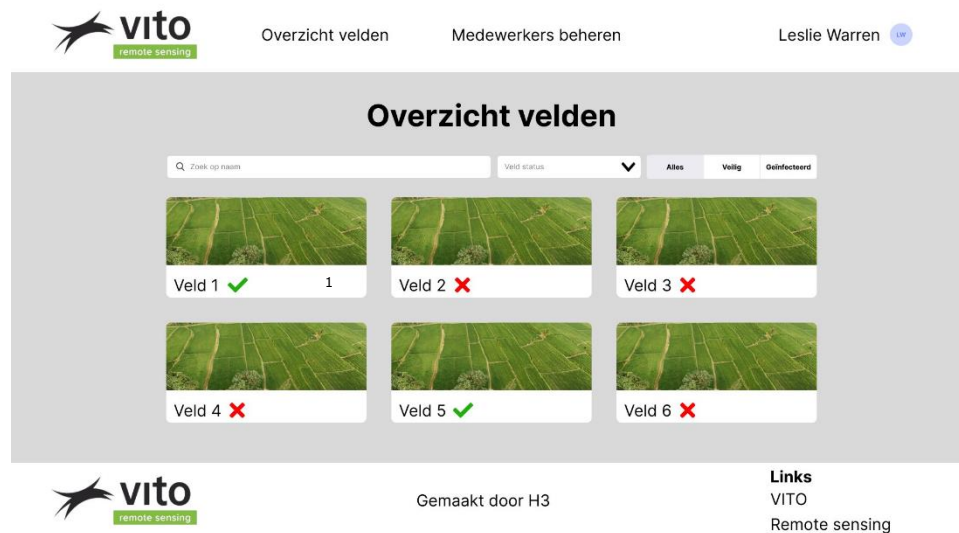
Functionaliteit: als boer, kan ik een overzicht van mijn velden opvragen.

Voorwaarde: de actor moet ingelogd zijn.

Normale werking: het systeem geeft een overzicht van alle velden.

Schermen:

3.1.4.1 Overzicht velden



Actie	Uitleg
1	Link naar pagina "3.1.5.1 Overzicht veld".

3.1.5 Overzicht veld

Functionaliteit: als boer, kan ik een heatmap opvragen.

Voorwaarde: de gebruiker is ingelogd.

Normale werking: het systeem laat een heatmap zien met wat extra informatie over het veld.

Schermen:

3.1.5.1 Overzicht veld



Actie	Uitleg
1	Link naar pagina "3.1.6.1 Overzicht veld".

3.1.6 Download SHAPE-file

Functionaliteit: als werker, kan ik een SHAPE-file downloaden.

Voorwaarde: als actor moet je ingelogd zijn.

Normale Werking: het systeem genereert een SHAPE-file en download het.

Schermen:

3.1.6.1 Overzicht veld

The screenshot shows the 'Overzicht veld 1' (Field Overview 1) interface. At the top, there is a navigation bar with the VITO logo (remote sensing) on the left, and links for 'Overzicht velden', 'Medewerkers beheren', and a user profile for 'Leslie Warren' on the right. The main content area has a title 'Overzicht veld 1' and a large satellite image of a field. A red circle highlights a specific area in the field. To the right of the image is a panel titled 'Extra informatie' (Extra information) containing the following details: 'Pastorijstraat 19', 'Mals', '25% van het veld bevat knolcypurus', and 'Veld van Jan Peeters'. Below this information is a green button labeled 'Download tractor bestand'. At the bottom of the interface, there is a footer with the VITO logo on the left, 'Gemaakt door H3' in the center, and a 'Links' section on the right listing 'VITO' and 'Remote sensing'.

3.1.7 Medewerkers beheren

Functionaliteit: als boer, kan ik mijn werkers beheren.

Voorwaarden: de actor is ingelogd.

Normale werking: het systeem geeft een overzicht van alle medewerkers. De actor drukt op de knop medewerker toevoegen. Het systeem open het formulier. De actor vult de gegevens in en het systeem slaat de gebruiker op.

Alternatieven:

Medewerker verwijderen: de actor klikt op het vuilnisbak icoon naast één van zijn medewerkers. Het systeem verwijdert de medewerker.

Schermen:

3.1.7.1 Medewerkers beheren

The screenshot displays the 'Medewerkers beheren' screen. At the top, there's a navigation bar with the VITO logo, 'Overzicht velden', 'Medewerkers beheren', and a user profile for Leslie Warren. The main content area has a title 'Mederwerkers beheren' (sic). Below the title is a search bar labeled 'Zoek op naam' and a green button 'Medewerker toevoegen'. A table lists three employees:

Voornaam	Achternaam	E-mail	
Jan	Peeters	jan.peeters@gmail.com	
Tom	Verachtert	tom.verachtert@gmail.com	
Toby	Claes	toby.claes@gmail.com	

The footer contains the VITO logo, 'Gemaakt door H3', and a 'Links' section with 'VITO' and 'Remote sensing'.

Actie	Uitleg
1	Link naar "3.1.7.2 Medewerker aanmaken".
2	Verwijdert de medewerker.

3.1.7.2 Medewerker aanmaken

The screenshot displays the 'Medewerkers beheren' (Manage Employees) interface. At the top, the VITO logo is on the left, and navigation links 'Overzicht velden' and 'Medewerkers beheren' are in the center. The user 'Leslie Warren' is logged in on the right. The main content area is titled 'Medewerkers beheren' and features a table with columns for 'Voornaam', 'Achternaam', and 'E-mail'. A modal window titled 'Medewerker toevoegen' (Add Employee) is open, showing a search bar and buttons 'Nodig uit' (Send invite) and 'Sluiten' (Close). The footer includes the VITO logo, 'Gemaakt door H3', and a 'Links' section with 'VITO' and 'Remote sensing'.

Actie	Uitleg
Nodig uit	Stuurt een uitnodigingsmail naar de medewerker
Sluiten	Sluit de modal

3.1.8 Gewassen beheren

Functionaliteit: als admin, kan ik de types van gewassen beheren.

Voorwaarde: de actor is ingelogd.

Normale werking: het systeem laat een overzicht zien van alle gewassen. De actor klikt op de knop gewas toevoegen. Het systeem toont het formulier. De actor vult deze in en het systeem maakt een nieuw gewas aan.

Alternatieven:

Gewas bewerken: het systeem laat een formulier zien. De actor vult de nieuwe gegevens in en het systeem past deze aan.

Gewas verwijderen: de actor klikt op het vuilbakje en het systeem verwijdert het gewas.

Schermen:

3.1.8.1 Gewassen beheren



Gewassen beheren

Gebruikers beheren

Foto uploaden

Gewassen beheren

Zoek op naam

Gewas toevoegen

Gewas

Maïs

Aardappelen

Graan



Actie	Uitleg
Gewas toevoegen	Link naar "3.1.7.2 Gewas aanmaken".
1	Link naar "3.1.7.2 Gewas aanmaken".
2	Verwijdert het gewas

3.1.8.2 Gewas aanmaken



Gewassen beheren

Gebruikers beheren

Foto uploaden

Gewassen beheren

Zoek op naam

Gewas toevoegen

Gewas

Maïs

Aardappelen

Graan



Gewas toevoegen

Naam

Naam

Toevoegen

Sluiten

Actie	Uitleg
Toevoegen	Voegt het gewas toe aan het systeem
Sluiten	Sluit de modal

3.1.9 Upload drone beelden

Functionaliteit: Een admin, kan ik foto's uploaden.

Voorwaarde: De actor is ingelogd

Normale Werking: De actor drukt op de knop, en selecteert de foto's om te uploaden. De actor selecteert ook de boer en het veld waarbij de foto's horen.

Schermen:

3.1.9.1 Drone beelden uploaden

Actie	Uitleg
Uploaden	De foto's worden toegevoegd op het systeem
Stoppen	Verwijdert de geselecteerde foto's

3.1.10 Gebruikers beheren

Functionaliteit: als admin, kan ik de gebruikers beheren.

Voorwaarde: De actor is ingelogd

Normale Werking: Het systeem laat een lijst met alle gebruikers zien. De actor klikt op de knop gebruiker toevoegen. Het systeem laat een formulier zien. De actor vult deze in en het systeem voegt de nieuwe gebruiker toe.

Alternatieven:

Gebruiker aanpassen: het systeem laat een formulier zien. De actor doet de nodige aanpassingen en het systeem slaat deze op.

Gebruiker verwijderen: het systeem verwijdert de gebruiker

Schermen:

3.1.10.1 Gebruikers beheren



Gewassen beheren







Gebruikers beheren

Foto uploaden

Gebruikers beheren

Zoek op naam

Gebruiker toevoegen

Voornaam	Achternaam	E-mail	Rol	
Jan	Peeters	jan.peeters@gmail.com	Boer	 
Tom	Verachert	tom@gmail.com	Expert	 
Toby	Claes	toby.claes@gmail.com	Overheid	 

Actie	Uitleg
Gebruiker toevoegen	Link naar "3.1.10.2 Gebruiker toevoegen".
1	Link naar "3.1.10.2 Gebruiker toevoegen".
2	Verwijdert de gebruiker uit het systeem

3.1.10.2 Gebruiker toevoegen



Gewassen beheren







Gebruikers beheren

Foto uploaden

Gebruikers beheren

Zoek op naam

Gebruiker toevoegen

Voornaam	Achternaam	E-mail	Rol	
Jan	Peeters	jan.peeters@gmail.com	Boer	 
Tom	Verachert			 
Toby	Claes			 

Gebruiker toevoegen

Voornaam

Achternaam

E-mail

Rol

Actie	Uitleg
Toevoegen	Voegt de gebruiker toe.
Sluiten	Sluit de modal.

3.1.11 Velden beheren

Functionaliteit: als admin, kan ik de velden beheren.

Voorwaarde: De actor is ingelogd.

Normale Werking: Het systeem laat een lijst met alle velden zien. De actor klikt op de knop veld toevoegen. Het systeem laat een formulier zien. De actor vult deze in en het systeem voegt het nieuwe veld toe.

Alternatieven:

Veld aanpassen: het systeem laat een formulier zien. De actor doet de nodige aanpassingen en het systeem slaat deze op.

Veld verwijderen: het systeem verwijdert de gebruiker.

Schermen:

3.1.11.1 Velden beheren



Gewassen beheren

Gebruikers beheren

Foto uploaden

Velden beheren

Zoek op boernaam
 Veld toevoegen

Boer	Straatnaam	Gewassen	
Jan Peeters	Kleinhoefstraat 4	Gewas	
Renée Koopman	Standonk 21	Gewas	
Arno Addens	Doornboomstraat 11	Gewas	

Actie	Uitleg
Veld toevoegen	Link naar "3.1.11.2 Veld toevoegen".
1	Link naar "3.1.11.2 Veld toevoegen".
2	Verwijdert het veld uit het systeem

3.1.11.2 Veld toevoegen



Gewassen beheren

Gebruikers beheren

Foto uploaden

Velden beheren

Q Zoek op boernaam Veld toevoegen

Boer	Straatnaam	Gewassen	
Jan Peeters	Kleinhoefstraat 4	Gewas	
Renée Koopman	Stand		
Arno Addens	Doornik		

Veld toevoegen

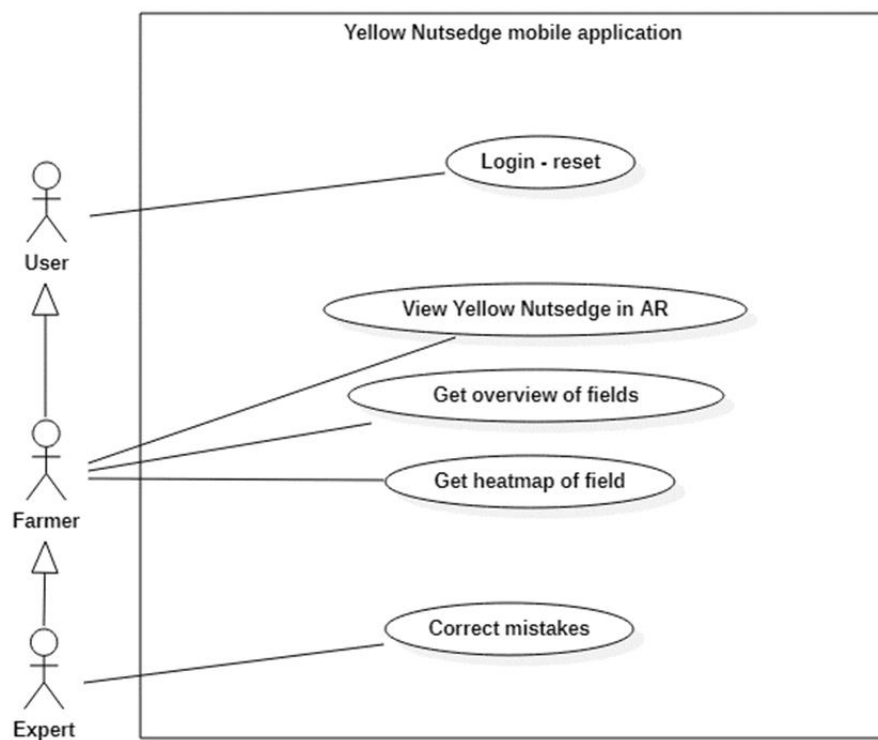
Boer

Gewas

Toevoegen Sluiten

Actie	Uitleg
Toevoegen	Voegt het veld toe.
Sluiten	Sluit de modal.

3.2 Usecase Diagram Mobile Applicatie



3.2.1 Login

Schermen:

3.2.1.1 Login

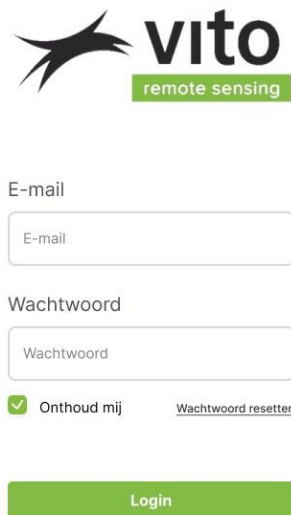
Functionaliteit: als gebruiker, kan ik me inloggen.

Normale werking: het systeem laat het login scherm zien. De gebruiker vult zijn inloggegevens in en klikt op login.

Alternatieven:

Foute inloggegevens: het systeem laat zien dat de inloggegevens fout zijn.

Wachtwoord resetten: het systeem gaat naar de pagina "Wachtwoord resetten".



Actie	Uitleg
Login	Link naar pagina "3.2.3.1 Overzicht velden".

3.2.2 Wachtwoord resetten

Schermen:

3.2.2.1 Wachtwoord resetten

Functionaliteit: als gebruiker, kan ik mijn wachtwoord resetten.

Normale werking: het systeem laat de pagina om het wachtwoord te resetten zien. De actor geeft het nieuwe wachtwoord in en klikt op wachtwoord resetten.



Wachtwoord

Bevestig wachtwoord

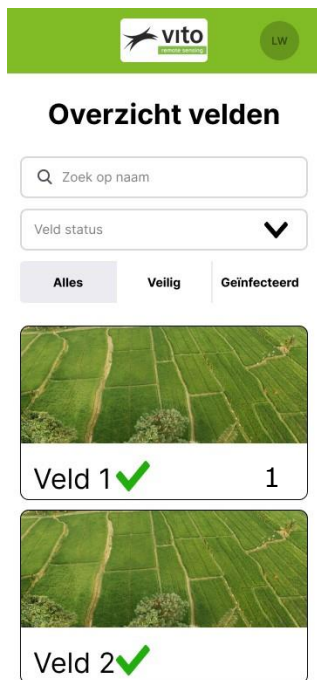
Wachtwoord resetten

Actie	Uitleg
Wachtwoord resetten	Link naar pagina "3.2.1.1 Login".

3.2.3 Overzicht velden

Schermen:

3.2.3.1 Overzicht velden



Functionaliteit: als boer, kan ik een overzicht van mijn velden zien.

Voorwaarde: de actor is ingelogd.

Normale werking: het systeem laat een lijst van velden zien.

Actie	Uitleg
1	Link naar pagina "3.2.4.1 Overzicht veld".

3.2.4 Heatmap van een veld

Schermen:

3.2.4.1 Overzicht veld



Overzicht veld1



Pastorijstraat 19

Maïs

25% van het veld bevat knolcyperus

Functionaliteit: als boer, kan ik een een heatmap van mijn veld zien.

Normale werking: het systeem laat een heatmap van het veld zien en extra informatie.



Actie	Uitleg
1	Link naar pagina "3.2.3.1 Overzicht velden".
2	Link naar pagina "3.2.5.1 AR"

3.2.5 Bekijk knolcyperus in AR

Schermen:

3.2.5.1 AR



Functionaliteit: als boer, kan ik knolcyperus in AR zien.

Voorwaarde: de actor is ingelogd.

Normale werking: het systeem opent de AR en duidt de knolcyperus aan.

Actie	Uitleg
1	Link naar pagina "3.2.4.1 Overzicht veld".

3.2.6 Fouten corrigeren

Schermen:

3.2.6.1 AR



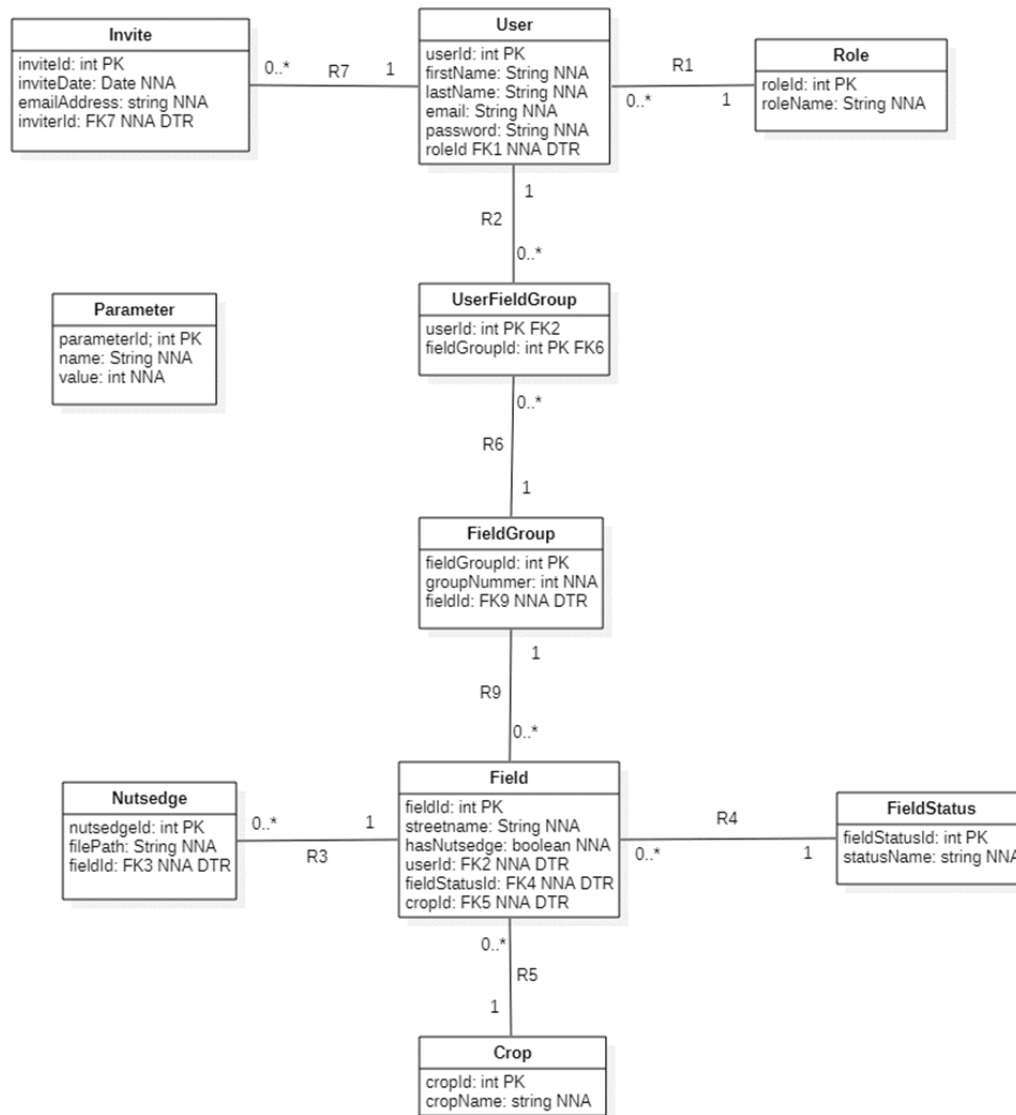
Functionaliteit: als expert, kan ik foute metingen corrigeren.

Voorwaarde: de actor is ingelogd.

Normale werking: het systeem geeft een waarschuwing. De actor bevestigt en het systeem corrigeert de foute meting.

Actie	Uitleg
1	Link naar pagina "3.2.4.1 Overzicht veld".
2	Corrigeer een foute meting

4 ERD



Ook hebben we een ERD opgesteld voor onze database structuur. De ERD bestaat uit de volgende tabellen, Invite, User, Role, UserFieldGroup, FieldGroup, Field, Nutsedge, FieldStatus, Parameter en Crop.

De Crop tabel is de tabel voor de verschillende types gewassen van de velden, bijvoorbeeld mais. Vervolgens hebben we dat gelinkt naar een veld, met een 1 op veel relatie, dit wil zeggen dat een veld 1 gewas kan hebben, en een gewas kan aan meerdere velden vastzitten. Vervolgens hebben we de Field tabel, over elk veld slaan we de straatnaam op, of het knolcyperus bevat, de id van de user die bij het veld hoort, en welk gewas er op staat.

Elk veld is ook verbonden met een FieldStatus, dit is een 1 op veel relatie, dit wil zeggen dat een veld 1 status kan hebben, maar een status kan bij meerdere velden voorkomen. Daarnaast hebben we de Nutsedge tabel, die heeft een veel op 1 relatie met de Field tabel, Dit wil zeggen dat een veld meerdere knolcyperus entries kan bevatten, maar een knolcyperus entry kan maar op 1 veld voorkomen. Voor de Nutsedge tabel slaan we een pad op naar een foto van de knolcyperus, en het id van het veld waarop de knolcyperus zit.

De FieldGroup tabel is een tabel waar we velden ids opslaan met een nummer, zo weten we welke velden precies bij elkaar horen. Vervolgens hebben we de tussentabel UserFieldGroup. Deze tabel zorgt ervoor dat elke user gelinkt kan worden met een FieldGroup.

We hebben ook een User tabel, van een user slaan we de voornaam, achternaam, email, password, en een roleid op. Deze tabel heeft een 1 op veel relatie met de Role tabel, dit wil zeggen dat elke user 1 rol kan hebben, maar een rol tot meerdere users kan behoren.

Vervolgens heeft de User tabel ook veel op 1 relatie met de Invite tabel, dit wil zeggen dat een user meerdere invites kan sturen, maar een invite maar bij 1 user kan horen. Deze tabel slaat de datum op van de uitnodiging, het email adres naar waar de uitnodiging gestuurd zal worden, en het id van wie de invite gestuurd heeft.

Tot slot hebben we ook de parameter tabel, dit is een losstaande tabel waar we parameters gaan opslaan, zoals hoelang een invite geldig is.

5 AI MODEL WERKING

5.1 Trainen

Om ons AI-model te trainen gaan we de beelden van VITO gebruiken om tegen ons model te zeggen welke foto's wel knolcyperus bevatten en welke niet. Dit zal niet rechtstreeks met de beelden van VITO gaan aangezien deze te groot zijn en een incompatibel bestandstype (TIF) hebben om een model te trainen.

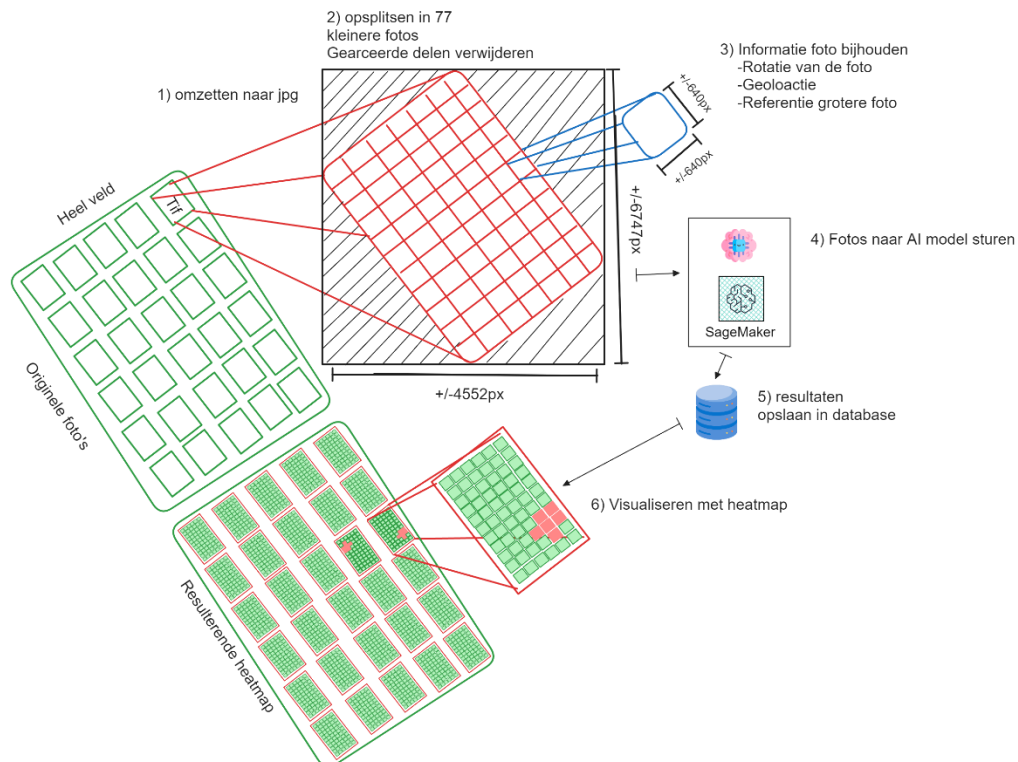
Daarom zullen we elke foto opsplitsen in +/-77 kleinere foto's om zo een duidelijker beeld te geven waar de knolcyperus staat en om het model makkelijker te kunnen trainen. We zullen al deze kleinere foto's moeten sorteren om tegen het model te zeggen deze foto bevat wel knolcyperus en deze niet. Het model zal dan de patronen van die foto's leren om deze later te herkennen in de nieuwe beelden.

De beelden dat gebruikt worden om te trainen zullen een eerlijke verhouding hebben tussen foto met knolcyperus en foto zonder knolcyperus zodat het model de patronen eerlijk leert zonder "bias".

5.2 Testen

Om het model te testen zullen we beelden gebruiken dat het model nog nooit gezien heeft. Deze beelden worden ook opgedeeld in kleinere foto's zodat het model hetzelfde soort foto's als de training set heeft.

5.3 Visuele weergave van de werking



6 SHAPEFILE TRACTOR

Om de shapefile van de tractor te maken hebben we de resultaten nodig van het AI-model in de vorm van een CSV. We kunnen per individuele foto zeggen of we de coördinaten nodig hebben in de shape file en ook of we omringende coördinaten willen weglaten in het geval van aanwezige knolcyperus. Deze CSV-file moeten we dan aan een package geven die dan de CSV-file omzet in de daadwerkelijke shapefile. Het omzetten kunnen we implementeren in Python of JavaScript.

Voor Python kunnen we de PyShp package gebruiken, voor JavaScript zouden we de Shp-write library kunnen gebruiken.

7 CONCLUSIE

In dit rapport hebben we een gedetailleerde blauwdruk geschetst voor Project 4.0, waarbij diverse technologiepakketten zijn samengebracht om een oplossing te ontwikkelen voor de klant VITO. Het project richt zich op het aanpakken van het probleem van de aanwezigheid van de plant "Knolcyperus" op boerenvelden in België.

Ons team, bestaande uit Application Development (APP), Cloud & Cyber Security (CCS), en Artificial Intelligence (AI) studenten, heeft zorgvuldig overwogen welke technologieën het meest geschikt zijn voor elk aspect van het project. We hebben een weloverwogen besluitvormingsproces gevolgd en gedocumenteerd in Weighted Decision Matrix'en (WDM), om zo de beste keuzes te maken voor het hostingplatform, de webserver, de database, het AI-model, en zowel de frontend als backend van de webapplicatie.

De architecturale voorstelling toont de implementatie van de oplossing op AWS. Gehele levenscyclus van het project wordt beheerd volgens de DevOps-lifecycle, dit houdt in: plannen, coderen, bouwen, testen, releasen, implementeren, opereren en monitoren.

Het Usecase-diagram geeft een gedetailleerd overzicht van de functionaliteiten van zowel de web- als mobiele applicatie, met specifieke usecases zoals inloggen, het verkrijgen van heatmaps van velden, het beheren van werkers, en het uploaden van foto's.

Tot slot hebben we een Entity-Relationship Diagram (ERD) gepresenteerd dat de databasestructuur weergeeft, inclusief tabellen voor gebruikers, rollen, velden, gewassen, knolcyperusinformatie en meer.

Dit rapport biedt een gestructureerde en diepgaande blik op de conceptfase van Project 4.0. Het document dient als leidraad voor ons team om het project succesvol te leiden en een oplossing te leveren die voldoet aan de behoeften van VITO.

ⁱ Een TIF-bestand is een veelgebruikt formaat voor het opslaan van afbeeldingen. Wanneer het wordt uitgebreid met GeoTIFF-functionaliteit, kan het ook geolocatie-informatie bevatten, zoals coördinaten en projectiegegevens. Dit maakt het geschikt voor toepassingen zoals GIS, cartografie en satellietbeelden waar geografische precisie belangrijk is.

ⁱⁱ Bias in een model wil zeggen dat het model meer weet over een bepaald kenmerk. In ons geval zou het meer kunnen weten over de kenmerken van een foto zonder knolcyperus dan foto's met knolcyperus. Dit zorgt ervoor dat de test resultaten niet zo nauwkeurig/correct zullen zijn.