



Mood Secure

mediaan.

Audio based crowd monitoring system

Zico Vanheuckelom

Rikko Keuppens

Stan Swinnen

Josse Van Looy

Els Bresseleers

Senne Van Meensel

Thomas More | 2022 - 2023

Voorwoord

Tijdens het laatste jaar van de bacheloropleiding IT Factory te Thomas More Geel worden de studenten uitgedaagd om zich volledig te verdiepen in een extern project. Hierbij worden alle krachten van de verschillende afstudeerrichtingen met elkaar gebundeld om een ijzersterk team op te stellen die het project volledig onder handen nemen van A tot Z.

Voor ons specifiek, Mood Secure, houdt dit in dat we met studenten van Application Development, Cloud & Cyber security en Artificial Intelligence het project van Mediaan uit Hasselt tot een volwaardig einde moeten volbrengen.

Graag willen wij toch even de tijd nemen om enkele mensen te bedanken die ons hebben bijgestaan om dit project tot een goed einde te brengen.

- Christine Smeets
 - o Mevr. Smeets is tijdens het semester onze coach geweest. Zij heeft ons het project aangebracht en ons steeds in de juiste richting helpen denken.
- Mediaan, met in het specifiek Tom Bosmans & Sébastien Bodart
 - o Dit was onze opdrachtgever en ook de twee personen die ons hebben geholpen om het project vorm te geven alsook het mogelijk gemaakt hebben inkijk te krijgen in een gelijkaardig project om onderzoek te kunnen voeren.
- Bram Verbruggen
 - o Mr. Verbruggen is een docent op de Thomas More Kempen en specialist op gebied van Cloud security. Omdat ons project toch enige security vraagt was Mr. Verbruggen een aangenaam aanspreekpunt om ons hiermee op weg te helpen.
- Bram Heyns
 - o Mr. Heyns is ook een docent op de Thomas More Kempen die gefocust is op de allernieuwste snufjes binnen het AI-werkveld. Een groot deel van ons project omvat AI, dus ook hij was een aanspreekpunt om deze specialisaties uit te kunnen werken.
- Mood Secure
 - o We kunnen elkaar ook bedanken voor deze leuke en leerrijke ervaring.

Inhoudsopgave

.....	0
Voorwoord	1
Inhoudsopgave	2
1. Afkortingenlijst	5
2. Inleiding	6
2.1. Teamvoorstelling	6
2.2. Voorstelling opdrachtgever	8
2.3. Projectvoorstelling	8
3. Onderzoek huidig crowd monitoringsysteem gebaseerd op Computer Vision	9
3.1. Functionele eisen Computer Vision systeem	9
3.1.1. Demo 1 – prediction on image	9
3.1.2. Demo 2 – prediction on video stream	9
3.1.3. Voordelen van beide demo's	10
3.1.4. Nadelen van beide demo's	10
3.2. Verwerking van data	10
3.2.1. Verwerking demo 1	10
3.2.2. Verwerking demo 2	11
3.3. Onderzoek aangeleverde Azure omgeving	11
4. Cloudapplicatie-ontwerp	12
4.1. Datamodel	12
4.2. Use Case diagram	13
4.3. Functionele eisen	15
4.3.1. Use case 0: Inloggen	15
4.3.2. Use case 1: uitnodigingsmail sturen naar teamleads	17
4.3.3. Use case 2: Evenement aanmaken & beheren	18
4.3.4. Use case 3: Teamlead koppelen aan een evenement	19
4.3.5. Use case 4: Account aanmaken & beheren	20
4.3.6. Use case 5: Applicatie opstarten/uitzetten	21
4.3.7. Use case 6: Uitnodigingsmail sturen naar securitypersoneel	22
4.3.8. Use case 7: Security toewijzen aan een evenement	23
4.3.9. Use case 8: Locaties toewijzen aan een evenement	24
4.3.10. Use case 9: Microfoons toewijzen aan een evenement	25
4.3.11. Use case 10: Microfoons toewijzen aan een locatie op een evenement	25
4.3.12. Use case 11: Dashboards bekijken	26

4.3.13.	Use case 12: Alerts bekijken	26
4.3.14.	Use case 13: Personeel uitsturen naar een locatie	27
4.3.15.	Use case 14: Geschiedenis opvragen	28
4.3.16.	Use case 14: Securitypersoneel kan zich registreren	29
4.3.17.	Use case 15: confirmeren om naar een locatie te gaan	29
4.3.18.	Use case 16: Feedback geven op locatie	30
4.4.	Technische keuzes	31
4.4.1.	Front-end keuze - Vue	31
4.4.2.	Back-end keuze - .NET	31
4.5.	Conclusie	31
5.	Server- en netwerkontwerp	32
5.1.	DBMS-keuze – MySQL	32
5.2.	Storage.....	32
5.3.	Automatisatie	33
5.4.	Back-up	33
5.5.	Azure componenten vergelijkingstabellen.....	33
5.5.1.	App omgeving.....	33
5.5.2.	Storage.....	35
5.5.3.	Database.....	35
5.6.	Schematische voorstelling omgeving	35
1.1.	Front-end.....	35
5.7.	Geschatte kostenplaatje.....	36
5.8.	Conclusie	36
6.	AI-ontwerp.....	37
6.1.	Dataset	37
6.1.1.	Event detection	37
6.1.2.	Mood detection.....	37
6.1.3.	Audio labeling.....	38
6.2.	Algoritme	40
6.3.	MLOPs.....	42
6.3.1.	De MLOPs cyclus.....	42
6.3.2.	De componenten van MLOPs	43
6.4.	Toepassing in applicatie	44
6.5.	Dashboard prototype	45
6.6.	Conclusie	45
7.	Algemeen besluit.....	46

Figurenlijst	47
Referenties	48

1. Afkortingenlijst

- APP = Application Development
- AI = Artificial Intelligence
- CCS = Cloud & Cyber Security
- GDPR = General Data Protection Rules
- WRM = Weighted Ranking Method
- VM = Virtual Machine
- API = Application Programming Interface
- EDA = Exploratory Data Analysis
- CNN = Convolutional Neural Networks
- CML = Continuous Machine Learning
- MLOPs = Machine Learning Operations

2. Inleiding

Alvorens er wordt overgegaan op de volledige uitwerking van de conceptfase van dit project delen we graag nog enkele algemene zaken mee.

Disclaimer. Dit document omvat de theoretische voorbereiding van het project. Op het moment van schrijven is dit project enkel theoretisch uitgewerkt, de praktische uitwerking volgt.

2.1. Teamvoorstelling

Voor u, de lezer, zijn wij natuurlijk onbekend. Daarom stellen we ons graag even kort aan u voor.



Hallo! Ik ben Zico Vanheuckelom, student Toegepaste Informatica, orthopedisch technoloog en natuurliefhebber. Ik ben graag in de natuur maar ook het tegenovergestelde inspireert me. Zien hoe de wereld verandert door IT moedigt mij alleen maar aan om hier deel van uit te maken. Terwijl ik weet hoe het hele menselijk lichaam in elkaar zit en hoe complex dit is, ben ik ook het IT-pad ingeslagen om deze overgrote wereld meer bekendheid te geven.



Hey! Ik ben Rikko Keuppens, student Toegepaste Informatica met afstudeerrichting Application Development. Ik ben een sociaal en sportief persoon met een passie om applicaties te ontwikkelen. Dit is dan ook de reden waarom ik mijn richting heb gekozen. De groeiende toekomst van de IT-wereld zorgt ervoor dat ik alsmaar meer gemotiveerd ben om mooie eindproducten te ontwerpen.



Goedendag, ik ben Josse Van Looy. Momenteel ben ik een student aan Thomas More, waar ik toegepaste informatica studeer. Als afstudeerrichting heb ik gekozen voor Applied Computer Science, omdat ik zeer veel interesse heb in alles dat te maken heeft met Artificial Intelligence. Hoewel dat ik me vaak achter mijn laptop bevind, ben ik een redelijk sportief persoon die graag eens gaat lopen en in de vakanties gaat bergwandelen.



Aangenaam, ik ben Stan Swinnen, student Applied Computer Science met als specialisatie Artificial Intelligence. Ik verdiep me graag in alles wat met data en datavisualisatie te maken heeft. Dat is ook de reden waarom ik voor AI gekozen heb. Als ik dan uiteindelijk genoeg heb gekregen van die computer, kan je me buiten in de natuur of in de bergen vinden om te ontspannen.



Hey! Ik ben Els Bresseleers, studente Elektronica ICT met afstudeerrichting Cloud & Cybersecurity. Ik ben ook bakker en zorgkundige maar IT zegt me toch iets meer. Ik ben een enthousiast persoon die af en toe haar overschot aan energie kwijt moet met Zumba, zwemmen of reizen. De wereld wordt steeds technischer en steeds meer data komt online of in de Cloud te staan. Ik wil ervoor zorgen dat mensen ook een veilig gevoel online krijgen.



Hallo, ik ben Senne Van Meensel, een student Elektronica ICT met als keuzetraject Cloud & Cybersecurity. Voor ik aan deze opleiding begon heb ik eerst een graduaatsdiploma Systeem en Netwerkbeheer behaald. Ik ben een sportief persoon en gepassioneerd door IT en security.

2.2. Voorstelling opdrachtgever

Mediaan is een bedrijf gelegen te Hasselt met de focus op business transformatie, technologische innovatie en data-driven organisaties. Ze zijn een jong bedrijf met een rijke geschiedenis. Ze werken voor internationale klanten en veelbelovende startups. Binnen Mediaan wordt u omringt door leuke collega's en de sfeer op hun kantoor is informeel en ontzettend plezierig.

Met een 170-tal werknemers volgens LinkedIn behoort Mediaan tot de middelgrote bedrijven, dit zorgt dat ze tal van expertise hebben op alle mogelijke vlakken. Hierdoor kunnen ze dan ook projecten realiseren in meerdere IT-omgevingen. Toch ligt hun focus voornamelijk op Azure, .NET in combinatie met VueJS/React en Python.

2.3. Projectvoorstelling

Veiligheid is voor iedereen een zeer belangrijke waarde en is praktisch niet meer weg te denken uit de hedendaagse maatschappij. Overal zijn er voorzieningen om dingen alsmaar veiliger te laten voorlopen. Denk bv aan alle veiligheidsnormen die genomen zijn tijdens uitbraak van de pandemie, aan alle regels en wetten die opgesteld zijn door regeringen om de veiligheid van hun inwoners te garanderen, aan de wetshandhavers die patrouilleren bij gevaarlijke situaties, ... Maar ook op evenementen zoals voetbalwedstrijden of festivals is veiligheid niet meer weg te denken. Hier is ons project dan ook op gebaseerd.

Vandaag de dag worden dergelijke evenementen beveiligd door securitypersoneel die patrouilleren op het domein, enkel met elkaar in contact via radiocommunicatie en misschien enkele geplaatste camera's op het domein. Hierdoor kan de veiligheid niet volledig gegarandeerd worden. Iedereen kan het securitypersoneel zien aankomen, iedereen ziet de camera's en de dode hoek. Dit zorgt nog steeds voor gevaarlijke situaties. Maar kan dit veiliger? Kan dit op een andere manier? Dat is de vraag die Mediaan ons oplegde en waar wij mee aan de slag gingen.

Een audio based crowd monitoring system, of beter gezegd: een systeem gebaseerd op geluid om evenementen te kunnen monitoren. Dit systeem moet geluid kunnen detecteren in functie van het terrein, vervolgens moet er een techniek gebruikt worden om dit geluid te kunnen classificeren zodat het bruikbaar is voor algoritmes. Deze classificatie kan dan behandeld worden door een verantwoordelijke door middel van de resultaten weer te geven in een webapplicatie. De teamlead kan dan op zijn beurt het securitypersoneel uitsturen naar de juiste locatie op het domein. Het securitypersoneel krijgt dan een berichtje op zijn of haar telefoon met de instructies die hij of zij moet uitvoeren.

Concreet zou er op het terrein van het evenement microfoons geplaatst moeten worden. Omdat in Mood Secure er geen enkele Internet Of Things-student zit, mag ervan uit gegaan worden dat de flow begint bij binnenkomende audiofragmenten. Er zal dus geen configuratie plaatsvinden van microfoons. Echter in een realistisch scenario zullen deze dus mee opgenomen moeten worden.

Alle conclusies en resultaten die verkregen worden door de AI-algoritmes en met behulp van Flask naar de applicatie gestuurd worden, moeten voor een tijdsperiode van twee jaar beschikbaar blijven. Dit brengt met zich mee dat er back-up voorzieningen moeten geïmplementeerd worden.

Aan dit project waren ook enkele algemene eisen verbonden waar wij rekening mee moesten houden. Zo waren volgende eisen opgelegd door Mediaan:

- Een budget van ongeveer €200/maand
- Het systeem moet uitbreidbaar en aanpasbaar zijn
- Er moet een optimale UI/UX opgebouwd worden

- Het systeem moet zich afspelen in een Azure en .NET omgeving
- Er wordt gerekend dat een drie teamleads en ongeveer 100-tal beveiligingspersoneel gebruik maken van het systeem
- Er moet clean code met documentatie opgeleverd worden bij de realisatiefase.

Dit alles kan onderverdeeld worden in vier grote onderdelen, deze vormen ook de basis van dit document.

- Onderzoek huidig crowd monitoringsysteem gebaseerd op Computer Vision
- Cloudapplicatie-ontwerp
- Server- en netwerkontwerp
- AI-ontwerp

3. Onderzoek huidig crowd monitoringsysteem gebaseerd op Computer Vision

Zoals eerder vermeld in het voorwoord heeft Mediaan alvast een project dat gelijkend is op wat er wordt verwacht van Mood Secure. Echter is dit project niet gebaseerd op audio, maar op Computer Vision. Mediaan heeft ons de mogelijkheid gegeven om een diepere inkijk te krijgen in deze code zodanig dat wij een idee zouden hebben van hoe het project er kan uitzien. Mediaan heeft ons code ter beschikking gesteld die onze APP studenten hebben uitgespit op zoek naar de functionele eisen alsook de voor- en nadelen. Onze AI studenten hebben deze code dan weer gebruikt om te achterhalen hoe de beelden juist werden omgezet. Ook heeft Mediaan ons een Azure Cloud omgeving aangeboden, die onder handen werd genomen door onze CCS studenten.

Er werden twee demo's aangeleverd. De eerste demo was een systeem op basis van foto's, de tweede demo was een systeem op basis van een videostream.

3.1. Functionele eisen Computer Vision systeem

Het systeem van Computer Vision gemaakt door Mediaan heeft enkele functionele eisen die hieronder kort opgesomd worden.

3.1.1. Demo 1 – prediction on image

- Als user moet u een afbeelding kunnen opladen.
- Het systeem moet een afbeelding kunnen lezen.
- Het systeem moet op een afbeelding een algoritme kunnen loslaten.
- Het systeem moet de verschillende objecten op een afbeelding kunnen onderscheiden.
- Het systeem moet rond deze objecten een kader kunnen tekenen.
- Het systeem moet een afbeelding kunnen opslaan op een harde schijf.
- Het systeem moet een tekstbestand met de bekomen resultaten als output kunnen geven.
- Het systeem moet de verkregen informatie kunnen opslaan op een harde schijf.
- Het systeem moet de verkregen informatie kunnen opslaan in de Cloud.
- Het systeem moet met een percentage kunnen aanduiden hoe groot de zekerheid is van het getraind model.

3.1.2. Demo 2 – prediction on video stream

- Het systeem moet constant binnenkomende video kunnen renderen en lezen.
- Het systeem moet op een video een algoritme kunnen loslaten.
- Het systeem moet in real time kunnen herkennen wat het object is in de video.
- Het systeem moet rond deze objecten een kader kunnen tekenen.

- Het systeem moet met een percentage weergeven van hoe groot de zekerheid is van het getraind model.
- Het systeem mag geen delay hebben.

3.1.3. Voordelen van beide demo's

- De code is niet heel uitgebreid en makkelijk te begrijpen.
- Het systeem werkt vrij goed en herkent de meeste foto's.
- Het systeem herkent mensen en objecten op video.
- Het systeem is een goed voorbeeld voor hoe we de verkregen info opslaan op een harde schijf en in de Cloud.

3.1.4. Nadelen van beide demo's

- De code is specifiek om videobeelden en afbeeldingen te herkennen en gaat dus niet gebruikt kunnen worden voor audiodata.
- Niet alles wordt herkend. In een afbeelding met meerdere mensen of andere dingen die herkend kunnen worden, wordt meestal om en bij de 20% van de objecten niet herkend.
- Het live filmen van mensen kan problemen geven over GDPR.

3.2. Verwerking van data

De verwerking van data gebeurt natuurlijk niet zomaar, hier zit een AI-algoritme achter die onze AI-studenten tot in het detail hebben bestudeerd.

3.2.1. Verwerking demo 1

Samenhang: Vermoedelijk via de Cloud, dit is niet duidelijk in de demo.

Werking: Dit kan omschreven worden in een normal flow:

1. Input: load image
2. Model inference: object detection
3. Custom logic: detect objects on image
4. Output: save data to disk or Cloud

Analyse van code:

Als eerste worden de packages van Computer Vision geïmporteerd. De details van de Architectuur van het model en de vooraf getrainde weights worden in variabelen opgeslagen. De eigenschappen van de te detecteren objecten in videobeelden worden opgeslagen in een lijst. In de variabele "net" wordt het vooraf getrainde model aangesproken.

Als er geen GPU-support is, dan worden de CPU-specifieke backend en target van CV2 gebruikt. Als er wel GPU-support is worden de GPU-specifieke backend en target van CV2 gebruikt.

Vervolgens wordt de afbeelding in de applicatie geladen en wordt de grootte hiervan aangepast. Hierna worden de hoogte en breedte van de afbeelding genomen en in variabelen opgeslagen. De blob wordt als input gebruikt voor het model. Een blob is een wikkkel rond daadwerkelijke gegevens die worden verwerkt en doorgegeven door Caffe. Wanneer het model zijn werk gedaan heeft, wordt de output, zijnde de locaties van de gedetecteerde objecten, in een array gestoken.

Vervolgens worden random kleuren gegeven aan de verschillende eigenschappen. Deze worden opgeslagen in een variabele. Er worden steeds drie kleuren aan een eigenschap gegeven en één van die drie wordt willekeurig gegeven aan elke eigenschap. De for-loop gaat met een lus over elk

gevonden voorwerp in de afbeelding gaan. Als de betrouwbaarheid groter is dan 50% gaat er een kader getekend worden rond de gedetecteerde onderwerpen met hun random bepaalde kleur en de naam van dit gedetecteerde object.

Als laatste wordt bepaald hoeveel objecten er gedetecteerd zijn en zullen deze opgeslagen worden in een tekstbestand.

3.2.2. Verwerking demo 2

Samenhang: Vermoedelijk via de Cloud, ook dit is weer niet duidelijk in de demo.

Werking: Wederom kan dit omschreven worden in een normal flow:

1. Input: load camera stream
2. Model inference: object detection on camera stream
3. Custom logic: detect objects on images
4. Output: show images with object detection in real time

Analyse van code:

Als eerste worden de packages van Computer Vision geïmporteerd. De details van de Architectuur van het model en de vooraf getrainde weights worden in variabelen opgeslagen. De eigenschappen van de te detecteren objecten in videobeelden worden opgeslagen in een lijst. In de variabele "net" wordt het vooraf getrainde model aangesproken.

Als er geen GPU-support is, worden de CPU-specifieke backend en target van CV2 gebruikt. Als er wel GPU-support is worden de GPU-specifieke backend en target van CV2 gebruikt. Hierna worden de hoogte en breedte van de videostream genomen en in variabelen opgeslagen.

Vervolgens wordt een videocapture opgestart. Zolang deze videocapture aan staat, zal de video frame per frame in een blob gestoken worden. Op deze blobs wordt dan het model losgelaten. Vervolgens worden random kleuren gegeven aan de verschillende eigenschappen. Deze worden opgeslagen in een variabele. Er worden steeds drie kleuren aan een eigenschap gegeven en één van die drie wordt willekeurig gegeven aan elke eigenschap.

De for-loop gaat met een lus over elk gevonden voorwerp in de afbeelding gaan. Als de betrouwbaarheid groter is dan 50%, gaat er een kader getekend worden rond de gedetecteerde objecten met hun random bepaalde kleur alsook de naam van dit gedetecteerde object. Zolang de while-loop loopt, wordt het resultaat live weergegeven. Als er op escape gedrukt wordt, stopt de while-loop.

Als de while-loop stopt, stop ook de video-opname.

3.3. Onderzoek aangeleverde Azure omgeving

Eén van de vereisten was dat het geheel gehost zou worden in een Azure omgeving. Omwille van deze reden heeft Mediaan ons hun Azure pakket aangeboden om te kijken wat er hiermee allemaal mogelijk is.

Zo weten wij dat een CI/CD via Azure niet mogelijk zal zijn, dit zit niet inbegrepen in het abonnement van Mediaan. We zullen dus zelf een andere methode moeten zoeken.

[Alle services >](#)

Azure DevOps ...

We've made it easier to manage Azure DevOps billing and subscriptions. You can [set up billing](#), [change your subscription](#) or [pay for more users and resources](#) within Azure DevOps. [Learn more](#)

Figuur 1: CI/CD pipeline Azure Mediaan

Het is voor ons ook niet mogelijk om een container aan te maken binnen Azure.

Mediaan development Computer Vision

▼

✖

Resourceprovider(s): Microsoft.ContainerInstance zijn niet geregistreerd voor het abonnement Mediaan development Computer Vision en u hebt geen machtigingen om een resourceprovider voor het abonnement Mediaan development Computer Vision te registreren

Project-TM_group2

▼

Nieuw

Figuur 2: Containers Azure Mediaan

4. Cloudapplicatie-ontwerp

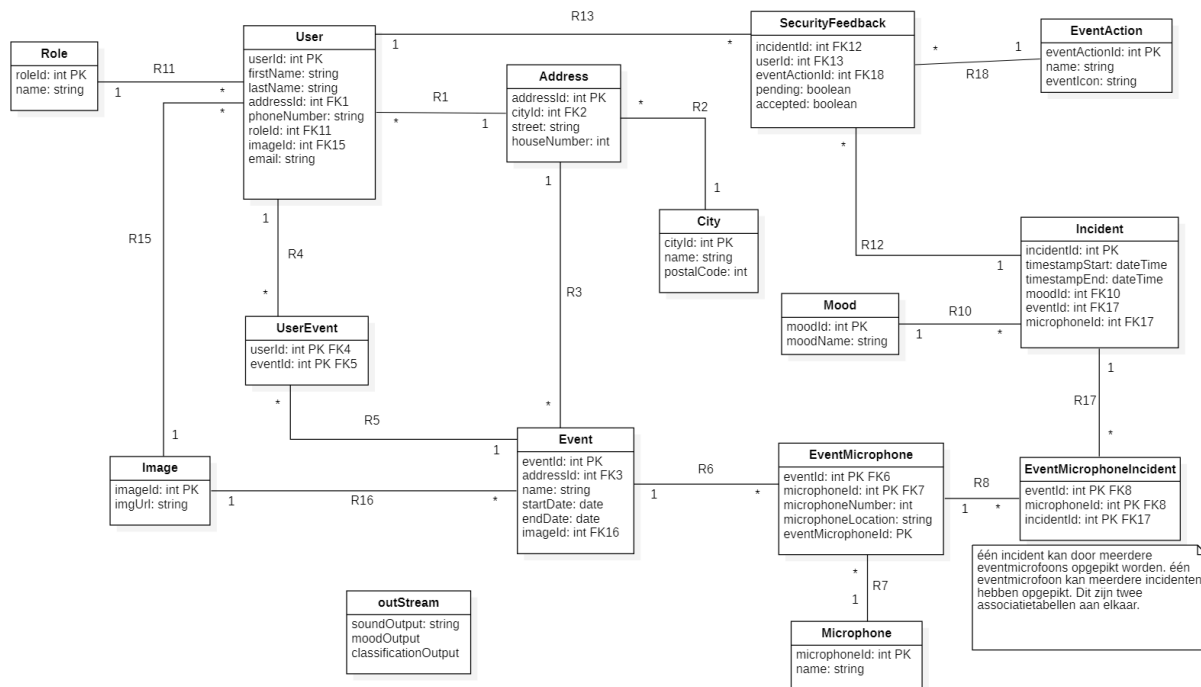
Voor dit project zal er een webapplicatie ontwikkeld worden. De teamleads zullen op een centraal punt geplaatst worden op het evenement met een computer. Deze kan dan overzichten zien en meldingen als er iets misloopt. Ook kan hij dan onmiddellijk van hieruit securitypersoneel uitsturen naar de locatie op het domein waar er een melding is gesignaleerd.

Een goed systeem begint natuurlijk met een goed ontwerp. Daarom hebben wij uitgebreid de tijd genomen om alles tot in de puntjes uit te zoeken zodanig dat als dit project effectief wordt geprogrammeerd er een goede basis aanwezig is waarop verder gewerkt kan worden. Dit onderwerp werd voornamelijk in handen genomen van onze studenten APP.

Een zeer belangrijke eis die in het achterhoofd gehouden moet worden is dat de gehele applicatie binnen een 30-tal minuten opgesteld moet kunnen worden.

4.1. Datamodel

Alles moet mooi met elkaar kunnen samenwerken. Er moeten gegevens opgeslagen worden, er moeten gegevens opgevraagd kunnen worden, ... Kortom, er is nood aan een databank. Om dit op de best mogelijke manier te kunnen opstellen is er een datamodel ontwikkeld. Dit is in feite een databank omgetoverd in visueel zichtbare tabellen zodat er een duidelijk overzicht ontstaat.



Figuur 3: Datamodel

Zoals u ziet is het een vrij groot datamodel met 15 tabellen. Hierin kan alles bijgehouden worden. Zo ziet u de verschillende personen die bijgehouden kunnen worden met hun rol (teamlead of personeel). Ook wordt het evenement en de locatie van het evenement bijgehouden.

Omdat er met audio gewerkt wordt moeten er natuurlijk microfoons in verwerkt zitten. Deze hangen dan ook weer vast aan een locatie en evenement. Zoals eerder vermeld valt de hardware van een microfoon niet onder dit project. Er wordt in dit project uitgegaan van binnenkomende audiofragmenten.

Het project draait om veiligheid. Bijgevolg moet het systeem dus meldingen kunnen tonen als er iets verkeerd loopt. Dit wordt bijgehouden onder de incidenten waarbij er ook de gemoedstoestand gezien kan worden, alsook wat voor type incident het is.

4.2. Use Case diagram

Een use case diagram is een digitale schets waarbij de werking tussen het systeem en de gebruikers visueel wordt voorgesteld aan de hand van relaties. Dit zorgt dat er een duidelijke korte voorstelling is van wat elk type gebruiker van het systeem moet kunnen.

Zoals u kan zien in onderstaande afbeelding maken er drie type gebruikers gebruik van het systeem dat wij gaan bouwen.

- Een admin: dit is de hoogste functie, de persoon die in kwestie zorgt dat het systeem operationeel kan werken.
- Een teamlead: deze staat in voor in te spelen op de meldingen die binnenkomen in het systeem om zo zijn team te kunnen aansturen.
- Securitypersoneel: deze zijn fysiek operationeel op het terrein en reageren op dispatchberichten van de teamlead.



Figuur 4: Use Case Diagram

4.3. Functionele eisen

Het use case diagram was een korte voorstelling van welke functionaliteiten elke gebruiker kan doen binnen de applicatie. Elk van deze functionaliteiten kan echter verder in detail besproken worden.

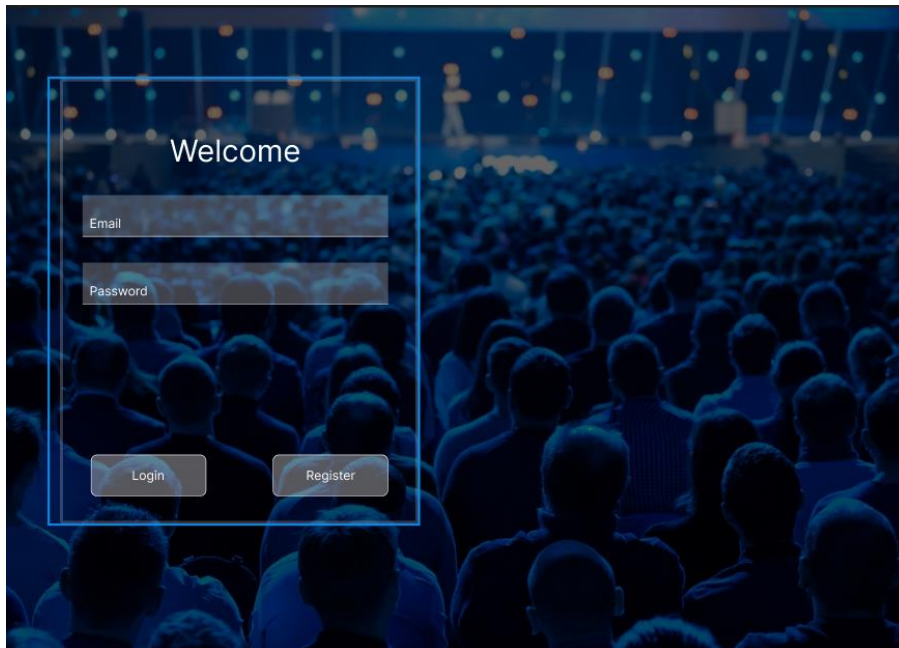
4.3.1. Use case 0: Inloggen

Functionality: Als een gebruiker van het systeem kan ik mij inloggen.

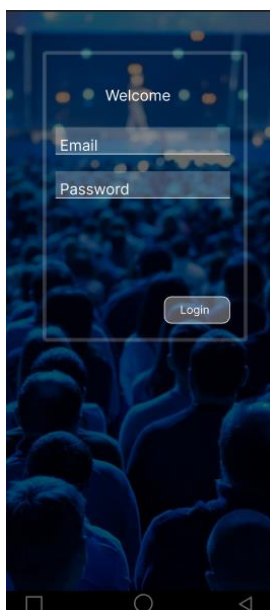
Normal flow: De actor vult zijn gegevens in. Het systeem voert de actor door naar de pagina.

Screens:

S1: Inloggen:

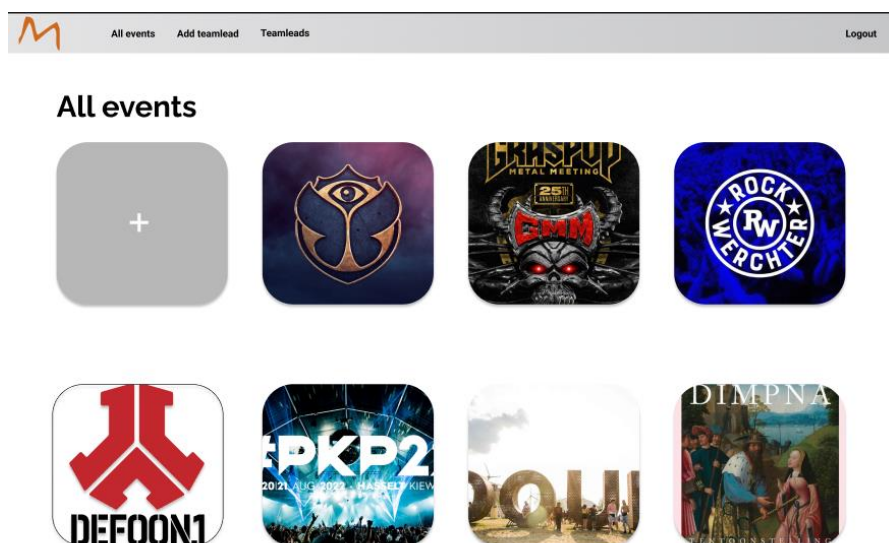


Figuur 5: inloggen PC



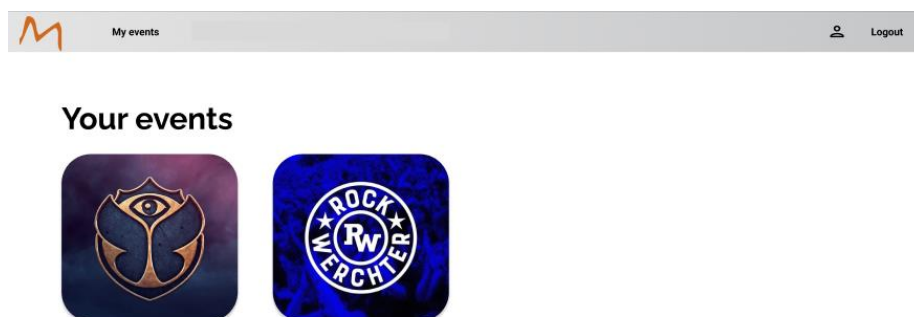
Figuur 6: Inloggen mobiel

S2: Homepage_Admin



Figuur 7: Homepage Admin

S3: Homepage_Teamlead



Figuur 8: Homepage Teamlead

4.3.2. Use case 1: uitnodigingsmail sturen naar teamleads

Functionality: Als admin kan ik een uitnodigingsmail versturen naar personen die de functie van teamlead opnemen.

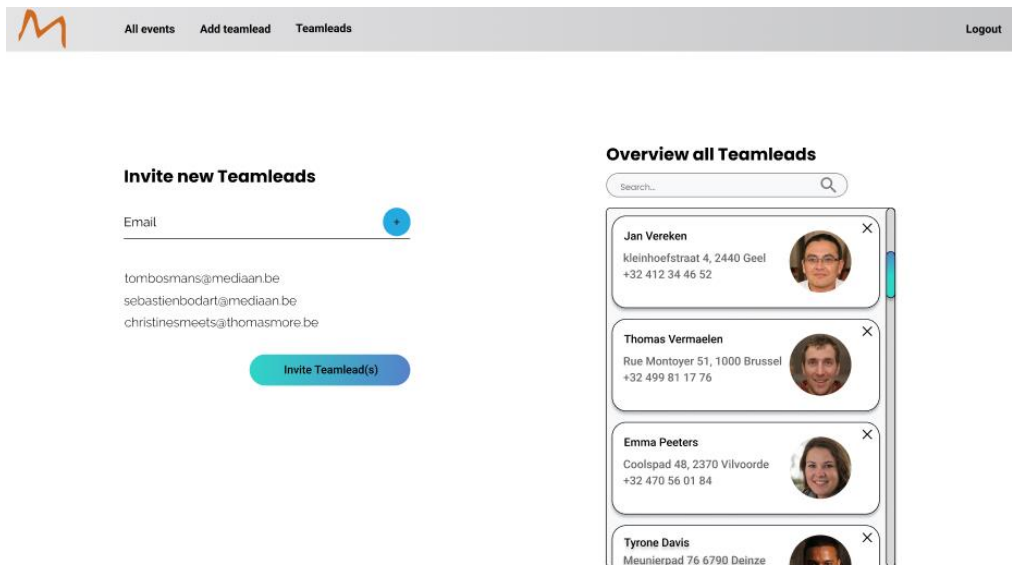
Precondition: De actor moet ingelogd zijn in het systeem.

Normal flow: De actor vult een e-mailadres in. Het systeem toont een bevestiging en stuurt een mail naar de mailbox van de ontvanger.

Alternative:

- Actor wil het e-mailsjabloon wijzigen van de uitnodigingsmail: De actor geeft aan dat hij het sjabloon wil wijzigen. Het systeem opent het huidige sjabloon. De actor voert de nodige aanpassingen uit en bevestigt.

Screens:



The screenshot displays two side-by-side web application screens. The top navigation bar includes a logo 'M', links for 'All events', 'Add teamlead', 'Teamleads', and a 'Logout' button.

The left screen, titled 'Invite new Teamleads', features an 'Email' input field with a blue '+' icon on the right. Below the input field, three email addresses are listed: 'tombosmans@mediaan.be', 'sebastienbodart@mediaan.be', and 'christinesmeets@thomasmore.be'. A blue button labeled 'Invite Teamlead(s)' is positioned at the bottom.

The right screen, titled 'Overview all Teamleads', has a search bar at the top. It displays a list of four team leads, each with a profile picture, name, address, and phone number, and a close 'X' button:

- Jan Vereken**: kleinhoefstraat 4, 2440 Geel, +32 412 34 46 52
- Thomas Vermaelen**: Rue Montoyer 51, 1000 Brussel, +32 499 81 17 76
- Emma Peeters**: Coolspad 48, 2370 Vilvoorde, +32 470 56 01 84
- Tyrone Davis**: Meunierpad 76 6790 Deinze

Figuur 9: Uitnodigingsmail sturen naar teamleads

4.3.3. Use case 2: Evenement aanmaken & beheren

Functionality: Als admin kan ik een evenement aanmaken & beheren.

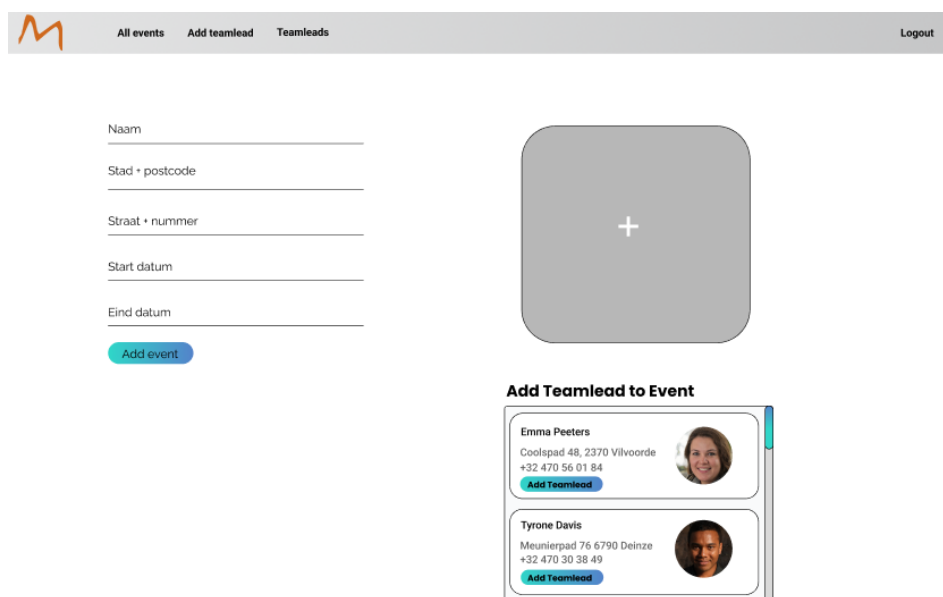
Precondition: De actor moet ingelogd zijn in het systeem.

Normal flow: De actor geeft de naam in van een evenement en bevestigt. Het systeem toont een succesbericht.

Alternative:

- Locatie toevoegen: De actor geeft een adres in. Het systeem toont een succesbericht.
- Datum toevoegen: De actor geeft één of meerdere datum(s) in. Het systeem toont een succesbericht.
- Evenement verwijderen: De actor kiest ervoor om het evenement te verwijderen. Het systeem vraagt bevestiging. De actor bevestigt. Het systeem toont een bericht dat aangeeft dat het evenement is verwijderd.

Screens:



The screenshot shows the 'Add event' form in the Mood Secure system. The form is located on the left side of the page, under the 'Add teamlead' tab. It contains several input fields: 'Naam', 'Stad + postcode', 'Straat + nummer', 'Start datum', and 'Eind datum'. Below these fields is a blue 'Add event' button. To the right of the form is a large grey square with a white plus sign, representing a placeholder for an event image. Below this is a section titled 'Add Teamlead to Event' which displays a list of team leads. The first team lead is Emma Peeters, with contact information 'Coolspad 48, 2370 Vilvoorde' and '+32 470 56 01 84'. The second team lead is Tyrone Davis, with contact information 'Meunierpad 76 6790 Deinze' and '+32 470 30 38 49'. Each team lead entry includes a small profile picture and a blue 'Add Teamlead' button.

Figuur 10: Evenement aanmaken & beheren

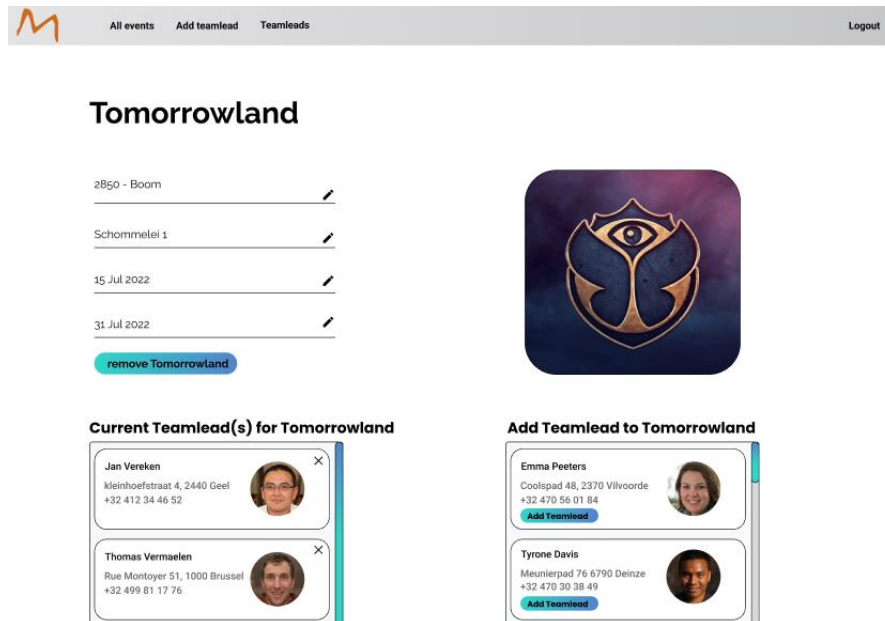
4.3.4. Use case 3: Teamlead koppelen aan een evenement

Functionality: Als admin kan ik een teamlead koppelen aan een evenement

Precondition: De actor moet ingelogd zijn in het systeem

Normal flow: Het systeem toont een lijst met mogelijke evenementen. De actor kiest een evenement en bevestigt. Het systeem toont vervolgens een lijst met alle teamleads. De actor kiest één of meerdere teamleads en bevestigt. Het systeem toont een bevestiging en stuurt een mail naar de teamlead(s).

Screens:



Tomorrowland

2850 - Boom

Schommel 1

15 Jul 2022

31 Jul 2022

remove Tomorrowland

Current Teamlead(s) for Tomorrowland

Jan Vereken
Kleinhoefstraat 4, 2440 Geel
+32 412 34 46 52

Thomas Vermaelen
Rue Montoyer 51, 1000 Brussel
+32 499 81 17 76

Add Teamlead to Tomorrowland

Emma Peeters
Coolspad 48, 2370 Vilvoorde
+32 470 56 01 84

Tyrone Davis
Meunierpad 76 6790 Deinze
+32 470 30 38 49

Figuur 11: Teamlead koppelen aan een evenement

4.3.5. Use case 4: Account aanmaken & beheren

Functionality: Als teamlead kan ik een account aanmaken & beheren.

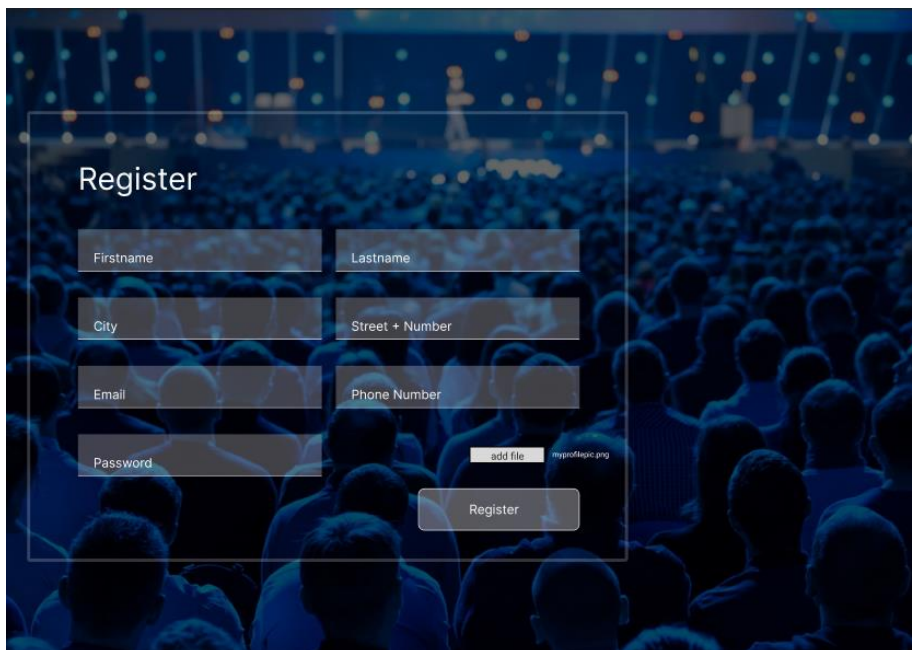
Precondition: De actor moet op de link in de registratiemail geklikt hebben.

Normal flow: De actor vult zijn e-mailadres en wachtwoord in alsook zijn persoonlijke gegevens en bevestigt. Het systeem toont een bevestiging.

Alternative:

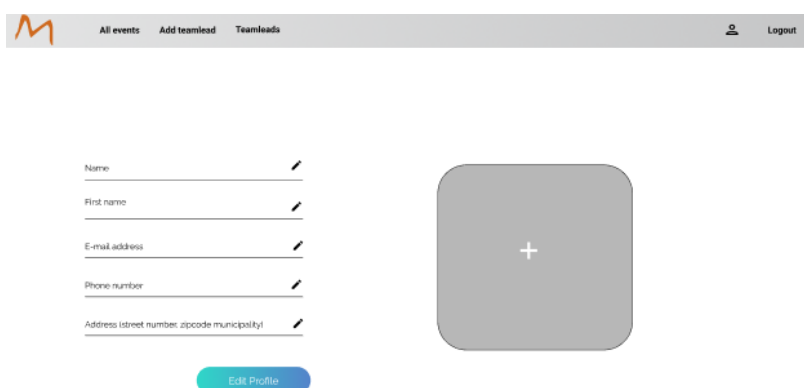
- E-mailadres al in gebruik: Het systeem toont een melding dat het e-mailadres al in gebruik is en verwijst de actor door om in te loggen.
- Gegevens aanpassen: De actor past zijn gegevens aan. Het systeem toont een bevestiging.

Screens:



The screenshot shows a 'Register' form overlaid on a background image of a crowd at night. The form contains the following fields: Firstname, Lastname, City, Street + Number, Email, Phone Number, and Password. There is also an 'add file' button with a file named 'myprofilepic.png' and a 'Register' button at the bottom.

Figuur 12: Account aanmaken



The screenshot shows the 'Account beheren' (Manage Account) form. It includes a header with a logo and navigation links: 'All events', 'Add teamlead', and 'Teamleads'. On the right, there are 'Logout' and 'Profile' links. The main form area contains input fields for 'Name', 'First name', 'E-mail address', 'Phone number', and 'Address (street number, zipcode municipality)'. Each field has an edit icon (pencil) to its right. To the right of the form is a large grey square with a white plus sign, likely for adding a profile picture. At the bottom, there is a blue 'Edit Profile' button.

Figuur 13: Account beheren

4.3.6. Use case 5: Applicatie opstarten/uitzetten

Functionality: Als teamlead kan ik de applicatie opstarten.

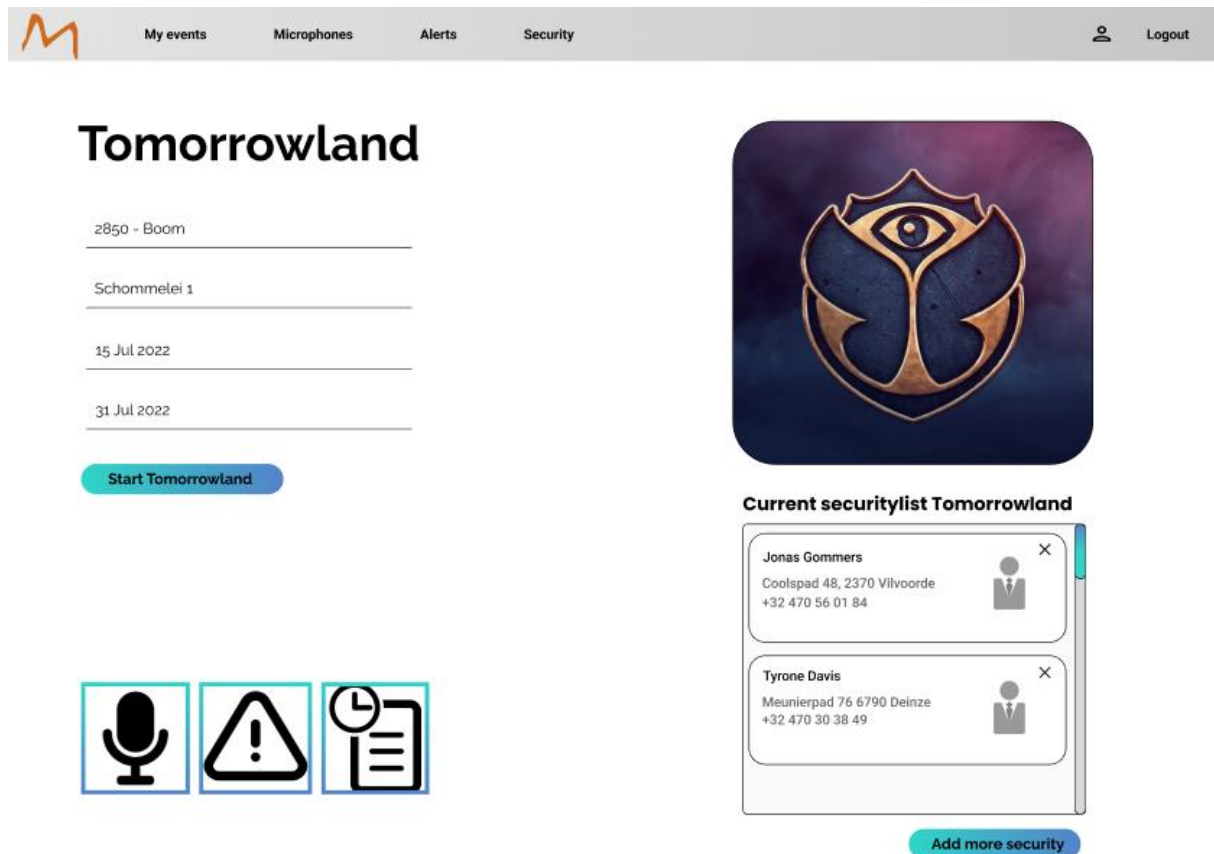
Precondition: De actor moet ingelogd zijn in het systeem.

Normal flow: Het systeem toont een melding dat de applicatie wordt opgestart en stuurt een opstartprocedure naar de applicatie.

Alternative:

- De applicatie uitzetten: Het systeem toont een melding dat de applicatie wordt uitgezet en stuurt een uitzetprocedure naar de applicatie.

Screens:



Figuur 14: Applicatie opstarten/uitzetten

4.3.7. Use case 6: Uitnodigingsmail sturen naar securitypersoneel

Functionality: Als teamlead kan ik een uitnodigingsmail sturen naar securitypersoneel.

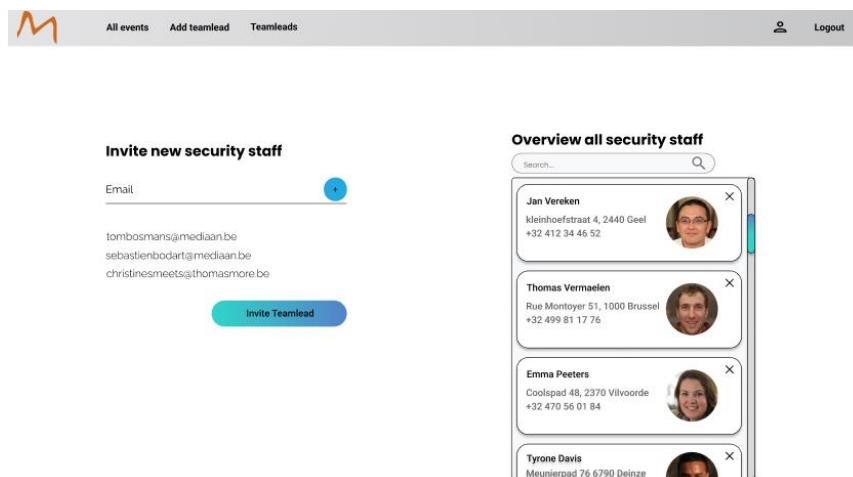
Precondition: De actor moet ingelogd zijn

Normal flow: De actor vult één of meerdere e-mailadressen in. Het systeem toont een bevestiging en stuurt een mail naar de mailbox van de ontvanger.

Alternative:

- Actor wil het e-mailsjabloon wijzigen van de uitnodigingsmail: De actor geeft aan dat hij het sjabloon wil wijzigen. Het systeem opent het huidige sjabloon. De actor voert de nodige aanpassingen uit en bevestigt.

Screens:



Figuur 15: Uitnodigingsmail sturen naar securitypersoneel

4.3.8. Use case 7: Security toewijzen aan een evenement

Functionality: Als teamlead kan ik securitypersoneel toewijzen aan een evenement

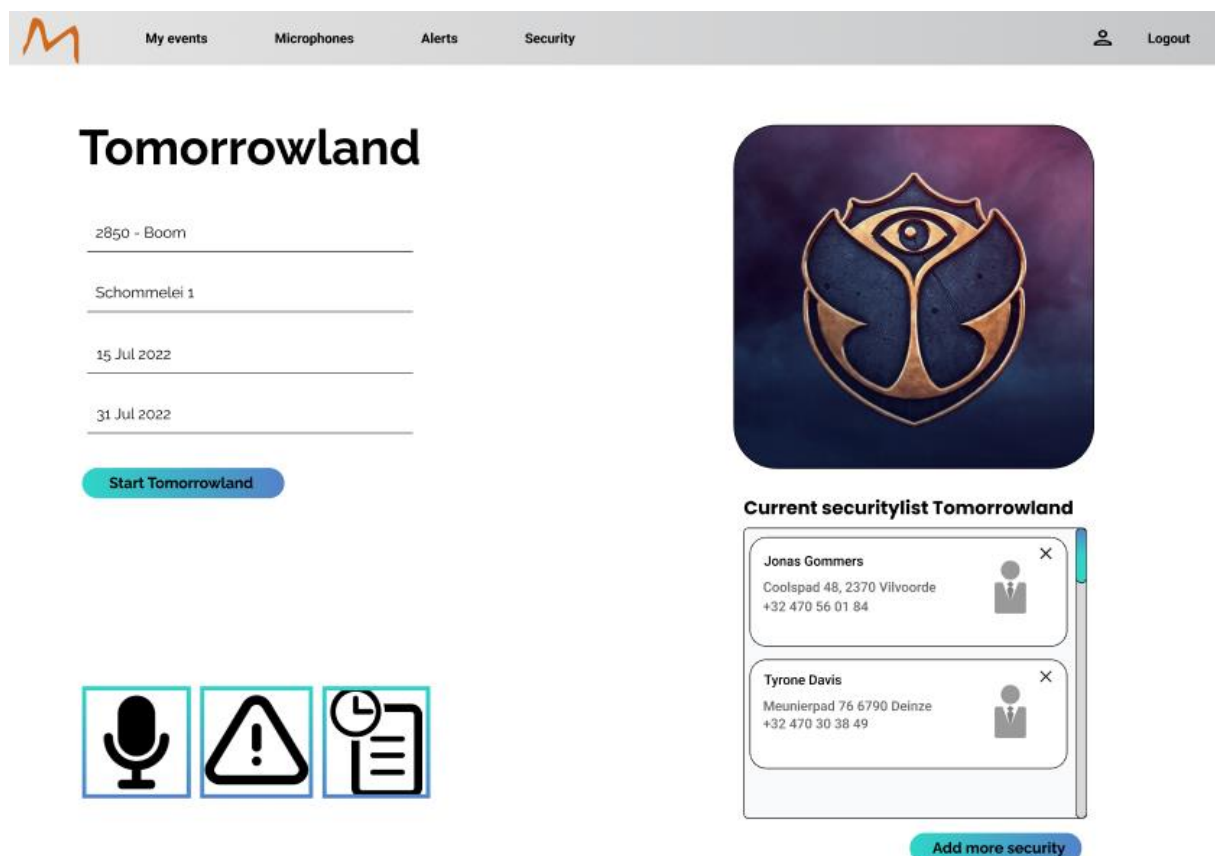
Precondition: De actor moet ingelogd zijn in het systeem

Normal flow: Het systeem toont een lijst met alle mogelijke evenementen. De actor kiest een evenement en bevestigt. Het systeem toont vervolgens een lijst met al het securitypersoneel dat komt werken op dat evenement. De actor bevestigt deze lijst. Het systeem toont een bevestiging. Het systeem stuurt een mail naar het personeel met het evenement en de nodige informatie.

Alternative:

- Security niet laten werken: De actor vinkt de naam aan van de security die niet aanwezig zal zijn op het evenement. Het systeem toont een bevestiging en verwijdert de securitypersoon uit de lijst.
- Extra security toevoegen: De actor geeft aan extra personeel te willen. Het systeem toont een lijst met personeel dat nog niet is toegewezen aan het evenement en beschikbaar is. De actor kiest één of meerdere personen en confirmeert. Het systeem voegt deze personen toe aan de lijst met securitypersoneel dat aanwezig is.

Screens:



Figuur 16: Security toewijzen aan een evenement

4.3.9. Use case 8: Locaties toewijzen aan een evenement

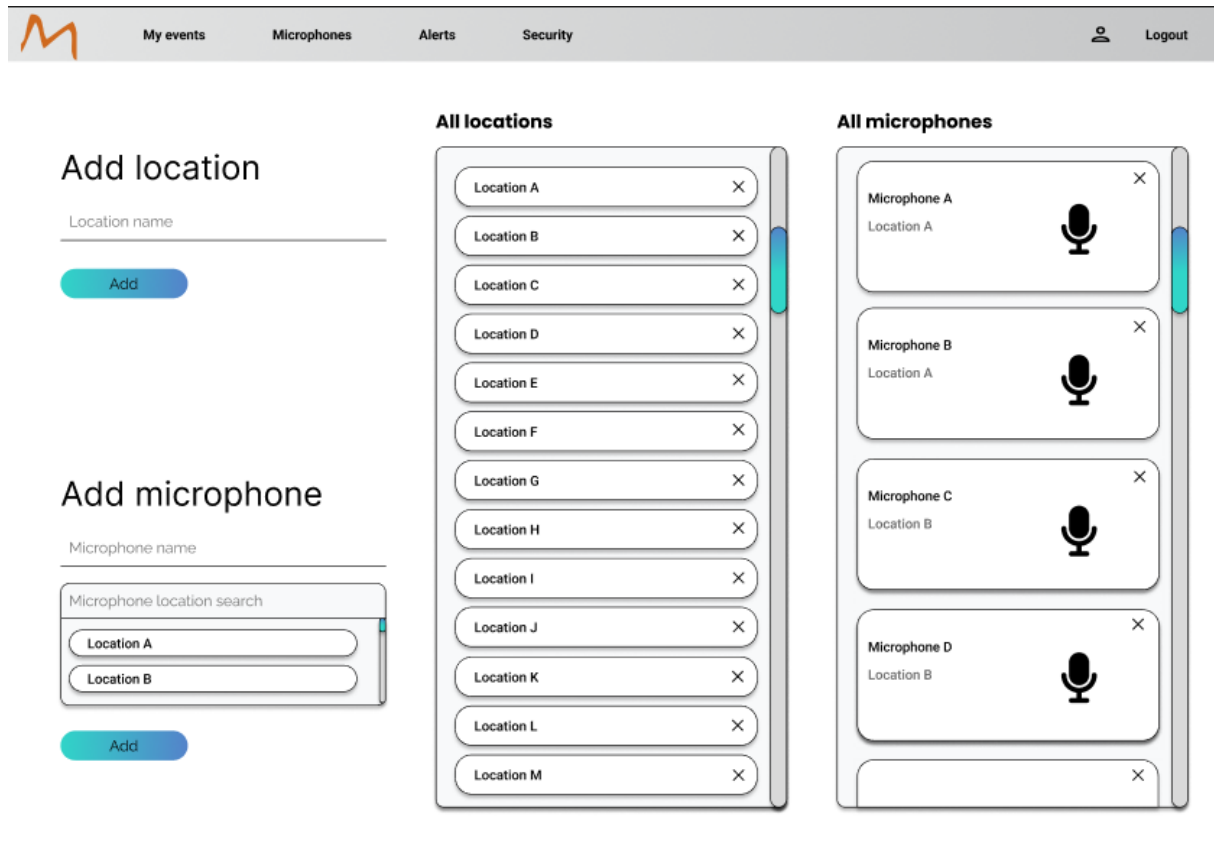
Functionality: Als teamlead kan ik locaties toewijzen aan een evenement

Precondition: De actor moet ingelogd zijn in het systeem

Normal flow: Het systeem toont een lijst met alle mogelijke evenementen. De actor kiest een evenement en bevestigt. De actor kan met behulp van een Form één of meerdere locaties toevoegen aan het gekozen evenement. De actor bevestigt zijn locaties. Het systeem toont een bevestiging.

Alternative:

- Een locatie verwijderen: De actor geeft aan een locatie niet langer meer te willen gebruiken. Het systeem verwijdert deze locatie uit de lijst.
- Een locatie van naam veranderen: De actor geeft aan welke locatie hij van naam wil veranderen, voert vervolgens zijn naamsverandering uit en bevestigt.
- Aantal security voor een locatie toevoegen: De actor geeft aan welke locatie hij wil aanpassen. De actor voert vervolgens in hoeveel securitypersoneel hij aan deze locatie wil koppelen. Het systeem geeft een bevestiging.



The screenshot displays the Mood Secure web application interface. At the top, there is a navigation bar with a logo 'M' and tabs for 'My events', 'Microphones', 'Alerts', and 'Security'. A 'Logout' button is visible on the right. The main content area is divided into two columns. The left column contains two forms: 'Add location' and 'Add microphone'. The 'Add location' form has a text input for 'Location name' and an 'Add' button. The 'Add microphone' form has a text input for 'Microphone name', a dropdown for 'Microphone location search' showing 'Location A' and 'Location B', and an 'Add' button. The right column contains two lists: 'All locations' and 'All microphones'. The 'All locations' list shows 13 items, 'Location A' through 'Location M', each with a delete icon. The 'All microphones' list shows 4 items, 'Microphone A' through 'Microphone D', each with a delete icon and a microphone icon. Microphone A is assigned to Location A, Microphone B to Location A, Microphone C to Location B, and Microphone D to Location B.

Figuur 17: Locaties toewijzen aan een evenement

4.3.10. Use case 9: Microfoons toewijzen aan een evenement

Functionality: Als teamlead kan ik microfoons toewijzen aan een evenement.

Precondition: De actor moet ingelogd zijn in het systeem.

Normal flow: Het systeem toont een overzicht van alle evenementen. De actor duidt een evenement aan. Het systeem toont een overzicht van alle microfoons die opgenomen zijn in het systeem. De actor duidt aan welke microfoons hij wil. Het systeem toont een bevestiging.

Alternative:

- Een microfoon verwijderen uit de lijst: De actor duidt aan welke microfoon hij definitief uit het systeem wil halen. Het systeem vraagt een bevestiging. De actor bevestigt.
- Een nieuwe microfoon toevoegen aan de lijst: De actor duidt aan dat hij een nieuwe microfoon wil toevoegen aan de lijst. De actor vult de gegevens in van de microfoon. Het systeem toont vervolgens een bevestiging dat de microfoon is toegevoegd aan het systeem.

Screens: idem use case 8

4.3.11. Use case 10: Microfoons toewijzen aan een locatie op een evenement

Functionality: Als teamlead kan ik microfoons toewijzen aan een locatie op een evenement

Precondition: De actor zit in het evenementenscherf.

Normal flow: Het systeem toont een overzicht van alle microfoons die toegekend zijn aan het evenement. De actor kiest welke microfoon hij aan welke locatie wil koppelen en bevestigt. Het systeem toont een bevestiging.

Alternative:

- Een microfoon van zijn locatie ontdoen: De actor duidt aan welke microfoon hij niet meer aan een locatie wil toe-eigenen.

Screens: idem use case 8

4.3.12. Use case 11: Dashboards bekijken

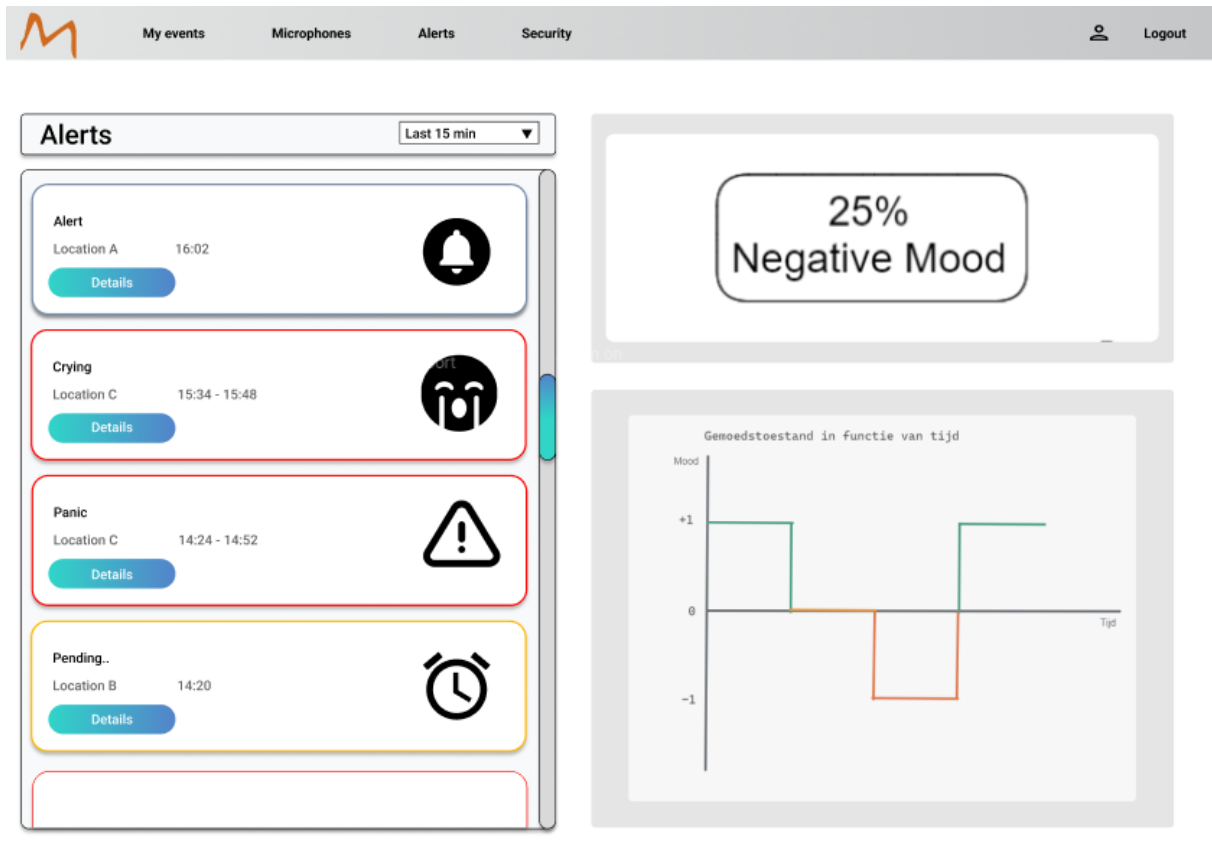
Functionality: Als teamlead kan ik dashboards bekijken.

Precondition: De actor moet ingelogd zijn in het systeem.

Normal flow: Het systeem toont een overzicht van alle binnenkomende data met behulp van grafieken & charts.

Alternative:

- De actor wil een meer gedetailleerde view van een grafiek: De actor klikt op de grafiek/chart die hij in detail wil bekijken. Het systeem vergroot deze grafiek/chart en toont gegevens over een langere periode.



Figuur 18: Dashboards bekijken

4.3.13. Use case 12: Alerts bekijken

Functionality: Als teamlead kan ik alerts bekijken

Precondition: Er moet een binnenkomende alert zijn geweest.

Normal flow: Het systeem toont een melding met benoeming van ernst en locatie als er een alert binnenkomt.

Alternative:

- Een alert in detail bekijken: De actor klikt op de alert die hij in detail wil bekijken. Het systeem toont extra informatie (locatie-ernst-duur-soort alert-...) over deze alert.
- Een alert bewerken: De actor klikt op de alert die hij wil bewerken. Het systeem toont de extra informatie over deze alert. De actor klikt op de informatie die hij wil aanpassen en voert zijn aanpassingen uit. Het systeem bevestigt.

Screens: idem use case 11

4.3.14. Use case 13: Personeel uitsuren naar een locatie

Functionality: Als teamlead kan ik securitypersoneel uitsuren naar een specifieke locatie.

Precondition: Er moet een binnenkomende alert zijn geweest.

Normal flow: Het systeem toont de details van de alert. De actor kan kiezen welk personeel hij effectief uitsuurt. Het systeem bevestigt. Het systeem stuurt een melding naar het personeel.

Screens:



Alert in location A



Figuur 19: Personeel uitsuren naar een locatie

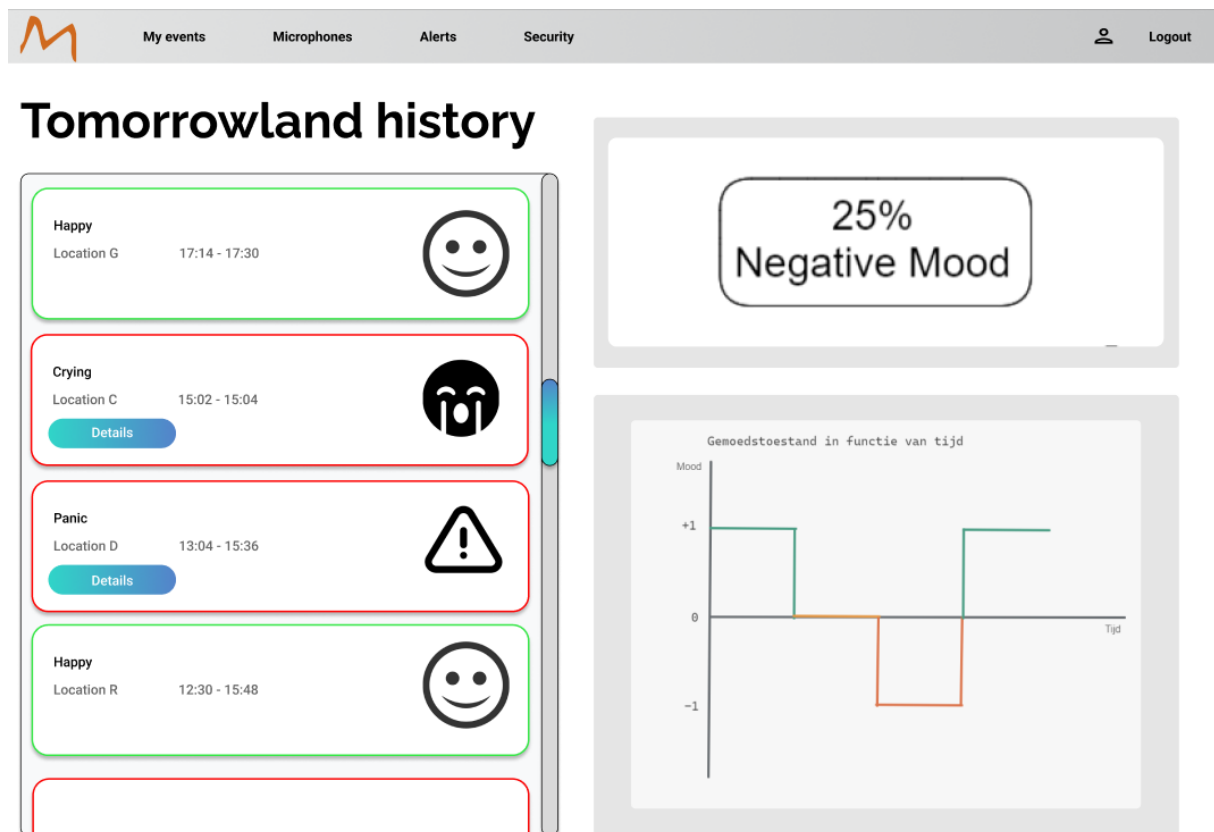
4.3.15. Use case 14: Geschiedenis opvragen

Functionality: Als teamlead kan ik de geschiedenis van alerts opvragen.

Precondition: Er moeten binnenkomende alerts zijn geweest

Normal flow: Het systeem toont een overzicht van alle alerts die er op dat evenement hebben plaatsgevonden.

Screens:



Figuur 20: Geschiedenis opvragen

4.3.16. Use case 14: Securitypersoneel kan zich registreren

Functionality: Als security kan ik mij registreren om als security te werken.

Precondition: De actor moet op de link in de uitnodigingsmail geklikt hebben.

Normal flow: Het systeem toont een Form. De actor vult zijn e-mailadres, naam en extra gegevens in. De actor bevestigt zijn gegevens. Het systeem toont een confirmatie.

Alternative:

- De link is expired: Het systeem verwijst de actor door naar een pagina waarop staat dat de link niet tijdig gebruikt is.
- De link is al gebruikt: Het systeem verwijst de actor door naar een pagina waarop staat dat de link al is gebruikt.

Screens: idem use case 4

4.3.17. Use case 15: confirmeren om naar een locatie te gaan

Functionality: Als security kan ik confirmeren om mezelf naar een locatie te begeven.

Precondition: Er is een melding binnengekomen van de teamlead opdat de actor zich naar een locatie begeeft.

Normal flow: De actor klikt op de melding. Het systeem toont een knop waarmee de actor kan confirmeren dat hij zich naar de locatie zal begeven. De actor drukt op de knop. Het systeem stuurt een melding naar de teamlead met deze bevestiging.

Alternative:

- De actor weigert het verzoek: Het systeem stuurt een melding naar de teamlead waarin staat dat de actor zich niet kan verplaatsen naar de locatie.

Screens:



Figuur 21: Confirmeren om naar een locatie te gaan

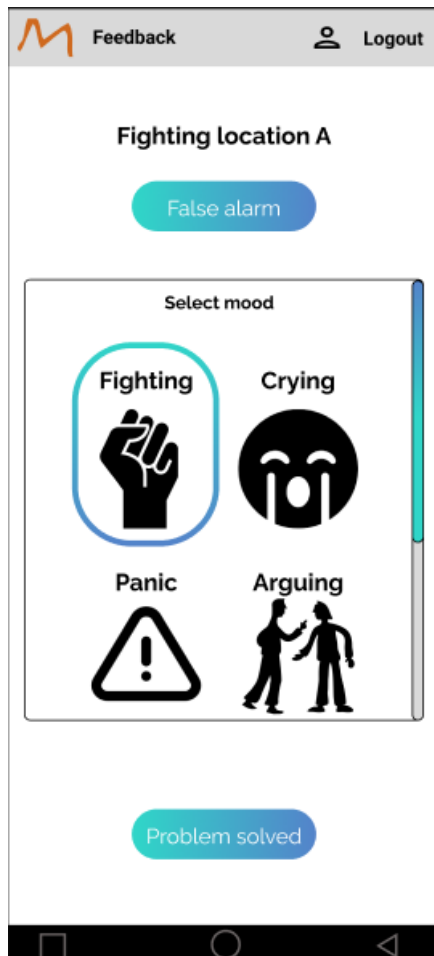
4.3.18. Use case 16: Feedback geven op locatie

Functionality: Als security kan ik feedback geven over de situatie op locatie.

Precondition: De actor moet geklikt hebben op de alert waar hij feedback op wil geven & waarvoor hij opgeroepen is.

Normal flow: De actor geeft aan feedback te willen geven. Het systeem toont enkele opties. De actor kiest één van de opties en confirmeert. Het systeem bevestigt.

Screens:



Figuur 22: Feedback geven op locatie

4.4. Technische keuzes

Bij een webapplicatie die gebouwd moet worden komen er allerlei technische keuzes aan te pas, welke tool wordt gebruikt om de front-end op te bouwen? Welke tool wordt gebruikt voor de back-end? ... ? Deze worden in dit hoofdstuk toegelicht.

4.4.1. Front-end keuze - Vue

De teamleads die de webapplicatie zullen gebruiken moeten worden voorzien van een website die gemakkelijk in gebruik is maar die er tevens ook goed uit ziet. Hij zal namelijk gedurende heel zijn werkperiode deze webapplicatie moeten gebruiken.

Omwille van deze reden hebben wij gekozen om Vue te gebruiken als onze front-end taal. Vue is namelijk een perfect framework voor het bouwen van gebruikersinterfaces en single-page applications. Vue bouwt verder op de standaard HTML, CSS en Javascript en biedt een declaratief en op componenten gebaseerd programmeermodel dat u helpt bij het efficiënt ontwikkelen van eenvoudige of complexe gebruikersinterfaces.

Vue is ook een framework dat in populariteit enorm aan het groeien is. Dit project vormt dan ook de ultieme gelegenheid om onszelf te verdiepen in deze taal om sterker in het werkveld te kunnen stappen.

Een derde reden waarom wij Vue verkiezen is omdat Mediaan hieraan de voorkeur geeft.

4.4.2. Back-end keuze - .NET

De back-end maakt het mogelijk om onze webapplicatie te laten communiceren met de database. Dit is van essentieel belang om een functioneel werkende applicatie te bekomen.

Als back-end heeft Mood Secure gekozen voor .NET, dit is een programmeertaal die ontstaan is vanuit Microsoft en die helpt om snel toepassingen van hoge kwaliteit te ontwikkelen. In combinatie met de uitgebreide klassenbibliotheken, gemeenschappelijke API's, meertalige ondersteuning en de krachtige tooling van de Visual Studio-familie is .NET het meest productieve platform voor ontwikkelaars.

.NET is tevens ook de voorkeurstaal van Mediaan, dit heeft ook een grote impact gehad waarom wij juist voor deze back-end hebben gekozen.

4.5. Conclusie

Na grondig onderzoek zijn er een zeventien functionaliteiten die geïmplementeerd moeten worden als 'must have'. Deze worden uitgewerkt met behulp van Vue.JS, .NET en MySQL zoals onderzocht in dit hoofdstuk. We gaan gebruik maken van ons datamodel om onze back-end te maken om in contact te komen met onze databank. Onze prototypes gaan gebruikt worden voor het design van de front-end van onze applicatie. Op deze manier werkt het systeem goed samen met het AI-algoritme en de security die later besproken worden in dit document.

5. Server- en netwerkontwerp

De front-end en back-end talen staan vast. Echter omvat een volledige webapplicatie veel meer dan dat. Zo moet er nog een DBMS en een manier om aan hosting te doen gekozen worden en moet de beveiliging van de applicatie zelf in rekening gebracht worden. Al deze zaken vallen onder het server- en netwerkontwerp.

5.1. DBMS-keuze – MySQL

In de front-end wordt er gewerkt met Vue. Deze communiceert met de back-end, namelijk .NET. Nu is er nog een managementsysteem nodig om een databank te kunnen bouwen, op te roepen en te kunnen onderhouden.

Hier was de keuze niet zo vanzelfsprekend. Om een definitief besluit hieruit te kunnen trekken hebben we een WRM of Weighted Ranking Method uitgevoerd op verschillende databasemanagementsystemen. WRM is een techniek om op objectieve wijze een keuze te maken tussen een aantal gelijksoortige oplossingen. Er wordt een lijst van criteria gebruikt, die worden gewogen op basis van het belang dat de ontwikkelaars of organisatie aan het criterium hechten.

Enkele puntjes vonden wij belangrijk dat een databasemanagementsysteem moet hebben:

- Flexibiliteit: het vermogen om zich aan te passen aan de noden.
- Schaalbaarheid: de mogelijkheid om een configuratie eenvoudig groter te maken.
- Eenvoud: dat het niet te moeilijk is om te bouwen of te onderhouden.
- Prestatie: hoe goed het presteert ten opzichte van de verwachtingen.
- Compatibiliteit: hoe goed het kan samenwerken met onze technische keuzes.

Deze puntjes hebben wij dan samen met enkele databasemanagementsystemen in een tabel gegoten om tot een definitief besluit te komen.

Tabel 1: WRM Databases

Criteria	Factor	MySQL	MariaDB	MongoDB	PostgreSQL	SQLite
Flexibiliteit	2	100%	100%	100%	90%	90%
Schaalbaarheid	2	80%	90%	100%	80%	40%
Eenvoud	1	100%	70%	80%	70%	100%
Prestaties	3	90%	70%	80%	70%	60%
Compatibiliteit	2	100%	90%	90%	80%	90%
Totaal	10	9.3	8.4	9	7.8	7.2
Totaal %	100%	93%	84%	90%	78%	72%

Zoals u kan zien scoort MySQL het beste in de WRM. Daarom zullen wij dus ook MySQL gebruiken om dit project uit te werken.

5.2. Storage

De binnenkomende audiodata moet tijdelijk bewaard worden zodat het AI-algoritme zijn conclusies hieruit kan trekken. Deze conclusies en eventueel mogelijk de acties die hieruit volgen met bijhorende audio, moeten worden bewaard voor een periode van twee jaar.

Er is dus nood aan een tijdelijk audiostorage. De “Azure web app service” bevat zelf geheugen dat we hiervoor kunnen gebruiken. Een permanente databank is nodig om de conclusies en acties in bij te houden. Deze zou in een ideaal scenario automatisch data moeten verwijderen die de tijdsperiode van twee jaar heeft bereikt.

Om het systeem zo veilig mogelijk te houden is het best practice om de bewaarde data los te koppelen van de app. Deze mag dus niet in dezelfde omgeving draaien.

5.3. Automatisatie

Mediaan wil deze toepassing kunnen aanbieden als dienst naar klanten toe. Daarom is het aangeraden een script te schrijven (Biceps) voor het opzetten van de omgeving. Op deze manier moet dit dan niet manueel voor elke klant gebeuren en blijft het kostenplaatje lager. Eventuele wijzigingen in de omgeving zijn hierdoor dan ook makkelijk door te voeren.

Een vereiste is dat de omgeving binnen een dertigtal minuten volledig operationeel is. Om de kosten te drukken verwacht Mediaan dat de toepassing enkel beschikbaar is tijdens het event.

Voor extra veiligheid willen wij de toegang tot de Azure omgevingen beperken. Daarom willen wij een script schrijven dat de omgeving met één enkele druk op een knop kan starten. Dit zal mee geïmplementeerd worden in de webapplicatie onder de vorm van een knop op de pagina.

5.4. Back-up

Omdat wij willen vermijden dat er gegevens verloren gaan, wordt er een back-up voorzien van de databank. Dit is dus een back-up van de conclusies die het AI-algoritme heeft gemaakt. Op vraag van Mediaan behouden we ook de audio van de data die bepaalde alerts triggeren voor verdere evaluatie. We maken de database 'Locally redundant', dit betekent dat we deze enkel op één plaats bewaren. Dit is normaal gezien genoeg, behalve bij natuurrampen. Op deze manier beperken we de kosten.

Voor de applicatie zelf is dit geen must. Wij hebben namelijk een automatisatiescript dat alles opzet. In dit script komt dan simpelweg te staan dat het aanmaken van een nieuwe databank enkel wordt uitgevoerd mits er nog geen aanwezig is.

5.5. Azure componenten vergelijkingstabellen

De Azure omgeving die gebruikt zal worden, moet voorzien worden van een methode om de applicatie te kunnen laten lopen. Hier valt de keuze tussen een VM, een container instance of een Azure Web app service. Wederom gebruiken wij een WRM om tot een conclusie te komen.

5.5.1. App omgeving

Tabel 2: WRM Azure omgeving

Methode	VM	Container instance	Azure Web App
Beschikbaarheid	Ja	Niet in abonnement	Ja
Compatibel met Flask	Ja		Ja
Gebruiksvriendelijkheid	Zelf alles configureren		App beschikbaar via internet
Updates	Manueel		Via GitHub mogelijk
Kostprijs/ maand met memory 3.5 GB	€42.63 (0.0573/uur)		€22.16 (N.v.t./uur)

De VM heeft zowel een kostprijs per uur als per maand. Dit is voor het kostenplaatje en de redenering dat de toepassing enkel moet draaien tijdens evenementen interessant. Echter heeft de Azure Web app meer voordelen op vlak van gebruiksvriendelijkheid en het toepassen van eventuele updates.

Het bedrag dat u misschien zou uitsparen met het abonnement per uur gaat u dan weer nodig hebben voor de configuratie en updates. Werkuren kosten meer dan het verschil tussen deze twee prijzen. Hierdoor raden wij de Azure Web app aan om het project te laten lopen.

Azure web app service heeft zelf geheugen voor de tijdelijke data. De AI-toepassing heeft vooral nood aan CPU-kracht. Het lijkt ons daarom verstandig om meerdere cores te voorzien. In onderstaande foto's kan u twee scenario's vinden waar de web app service tien GB bevat. Dit is normaal gezien voldoende voor de tijdelijke audio. In de eerste foto is de prijs zichtbaar voor twee cores, in de volgende foto voor vier cores.

Tabel 3: Kostprijs per maand en uur voor 2 en 4 cores

/maand	/uur
<div> <div> App Service Basic Tier; 1 B2 (2 Core(s), 3.5 GB RAM, 10 GB Storage... Upfront: €0.00 Monthly: €105.31 </div> <div> <div>App Service</div> <div> Region: West Europe Operating system: Windows Tier: Basic </div> <div> Basic </div> <div> <div> <div>INSTANCE:</div> <div>B2: 2 Cores(s), 3.5 GB RAM, 10 GB Storage, €0.144</div> </div> <div> <div>1</div> <div>×</div> <div>1</div> <div>Month</div> </div> <div> <div>Instances</div> </div> </div> <div> <div>SSL Connections</div> </div> <div> <div>Upfront cost</div> <div>€0.00</div> </div> <div> <div>Monthly cost</div> <div>€105.31</div> </div> </div> </div>	€0.14
<div> <div> App Service Basic Tier; 1 B3 (4 Core(s), 7 GB RAM, 10 GB Storage... Upfront: €0.00 Monthly: €210.63 </div> <div> <div>App Service</div> <div> Region: West Europe Operating system: Windows Tier: Basic </div> <div> Basic </div> <div> <div> <div>INSTANCE:</div> <div>B3: 4 Cores(s), 7 GB RAM, 10 GB Storage, €0.289</div> </div> <div> <div>1</div> <div>×</div> <div>1</div> <div>Month</div> </div> <div> <div>Instances</div> </div> </div> <div> <div>SSL Connections</div> </div> <div> <div>Upfront cost</div> <div>€0.00</div> </div> <div> <div>Monthly cost</div> <div>€210.63</div> </div> </div> </div>	€0.29

Wij raden de oplossing met vier cores aan zodat de AI-toepassing optimaal kan werken. U kan zien in de foto's dat de maandelijkse schatting boven de gevraagde €200 is. De toepassing gaat gebruikt worden voor evenementen van enkele uren of dagen en wordt daarna afgezet. Door het pay-as-you-go model zal u dus enkel de periode dat de app gebruikt wordt, moeten betalen waardoor u dus zeker niet aan dit bedrag zal komen. U kan zien in de tabel ernaast dat de prijs per uur voor de twee extra cores niet veel verschilt. Dit zal echter wel een groot verschil geven in de algemene werking tussen de app en AI-toepassing.

5.5.2. Storage

Zoals eerder vermeld, moet er storage voorzien worden voor zowel kort gebruik (voor audio en video) als voor langdurig gebruik van twee jaar (gelinkt aan de databank).

Tabel 4: WRM-storage

Pay-as-you-go	1ste 50 TB/maand	€0.15004
	Volgende 450 TB/maand	€0.15004
	Meer dan 500 TB/maand	€0.15004
Reserved Capacity	100 TB/maand	€1546
	1 PB/maand	€15053

Voor beide storages raden wij een pay-as-you go betaalmodeel aan. Hier betaalt u enkel voor wat u gebruikt.

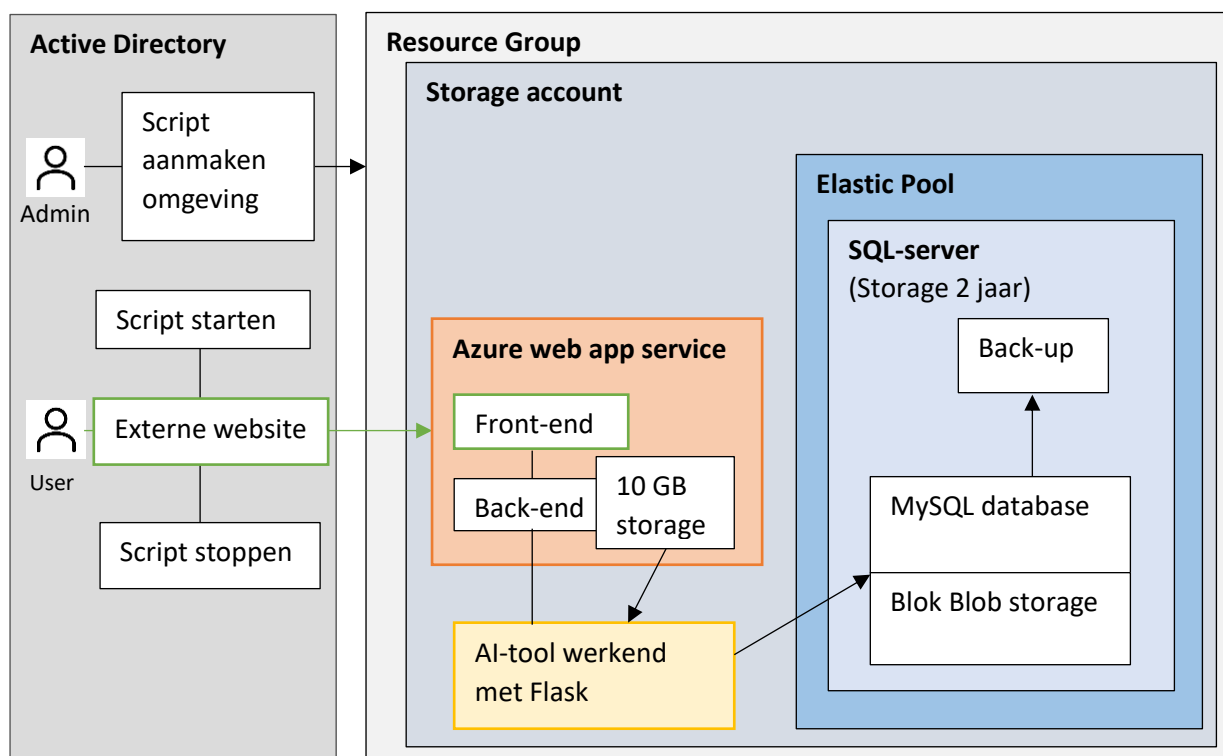
5.5.3. Database

Uit de WRM van het databank managementsysteem hebben we geconcludeerd dat MySQL het meest geschikt is. Hiervoor moeten wij eerst een SQL-databaseserver opzetten. Dit vraagt geen extra kosten.

Vervolgens moeten we Azure database voor MySQL hieraan linken. Ook moeten wij binnen de Azure-omgeving een elastic pool aanmaken waar de database gebruik van gaat maken.

5.6. Schematische voorstelling omgeving

Om de omgeving op een eenvoudige manier weer te kunnen geven hebben wij deze in een visueel component gegoten.



Figuur 23: Schematische voorstelling omgeving

5.7. Geschatte kostenplaatje

Tabel 5: Totale geschatte kostenplaatje

Azure component	1 maand	1 uur
Resource group	Gratis	
Storage account 1000 GB (1TB)	€20,82	
Azure web app service 4 cores 10GB geheugen	€210,63	€0,29
Elastic pool	€69,46	
SQL-server	Gekoppeld aan pool dus gratis	
Back-up	Gekoppeld aan SQL-server dus SQL-server	
TOTAAL	€300,91	€90,57

Bij de maandprijs kan u zien wat het kostenplaatje zou zijn zonder de toepassing af te zetten. Bij het uur kan u zien dat voor één uur de prijs op €90,57 komt. Dit lijkt veel, maar de storage account en de elastic pool zijn een vaste kost. Deze gaan dus niet mee omhoog.

Volgens onze berekening is het mogelijk om 377 uur de app te draaien tot we aan de gevraagde €200 euro zitten. Dit zijn vijftien volledige dagen van 24 uur. Er zullen misschien nog enkele registratiekosten bij komen maar zelfs met deze kosten erbij, komt u normaal gezien niet boven €200/maand.

5.8. Conclusie

Onze toepassing werkt met de active directory van de klant. Hier is het mogelijk door vanuit Mediaan de omgeving via een script op te zetten en af te breken. Op deze manier zal enkel Mediaan de toepassing kunnen aanmaken bij klanten. Hierdoor kunnen de klanten van Mediaan zelf geen extra omgevingen aanmaken die zullen zorgen voor nieuwe kosten. De verwijder-optie maakt het mogelijk voor Mediaan de omgeving te verwijderen als de klant besluit het contract stop te zetten. De admin kan via een externe website die verbonden is aan de active directory, de omgeving starten en stoppen. Dit doen we zodat externen geen toegang hebben om deze acties uit te voeren.

We hebben gekozen de app te laten draaien via Azure web app service. De web app service vraagt een resource group en storage account. De keuze voor de Azure web app service is omdat deze zeer gebruiksvriendelijk is: het is compatibel met GitHub en met Flask. Dit is zeker belangrijk omdat de AI-toepassing op Flask werkt. De AI-toepassing heeft nood aan veel CPU. Daarom hebben we ervoor gezorgd dat de Azure web app service vier cores heeft en daardoor ook tien GB aan storage. Dit zorgt ervoor dat we geen tijdelijke storage nodig hebben. Wel hebben we nog een database nodig voor de tweejaarlijkse storage. Deze is 100GB, dit lijkt ons voldoende voor de incidenten, de audio en conclusies te bewaren. Deze hangt vast aan een back-up die elke dag kan updaten. Deze database is op een SQL-server in een elastic pool. Dit zorgt ervoor dat als er nood is aan een langere bewaarperiode of meer opslag, dit ook mogelijk is.

Deze oplossing werkt met een pay-as-you-go-model en heeft een vaste kost van €90,57. Per uur komt er nog eens €0,29 bij. Bij dit bedrag zal nog een administratieve kost komen.

6. AI-ontwerp

Er werd gevraagd aan de verschillende projectgroepen om een applicatie te maken die de mood en de events kan detecteren op evenementen. Hier komt natuurlijk AI aan te pas. Als allereerste moeten we een dataset samenstellen. Wanneer de kwaliteit van deze dataset voldoende hoog is, kunnen hiermee AI-modellen getraind worden. AI-modellen zijn steeds gekenmerkt door een bepaald AI-algoritme. Het is essentieel dat deze voortdurend worden verbeterd zodat de nauwkeurigheid van de modellen hoog is. Eenmaal we nauwkeurige AI-modellen hebben, kunnen we deze gebruiken om audio te detecteren. Deze informatie kunnen we dan via een API doorsturen naar de applicatie. In dit deel van het rapport bespreken we de aard van onze dataset en AI-model.

6.1. Dataset

Wat betreft de dataset, kunnen we het project opsplitsen in twee delen: event detection en mood detection. Aangezien dit twee verschillende dingen zijn die geclassificeerd moeten worden, zullen hier dus ook twee verschillende datasets voor nodig zijn.

6.1.1. Event detection

Voor event detection gingen we op zoek naar datasets die voldoende data bevatten van mogelijke events die zich op een evenement kunnen voordoen. We denken hierbij aan een vuurpijl die ontploft, een geweer dat wordt afgeschoten, een hond die blaft, onweer dat uitbreekt, ... Voor het zoeken van de data maakten we gebruik van de site Kaggle. Op deze site staan tal van gratis te gebruiken datasets.

Wij hebben gekozen voor de dataset 'Environmental Sound Classification 50'. Deze dataset bestaat uit vijftig klassen van veelvoorkomende geluiden en omvat dus zeker onze selectie aan geluiden. Voor elke klasse zijn er een vijftigtal audiobestanden (WAV) van vijf seconden. Dit geeft ons dus voldoende data om een model voor event detection te trainen. Deze audiobestanden zijn al gelabeld en dit hoeven we dus niet meer zelf te doen.

6.1.2. Mood detection

Op vlak van mood detection bleek het onmogelijk valide datasets te verzamelen online. De beschikbare datasets online waren datasets die te maken hadden met mood detection, maar dan van slechts één persoon. Het betrof steeds data van één persoon die één zin op verschillende intonaties en met een verschillende emotie uitsprak. Dit is voor ons AI-model niet van toepassing aangezien we mood detection van menigtes willen doen. De verschillen liggen hier dan bij het feit dat deze data zich focust op de manier waarop de zin wordt uitgesproken en bij menigtes gaat het eerder over het algemene stemgeluid. Deze data gebruiken om een model te trainen zou dus leiden tot een fout model.

In samenspraak met Mediaan hebben we dan besloten deze data zelf te verzamelen, knippen en labelen. Om de audio te verzamelen doen we beroep op sites zoals YouTube en Freesound. Deze sites bevatten een uitgebreide bibliotheek aan audiofragmenten die we kunnen downloaden om ons model op te baseren. Aangezien dit een repetitief werk is, zal het verzamelen en labelen van de audio verdeeld worden over de drie teams.

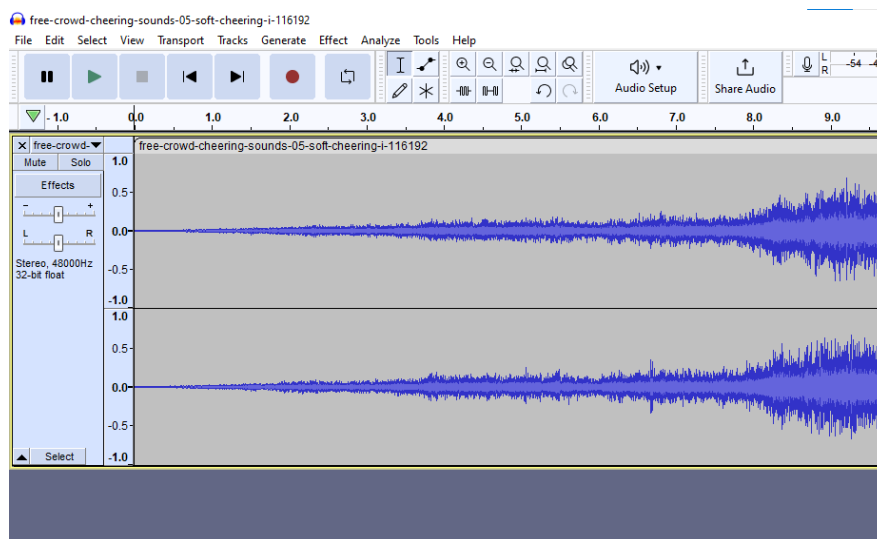
We zullen deze fragmenten opsplitsen in drie verschillende categorieën: negatief (-1), neutraal (0) en positief (+1). Onder de negatieve categorie gaan we zowel bange als boze gevoelens zetten omdat het ons te complex lijkt dit te onderscheiden. De neutrale categorie zal de grondlijn zijn waarop we ons baseren, het ruisgeluid als het ware. De positieve categorie spreekt voor zich en zal alle emoties van geluk en uitlating omvatten.

Eenmaal ze verdeeld zijn over de verschillende categorieën is het tijd om ze te labelen aan de hand van een audio labeling tool die verder beschreven wordt.

6.1.3. Audio labeling

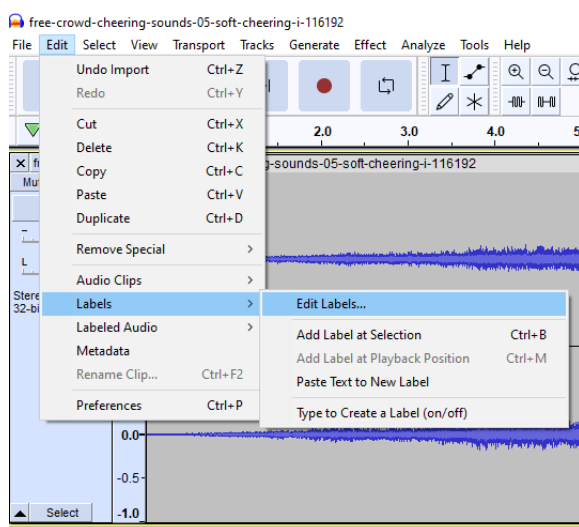
We gaan zelf een dataset aanmaken, wat betekent dat we de data zelf moeten labelen. Het labelen van deze data is essentieel voor het classificeren van geluid.

De annotatie tool die we gebruiken is Audacity. Dit is veruit de meest toegankelijke tool die we kunnen gebruiken. Het is gemakkelijk te installeren, labels geven aan stukken audio gaat zeer snel en we kunnen de gelabelde audiostukken meteen opslaan in een .txt file. Dit heeft als voordeel dat we de audio fragmenten kunnen labelen en knippen in één tool. Om ervoor te zorgen dat we ze kunnen classificeren, krijgen de bestanden steeds een naam die overeenkomt met hun klasse, bijvoorbeeld positief_1.wav. Hieronder wordt een korte demonstratie van dit proces gegeven.



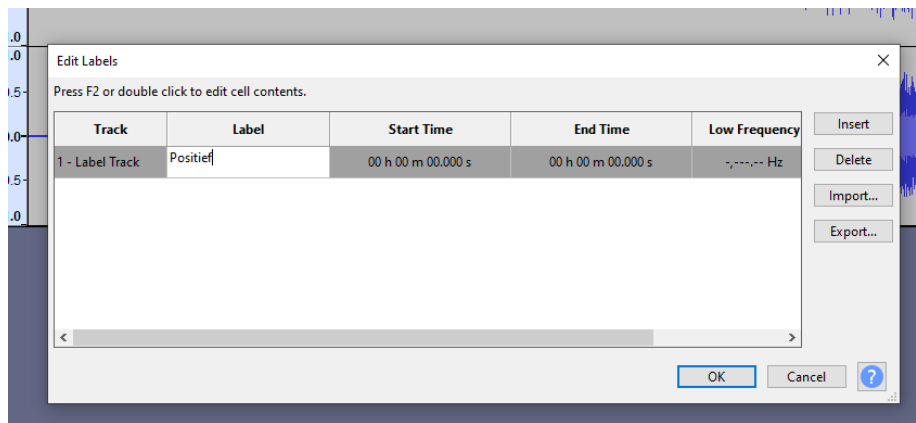
Figuur 24: Stap 1 (audio labeling)

Als eerste stap openen we een audiobestand in Audacity.



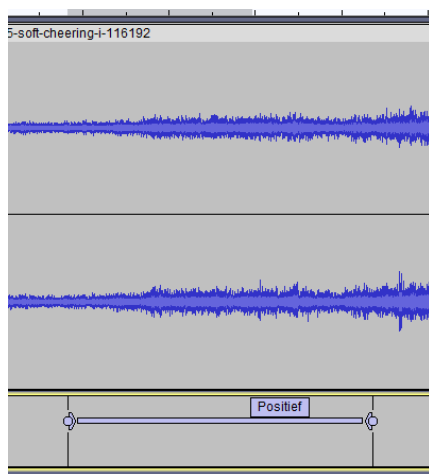
Figuur 25: Stap 2 (audio labeling)

Daarna gaan we naar Edit => Labels => Edit Labels en klikken hierop.



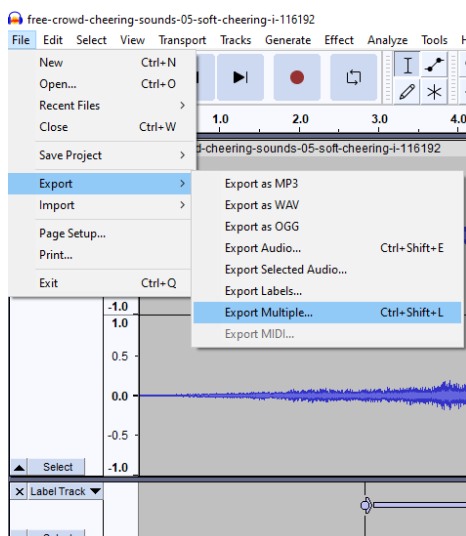
Figuur 26: Stap 3 (audio labeling)

Vervolgens geven we een naam aan de label en duwen hier op OK.



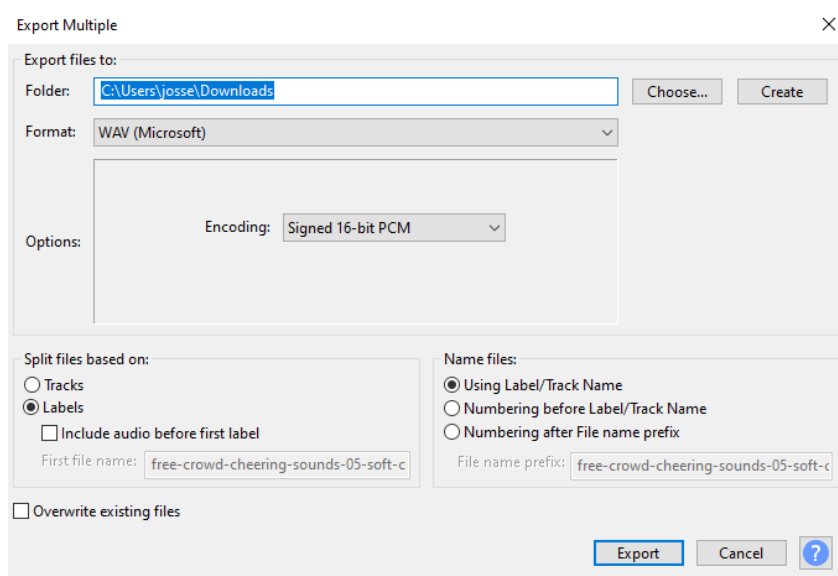
Figuur 27: Stap 4 (audio labeling)

Daarna kiezen we aan welke tijdstippen we een label geven.



Figuur 28: Stap 5 (audio labeling)

Om de stukken audio die gelabeld zijn te exporteren, gaan we naar File => Export => Export Multiple en klikken hierop.



Figuur 29: Stap 6 (audio labeling)

Als laatste duiden we aan in welke folder de bestanden moeten opgeslagen worden en klikken op Export.

 Positief-2.wav	18/12/2022 20:20	WAV-bestand	663 kB
--	------------------	-------------	--------

Figuur 30: Naam opgeslagen file

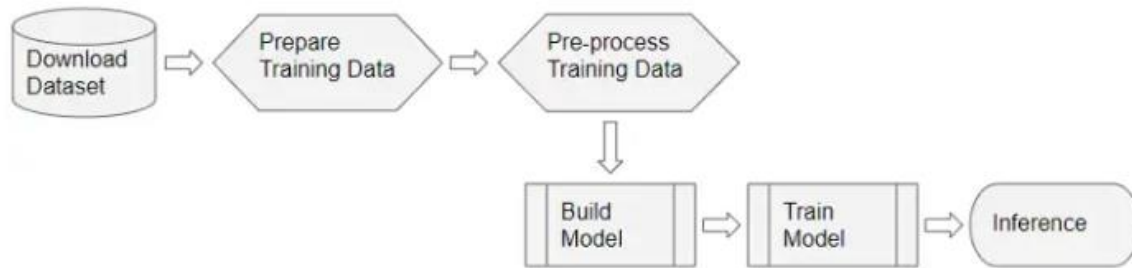
Het bestand is nu opgeslagen in de gewenste folder waarbij de naam overeenkomt met het label dat hieraan werd gegeven. De bestanden worden steeds genummerd per label.

6.2. Algoritme

Alvorens een model te kunnen trainen, moeten we weten welk soort algoritme we gaan gebruiken. Een AI-algoritme is een reeks instructies om logische en wiskundige problemen op te lossen en vertelt de computer hoe hij zelfstandig moet leren werken. Hieronder wordt duidelijk welk AI-algoritme we gekozen hebben en waarom we deze kozen.

Bij het onderzoek naar mogelijke bruikbare algoritmen, bleek vrij snel dat CNN als beste keuze uit de bus kwam.

Voordat we deze keuze konden maken, moesten we ons eerst de vraag stellen: welk type AI moeten we gebruiken? Omdat we een AI-model moeten trainen die zonder enige menselijke inbreng een geluid moet kunnen classificeren, hebben we gekozen voor deep learning. Het meest gebruikte deep learning algoritme bleek CNN te zijn. Dit was ook het algoritme dat steeds gebruikt werd bij al bestaande projecten over de classificatie van geluid. In de onderstaande figuur worden de stappen weergegeven die nodig zijn om een CNN-model aan te maken.



Figuur 31: Deep Learning Workflow

Eerst wordt er dus data verzameld. Deze data zijn nog niet meteen klaar voor gebruik, ze kunnen niet rechtstreeks in het model worden ingevoerd. We moeten de audiogegevens uit het bestand laden en zodanig verwerken dat ze een formaat hebben dat het model verwacht. Bij de verwerking van de audiogegevens zorgen we er ook voor dat ze gelabeld worden, het gelabelde deel van de audiogegevens wordt apart opgeslagen in het gewenste formaat. Deze audio fragmenten worden steeds bewaard in een folder waarvan de naam overeenkomt met de label die ze hebben gekregen, deze folders bevinden zich in één “main” folder.

Vanaf het moment dat alle data in de “main” folder zit, kan de trainingsdata klaargemaakt worden. In deze stap zullen de paden van de folders die werden aangemaakt opgeslagen worden in variabelen. Eenmaal de paden zijn vermeld kunnen we deze audiofiles bekijken op Jupyter Notebook aan de hand van python code.

Vervolgens gaan we de dataset standaardiseren en augmentaties uitvoeren op de dataset. Het standaardiseren van de dataset zorgt ervoor dat de kwaliteit verbetert. De augmentaties verbeteren dan weer de prestaties en resultaten van ML-modellen. Deze stap zal pas uitgevoerd worden tijdens de runtime wanneer we audiobestanden zullen lezen en laden. Aangezien audiogegevens vaak behoorlijk groot en geheugenintensief zijn, willen we niet de hele dataset op voorhand in het geheugen lezen. We bewaren alleen de audiobestandsnamen in onze training data. Pas tijdens de runtime laden we de audiogegevens en verwerken we deze door transformaties op de audio toe te passen. We moeten er ook voor zorgen dat de audiofiles omgezet worden in de vorm van spectrogrammen, deze zullen gebruikt worden door ons deep learning model.

Voordat we beginnen met het bouwen van het deep learning model, maken we eerst de data pipeline. Om deze te kunnen maken, moeten we de data inladen die we hebben verkregen bij de pre-processing. We vermelden hier ook de batch size. Dit zorgt ervoor dat niet alle data in één keer aan het algoritme wordt doorgegeven.

Bij de volgende stap bouwen we een Sequential model. We kunnen deze architectuur ontwikkelen met een aangepast modelarchetype. Vervolgens voegen we de convolutielagen toe. Hierna kan de verkregen output van de convolutielagen worden afgevlakt om deze geschikt te maken voor verdere verwerking. Tot slot voegen we de volledig verbonden lagen toe met de Sigmoid-activeringsfunctie.

Eenmaal de architectuur van het model gebouwd is, kunnen we het model compileren en trainen. We zullen dit model trainen aan de hand van de eerder vermelde data. Het trainen en fitten van het model zullen we enkele epochs laten lopen. Na het trainen van het model zullen we de resultaten analyseren en valideren. Als het een goed model is, zullen we het gebruiken in onze applicatie. Maar als de resultaten niet zoals verwacht zijn, zullen we het model opnieuw trainen met andere parameters om toch tot een goed resultaat te komen.

Tot slot zullen we het model gebruiken om voorspellingen uit te voeren. Dit zullen we doen door de audioclips die het model gaat verwerken, op te splitsen in kleinere fragmenten zodat het de detectie kan uitvoeren. Zodra we de events voor elk fragment bepaald hebben, kunnen we het totale aantal berekenen door alle individuele waarden bij elkaar op te tellen.

6.3. MLOPs

Als we eenmaal een model hebben, is het essentieel dat dit model continu kan leren. Continu leren is het vermogen van een model om voortdurend bij te leren van een gegevensstroom. Het model zou in praktijk dus ondersteund moeten worden zodat het zelfstandig kan leren en zich kan aanpassen tijdens de productie als er nieuwe data binnenkomt. Het doel hiervan is dus om gegevens die binnenkomen te gebruiken en daarmee het model automatisch te trainen zodat het model een hoge nauwkeurigheid behoudt en eventueel zelfs nauwkeuriger wordt.

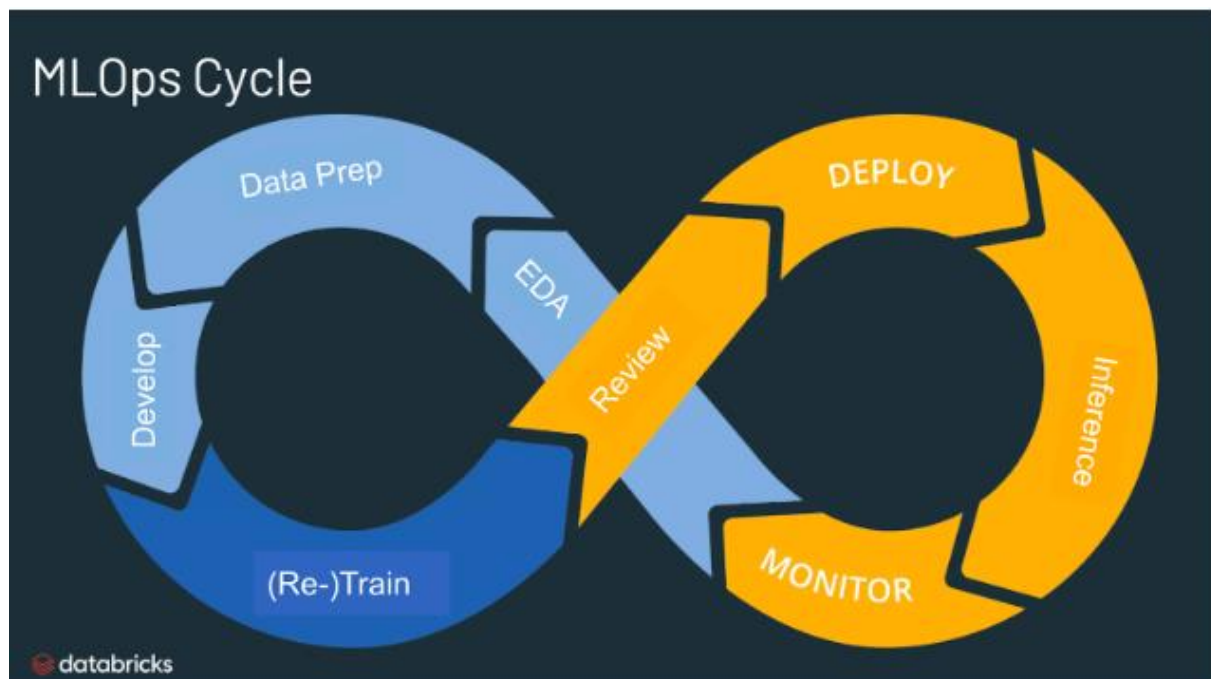
Een andere gewenste eigenschap van het model is, dat het herbruikbaar moet zijn. Dit betekent dus dat het model ook moet kunnen gebruikt worden voor andere toepassingen.

Om deze eigenschappen toe te passen op ons model maken we gebruik van MLOPs. MLOPs is een kernfunctie van Machine learning engineering en richt zich op het stroomlijnen van het proces waarbij AI-modellen omgezet worden naar de productie. Vervolgens richt MLOPs zich op het onderhouden en bewaken van deze modellen.

Het volgende deel omschrijft een cyclus van een continu lerend model. Maar aangezien dit vrij complex is om te realiseren op de korte tijdspanne zien we dit niet als haalbaar. Het onderzoek is echter wel gebeurd en dit is dus een 'Nice to have' die we zullen implementeren als er tijd voor is.

6.3.1. De MLOPs cyclus

De werking van MLOPs kan worden voorgesteld door een cyclus. In de onderstaande cyclus worden alle stappen die getoond die gevolgd worden bij de implementatie van MLOPs in projecten.

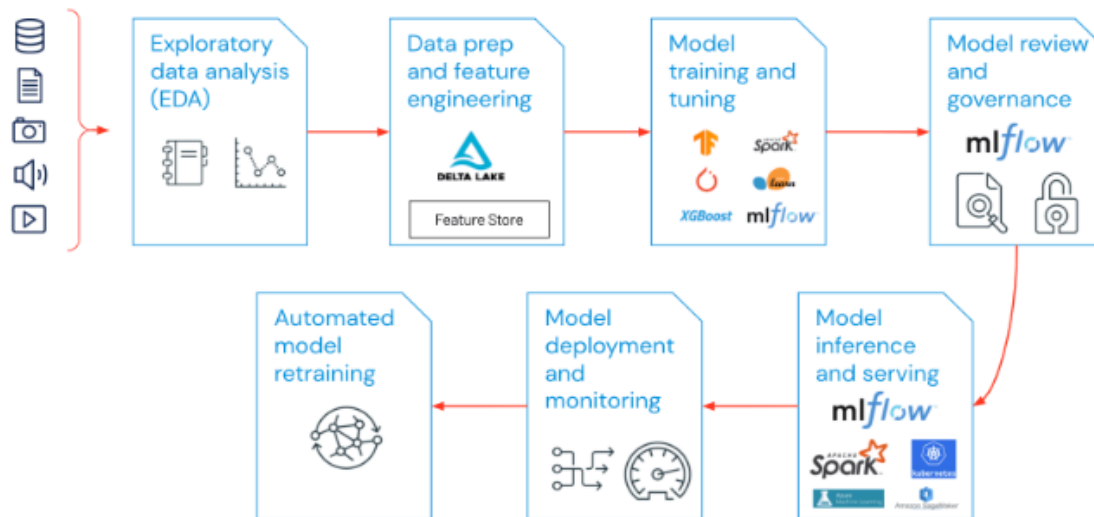


Figuur 32: MLOPs cyclus

In het begin van een project zal eerst de stap EDA uitgevoerd worden. We kunnen dan uit deze foto afleiden dat de laatste stap MONITOR zal zijn. Hoewel dit niet helemaal klopt. Deze cyclus is namelijk itererend, wat betekent dat hij steeds wordt herhaald. MLOPs omvat het experimenteren, itereren en continu verbeteren van deze cyclus.

6.3.2. De componenten van MLOPs

MLOPs bestaat uit verschillende specifieke componenten. Deze componenten worden getoond in de onderstaande figuur.



Figuur 33: Componenten van MLOPs

We zullen nu kort even de inhoud van deze componenten bespreken, zodat het duidelijk is wat er in elk van deze stappen gebeurt. Enkel de componenten die voor ons van toepassing zijn, zullen we hieronder vermelden.

EDA: In deze stap wordt de data iteratief verkend, gedeeld en voorbereid voor de machine learning cyclus. Dit gebeurt door het creëren van reproduceerbare, bewerkbare en deelbare datasets en visualisaties.

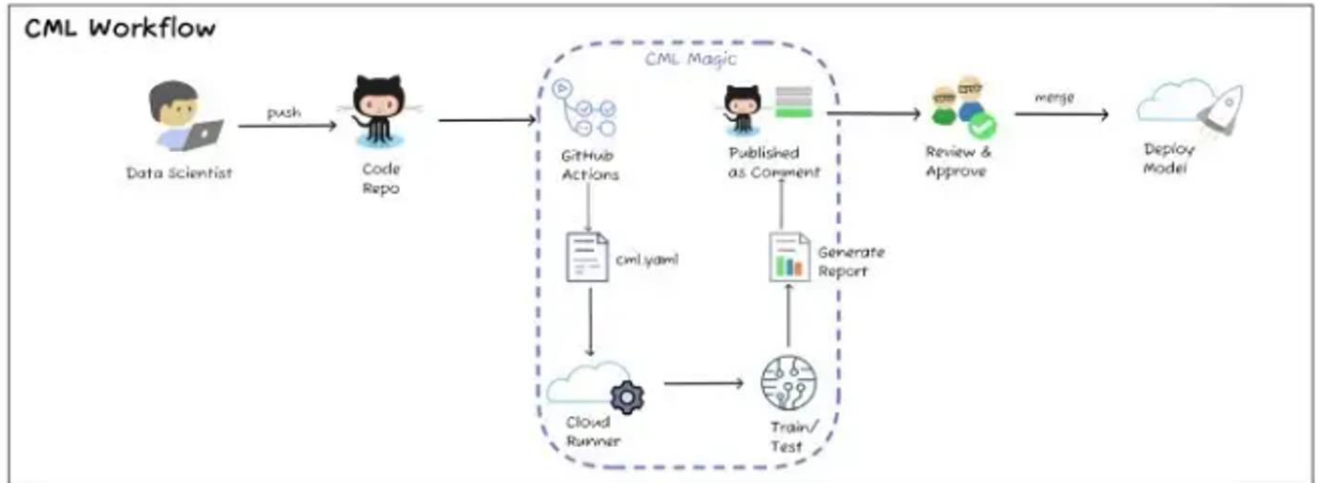
Data preparatie en feature engineering: Vervolgens moeten we iteratief de data transformeren, samenbrengen en redupliceren, hierdoor krijgen we verfijnde features.

Trainen en afstemming van het model: We maken in deze stap gebruik van open source libraries om het model te trainen en verbeteren. Eén van de libraries die we al zeker voor dit project kunnen gebruiken is PyTorch. Deze library wordt in vele projecten over de classificatie van geluid gebruikt.

Beoordeling van het model: Als het model getraind is, moeten we het beoordelen. Dit gebeurt door het berekenen van de accuraatheid van het model en plotten van grafieken waarin wordt weergegeven hoe goed ons model is getraind.

Deployen van het model: Indien het model goed beoordeeld is, kunnen we het opslaan als een pikkel file. Dankzij Flask, een webapplicatie-framework voor Python, kan de data die het model levert, verstuurd worden via een API naar onze applicatie. Rond deze Flask staat een Unicorn server, een webserver voor python applicaties. Deze zorgt voor de beveiliging tijdens het deployen van een model.

Geautomatiseerde retraining van het model: Het model moet continu kunnen bijleren, het is daarom belangrijk dat het direct datasets kan trainen in het geval er nieuwe data binnenkomt. Dit gebeurt aan de hand van CML. Bij deze methode worden yaml files gebruikt om het leerproces steeds opnieuw op te starten. Hieronder wordt dit proces wat duidelijker gemaakt aan de hand van een foto en een korte beschrijving.



Figuur 34: CML-workflow

In dit proces sturen degene die verantwoordelijk zijn voor AI hun ML model door naar een GitHub repository. Eenmaal het model op GitHub staat, kan er via GitHub Actions een yaml file gemaakt worden waarin alle configuraties gedefinieerd worden. Wanneer een pull request wordt ingediend, gebruikt GitHub actions deze workflow en voeren ze activiteiten uit die zijn gespecificeerd in het configuratiebestand.

6.4. Toepassing in applicatie

Tijdens het evenement zullen de microfoons continu alles opnemen wat er gebeurt. Op deze audiodata worden de vooraf getrainde modellen losgelaten. Onze modellen zullen dan twee verschillende dingen doen.

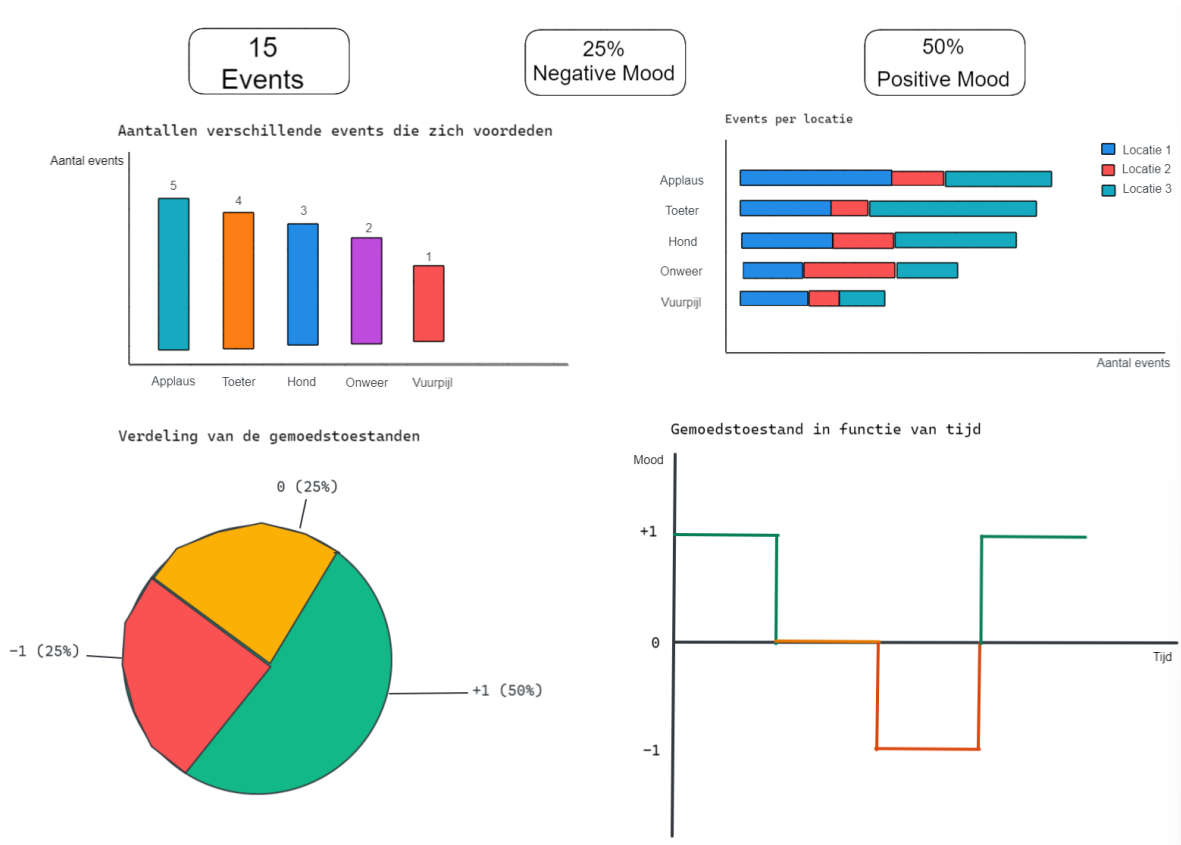
Ten eerste gaan alle geluiden die als abnormaal of gevaarlijk beschouwd worden, gedetecteerd worden. Dit zijn de geluiden zoals de vuurpijl die ontploft die we in een vorig hoofdstuk besproken hebben. Om deze audiodata te lokaliseren, zal gekeken worden naar welke microfoon het geluid het sterkst heeft waargenomen.

In het andere aspect van het AI-systeem, zal het systeem continu bijhouden in welke gemoedstoestand de menigte zich bevindt. Het systeem gaat op elk tijdstip een waarde meegeven aan de applicatie die overeenstemt met één van de mogelijke gemoedstoestanden.

Zoals eerder vermeld zal het AI-systeem deze data doorsturen naar de applicatie via een API en zal de applicatie een melding geven aan de teamlead van wat het systeem gedetecteerd heeft en wat er zich dus juist afspeelt op het evenement. De teamlead zal ook op het einde enkele statistieken over de avond en een grafiek die de mood van het publiek over heel de avond weergeeft, te zien krijgen.

6.5. Dashboard prototype

Hieronder kan u een prototype terugvinden van een dashboard. Dit is een voorbeeld hoe het er uit zou kunnen zien. De data zijn echter nog niet representatief. We geven enkele grafieken weer in verband met de events die zich hebben voorgedaan, de locaties van die events en de mood van de menigte.



Figuur 35: Prototype van een dashboard

Tijdens de realisatiefase zullen we werken met PowerBI om dit dashboard te realiseren. Om de kosten te drukken zullen we dit in de desktopversie van PowerBI realiseren in plaats van in de onlineversie.

6.6. Conclusie

Wat kunnen we nu concluderen uit het AI-gedeelte? Als eerste hebben we datasets gezocht op het internet. Datasets in verband met event detection hebben we zeer snel gevonden. Datasets bruikbaar voor mood detection vonden we echter helemaal niet. Dus hebben we besloten om deze dataset zelf aan te maken. Hiervoor moeten we een annotatietool gebruiken. De annotatietool die we uiteindelijk gekozen hebben is Audacity. Deze bleek de meest gebruiksvriendelijk tool te zijn. We kunnen eenvoudig de gelabelde stukken audio meteen opslaan in folders. Omdat we alleen gebruik maken van Audacity, zijn we genooddaakt om alleen .wav-bestanden te gebruiken als dataset. Csv-bestanden opstellen waarin de labels en naam van de audiofragmenten staan, zou te veel tijd vragen.

Als tweede hebben we gekozen welk algoritme we zullen gebruiken. CNN blijkt de beste keuze te zijn. Dit is veruit het meest gebruikte deep learning-algoritme en wordt ook vaak gebruikt bij classificatie van geluid. Omdat dit algoritme zo populair is, hebben we toegang tot veel informatie over de implementatie van dit algoritme in projecten via het internet.

Verder is het ook de bedoeling om MLOPs te implementeren in het project. Aangezien dit vrij complex is om te realiseren op de korte tijdspanne, zien we dit niet als haalbaar. Het onderzoek is echter wel gebeurd en dit is dus een 'Nice to have' die we zullen implementeren als er tijd voor is.

Ook moet het mogelijk zijn om met AI-modellen audiodata te classificeren via een applicatie. We hebben besloten dat we met Flask via een API dit model zullen implementeren in de applicatie. Omdat we met Flask werken, zullen we ook gebruik maken van Gunicorn, wat zorgt voor de beveiliging van deze API.

Tot slot zullen we de desktopversie van PowerBI gebruiken om de dashboards te realiseren. Dit heeft verder geen effect op de uitwerking van de 'Proof of concept'.

7. Algemeen besluit

Het opzet van dit project is een 'Proof of concept' uit te werken voor een cloudapplicatie die het mogelijk maakt aan crowd monitoring te doen op basis van audio. Tot op heden gebeurde crowd monitoring voornamelijk op basis van camerabeelden. Dit brengt enkele problemen met zich mee die dit project probeert op te lossen. De moods die het AI-systeem zal kunnen detecteren zijn negatief, neutraal en positief. Het AI-systeem zal ook events kunnen detecteren. Deze informatie zal dan doorgestuurd worden naar de teamlead die aan de hand van een webapplicatie securitypersoneel kunnen uitsturen naar de desbetreffende locatie. De hele webapplicatie zal draaien aan de hand van web app services binnen een Azure-omgeving. Om dit te realiseren, wordt dit project opgedeeld in drie verschillende domeinen: AI, CCS & APP.

Uit de volledige analyse is gebleken dat VueJS de beste mogelijkheid is om als front-end tool te dienen. Deze zal ondersteund worden door een .NET back-end. Het gebruikte databasemanagementsysteem dat van toepassing zal zijn tijdens de uitwerking van dit project is MySQL. Dit bleek de beste optie uit een reeks van WRM onderzoeken.

De toepassing is alleen op te bouwen en af te breken door Mediaan. De klant kan via een script, geschreven via Bicep, de toepassing opbouwen en afbreken. Op deze manier blijven de kosten beperkt. De app zal gedraaid worden via Azure web app service. Deze is compatibel met Flask wat nodig is voor de AI-toepassing. Deze hangt vast aan een database met back-up. Dit zal een vaste kost geven van €90,57 en per uur zal er nog €0,29 bijkomen. Dit bedrag, samen met de vaste kost, zorgt ervoor dat het bedrag van €200/maand zeker niet overschreden wordt.

Om de moods en events te detecteren zal het AI-systeem gebruik maken van deep learning-modellen die getraind zijn met data in verband met events en moods. Dit zal leiden tot twee modellen die door middel van een Flask-API informatie doorsturen naar de rest van de applicatie. De verkregen data zullen gevisualiseerd worden aan de hand van een dashboard in PowerBI.

Zo hebben we een volledige applicatie van A tot Z uitgewerkt die voldoet aan alle requirements. Deze 'Proof of concept' zal gebruikt worden om dit project te realiseren in de realisatiefase.

Figurenlijst

Figuur 1: CI/CD pipeline Azure Mediaan	12
Figuur 2: Containers Azure Mediaan	12
Figuur 3: Datamodel	13
Figuur 4: Use Case Diagram	14
Figuur 5: inloggen PC	15
Figuur 6: Inloggen mobiel	15
Figuur 7: Homepage Admin	16
Figuur 8: Homepage Teamlead	16
Figuur 9: Uitnodigingsmail sturen naar teamleads	17
Figuur 10: Evenement aanmaken & beheren	18
Figuur 11: Teamlead koppelen aan een evenement	19
Figuur 12: Account aanmaken	20
Figuur 13: Account beheren	20
Figuur 14: Applicatie opstarten/uitzetten	21
Figuur 15: Uitnodigingsmail sturen naar securitypersoneel	22
Figuur 16: Security toewijzen aan een evenement	23
Figuur 17: Locaties toewijzen aan een evenement	24
Figuur 18: Dashboards bekijken	26
Figuur 19: Personeel uitsturen naar een locatie	27
Figuur 20: Geschiedenis opvragen	28
Figuur 21: Confirmeren om naar een locatie te gaan	29
Figuur 22: Feedback geven op locatie	30
Figuur 23: Schematische voorstelling omgeving	35
Figuur 24: Stap 1 (audio labeling)	38
Figuur 25: Stap 2 (audio labeling)	38
Figuur 26: Stap 3 (audio labeling)	39
Figuur 27: Stap 4 (audio labeling)	39
Figuur 28: Stap 5 (audio labeling)	39
Figuur 29: Stap 6 (audio labeling)	40
Figuur 30: Naam opgeslagen file	40
Figuur 31: Deep Learning Workflow	41
Figuur 32: MLOPs cyclus	42
Figuur 33: Componenten van MLOPs	43
Figuur 34: CML-workflow	44
Figuur 35: Prototype van een dashboard	45
Tabel 1: WRM Databases	32
Tabel 2: WRM Azure omgeving	33
Tabel 3: Kostprijs per maand en uur voor 2 en 4 cores	34
Tabel 4: WRM storage	35
Tabel 5: Totale geschatte kostenplaatje	36

Referenties

- Bharath, K. (2022, Juni). *Audio Classification with Deep Learning*. Opgehaald van PaperspaceBlog: <https://blog.paperspace.com/audio-classification-with-deep-learning/>
- Community. (2022, December 19). *Azure subscription and service limits, quotas, and constraints*. Opgehaald van Learn Microsoft: <https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits#api-management-limits>
- Databricks. (sd). *MLOps*. Opgehaald van Databricks: <https://www.databricks.com/glossary/mlops>
- Doshi, K. (2021, Maart 18). *Audio Deep Learning Made Simple: Sound Classification, Step-by-Step*. Opgehaald van Towards Datascience: <https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5>
- Hermann, & Andrew, M. (2017, December 13). *Difference between Azure Container Service and Web App for Containers*. Opgehaald van Stackoverflow: <https://stackoverflow.com/questions/47232013/difference-between-azure-container-service-and-web-app-for-containers>
- jimmart-dev, stevenmatthew, atikmapari, & tamram. (2022, Juli 14). *Scalability and performance targets for Blob storage*. Opgehaald van Learn Microsoft: <https://learn.microsoft.com/en-us/azure/storage/blobs/scalability-targets>
- jimmart-dev, stevenmatthew, tamram, mittalpri, alexbuckgit, bandersmsft, . . . jtracey93. (2022, 14 Juli). *Optimize costs for Blob storage with reserved capacity*. Opgehaald van Learn Microsoft: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-reserved-capacity>
- Microsoft. (sd). *API Management pricing*. Opgehaald van Azure Microsoft: <https://azure.microsoft.com/en-us/pricing/details/api-management/>
- Microsoft. (sd). *Microsoft Communication*. Opgehaald van Portal Azure: <https://portal.azure.com/#create/Microsoft.Communication>
- Microsoft. (sd). *Microsoft Server Elasticpools*. Opgehaald van Portal Azure: <https://portal.azure.com/#view/HubsExtension/BrowseResource/resourceType/Microsoft.Sql%2Fservers%2Felasticpools>
- Microsoft. (sd). *Microsoft Website*. Opgehaald van Portal Azure: <https://portal.azure.com/#view/HubsExtension/NoSubscriptionsBlade/collectorBindingInternals-inputs~/%7B%22options%22%3A%7B%22galleryOptions%22%3A%7B%22uiDefinition%22%3A%7B%22markup%22%3A%7B%22reference%22%3A%7B%22provisioningPartName%22%3A%22NoSubscripti>
- Microsoft. (sd). *Pricing Calculator*. Opgehaald van Azure Microsoft: <https://azure.microsoft.com/en-us/pricing/calculator/>
- Microsoft. (sd). *Web Apps*. Opgehaald van Azure Microsoft: <https://azure.microsoft.com/en-gb/products/app-service/web/>

- mumian, alexbuckgit, tejas-nagchandi, alex-frankel, ucheNkadiCode, tfitzmac, . . . davidsmatlak. (2022, November 3). *What is Bicep?* Opgehaald van Learn Microsoft: <https://learn.microsoft.com/en-us/azure/resource-manager/bicep/overview?tabs=bicep>
- Nandi, P. (2021, Maart 24). *CNNs for Audio Classification*. Opgehaald van Towards Datascience: <https://towardsdatascience.com/cnns-for-audio-classification-6244954665ab>
- Nayak, Y. (2022, Maart 29). *Continuous Machine Learning*. Opgehaald van Towards Datascience: <https://towardsdatascience.com/continuous-machine-learning-e1ffb847b8da>
- pemari-msft, tamram, ShannonLeavitt, mhopkins-msft, & robinsh. (2021, Juli 20). *Understanding block blobs, append blobs, and page blobs*. Opgehaald van Learn Microsoft: <https://learn.microsoft.com/en-us/rest/api/storageservices/understanding-block-blobs--append-blobs--and-page-blobs>
- Sharma, H. (2021, Oktober 7). *Azure Storage: A complete Guide for Beginners*. Opgehaald van Intellipaat: <https://intellipaat.com/blog/tutorial/microsoft-azure-tutorial/azure-storage/#:~:text=Azure%20Storage%20Types%201%20Azure%20blob%20storage%3A%20It,the%20Server%20Message%20Block%20%28SMB%29%20protocol.%20Meer%20items>
- tamram, normesta, jimmart-deve, v-dalc, jangelfdez, DCtheGeek, . . . tompratt-AQ. (2022, November 08). *Introduction to Azure Blob Storage*. Opgehaald van Learn Microsoft: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>
- v-amallick, v-hearya, rolyon, dcurwin, DCtheGeek, georgewallace, . . . Sogup. (2022, Juni 23). *Recovery Services vaults overview*. Opgehaald van Learn Microsoft: <https://learn.microsoft.com/en-us/azure/backup/backup-azure-recovery-services-vault-overview>
- Wadhwani, P. (2022, December 16). *Why Use Vue JS? Frontend's Every Aspect Covered*. Opgehaald van Bacancytechnology: <https://www.bacancytechnology.com/blog/why-use-vue-js#:~:text=and%20Market%20Share-,Vue.,frameworks%20for%20front%2Dend%20development>
- Wikipedia. (2022, April 28). *Vue.js*. Opgehaald van Wikipedia: <https://nl.wikipedia.org/wiki/Vue.js>
- WilliamDAssafMSFT, verargulla, alexbuckgit, v-dirichards, rothja, cmcclister, . . . ShawnJackson. (2022, September 28). *What is Azure SQL Database?* Opgehaald van Learn Microsoft: <https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql#availability-capabilities>
- yochze. (2019, Juli 11). *How to apply continual learning to your machine learning models*. Opgehaald van Towards Datascience: <https://towardsdatascience.com/how-to-apply-continual-learning-to-your-machine-learning-models-4754adcd7f7f>