

## A

In 2018 toonde Rammeloo aan, dat China het verst vooruit loopt als het gaat om gezichtsherkenningstechnologie:

- het opengaan van metropoortjes;
- het betalen van eten;
- toegang krijgen tot pretparken;
- opsporen van criminelen.

In Twente werkt het bedrijf 20face met gezichtsherkenningstechnologie dat ontwikkeld is aan de Universiteit in Twente. In het voetbal stadion van Heracles Almelo loopt momenteel een proef met dit systeem (20face, z.j.):

- Vip Klanten krijgen via gezichtsherkenning toegang.

In 2018 schreven Kist en Hartholt twee artikelen over het gebruik van deze software in Nederland. De voorbeelden uit deze artikelen zijn:

- In smartphones is deze techniek een ontgrendelingsmethode.
- Nederlandse politie spoort verdachten op door beelden van beveiligingscamera's te koppelen aan een database met gezichten van criminelen.
- Gemalto een Frans-Nederlands beveiligingsbedrijf bracht in 2018 een systeem op de markt waarmee beveiligingscamera's in grote hoeveelheden live gezichten kunnen herkennen.



## B

Frew (2017) constateert dat gezichtsherkenningstechnologie een grote aanslag op de privacy heeft, mensen nog niet klaar zijn om overal achtervolgt te worden. Onopgemerkt worden persoonlijke gegevens opgeslagen zonder dat je daar invloed op hebt. Volgens De Bryne (2018) is dit in strijd met de Europese privacyverordening, welke zegt dat een persoon waarvan gegevens worden verwerkt, hiervan op de hoogte moet zijn. Deze privacyverordening wordt de algemene verordening gegevensbescherming (AVG) genoemd.

- aanslag privacy
- tegenstrijd AVG

The Economist publiceerde een bericht over twee onderzoekers die beweren gezichtsherkenning software te hebben ontwikkeld, wat naast het geslacht ook laat zien of een persoon hetero of homoseksueel is (Gershorn, 2017)

Als deze technologie in de handen komt van strenge regimes in landen waar homoseksuelen vervolgd worden, kan dit vervelende gevolgen hebben. Naast dat homoseksuelen in zulke landen niet meer veilig zijn, kunnen mensen ook ten onrechte homoseksueel verklaard worden, wat kan leiden tot gevangenschap of zelf de dood.

- Misbruik / vervolging die niet binnen de standaard definitie van een geloof/land vallen.

## C

In 2017 onderzocht Anthony de werking de Microsoft Azure Face API. De cloud-gebaseerde Face API die ontwikkelaars toegang biedt tot geavanceerde gezicht algoritmen dat menselijke gezichten detecteert in digitale afbeeldingen. Hierbij kunnen één of meerdere menselijke gezichten in afbeeldingen worden gedetecteerd (Azure microsoft, z.j.).

Voordelen	Nadelen
Beschikt over gezichtsherkenning API	Gratis beperkt te gebruiken
Beschikt over gezichtsdetectie API	
Beschikt over emotie API	
Beschikt over een gratis SDK	
Werkt met meerdere gezichten	
Veel documentatie	

Tabel 1: Voor-en nadelen Microsoft Azure Face API

Bronnenlijst: Azure microsoft (z.j)

Kairos biedt een breed scala oplossingen voor gezichtsherkenning via hun API aan. Hun API-eindpunten omvatten, het identificeren van geslacht, emotie, leeftijd en gezichtsherkenning in zowel foto's als video's. Kairos beschikt over een SDK, welke alleen tegen betaling beschikbaar wordt gesteld (Kairos, z.j.).

Voordelen	Nadelen
Beschikt over gezichtsherkenning API	Gratis beperkt te gebruiken
Beschikt over gezichtsdetectie API	Alleen tegen betaling SDK beschikbaar
Kan emotie herkennen	
Werkt met meerdere gezichten	
Veel documentatie	

Tabel 2: Voor-en nadelen Kairos

Bronnenlijst: Kairos (z.j)



Visual begrijpt de inhoud van afbeeldingen. Het analyseert afbeeldingen van objecten, gezichten, eten, kleuren en tekst en is gericht op het trainen van je eigen model in minuten. Met slechts enkele afbeeldingen kan het elk nieuwe object, elk nieuwe kenmerk of elke persoon leren (IBM, z.j.).

Voordelen	Nadelen
Beschikt over gezichtsherkenning API	Gratis beperkt te gebruiken
Beschikt over gezichtsdetectie API	Werkt alleen met afbeeldingen, geen video
Kan emotie herkennen	
Werkt met meerdere gezichten	
Gratis SDK beschikbaar	
Veel documentatie	

Tabel 3: Voor-en nadelen IBM Watson Visual Recognition

Bronnenlijst: IBM (z.j)

Google Cloud Vision API geeft inzicht in de inhoud van afbeeldingen aan de hand van krachtige modellen voor machine learning te integreren REST API. Deze API biedt mogelijkheden tot het detecteren van meerdere gezichten in afbeeldingen, echter wordt gezichtsherkenning niet ondersteund (Google, z.j.).

Voordelen	Nadelen
Werkt met meerdere gezichten	Gratis beperkt te gebruiken
Beschikt over gezichtsdetectie API	Beschikt niet over gezichtsherkennings API
Veel documentatie	

Tabel 4: Voor-en nadelen Google Cloud Vision

Bronnenlijst: Google (z.j)



## D

Uit tabel 9 is op te maken dat Microsoft en Kairo het beste voldoen aan de vooropgestelde eisen. Google valt af omdat deze API niet over gezichtsherkenningstechnologie beschikt. IBM valt af omdat deze API alleen werkt met afbeeldingen (geen video).

Kijkend naar wat Microsoft en Kairo gratis aanbieden, komt naar voren dat Microsoft over een gratis SDK beschikt en Kairo niet. Het ontbreken van een gratis SDK maakt Kairo in dit een minder geschikte keuze. Volgens Sandoval (2016) versnelt een SDK het process, aangezien er beperkte tijd is biedt dit een groot voordeel. Een SDK: software development kit is een set van ontwikkelingstools, dankzij deze tools zoals beschikbare bibliotheken hoeft de programmeur minder zelf te programmeren en kan gebruik gemaakt worden van al bestaande functies (Sandoval, 2016). Het wiel hoeft als ware niet opnieuw uitgevonden te worden, vandaar dat werken met een SDK het process kan versnellen.



## E

JSON: Javascript-object notatie, een syntaxis voor het opslaan en uitwisselen van data.

Bij het uitwisselen van data tussen browser en server, kan de data alleen uit tekst bestaan. JSON is tekst dat omgezet kan worden, elk Javascript-object kan omgezet worden in JSON en verzonden worden naar de server. Andersom is natuurlijk ook mogelijk, elke JSON die van de server is ontvangen, kan geconverteerd worden naar Javascript-objecten (W3Schools, z.j.).

In 2017 onderzocht Sharma de voor-en nadelen van JSON. Deze voor-en nadelen zijn terug te vinden in tabel 20.

Voordelen	Nadelen
Makkelijk te gebruiken	Geen foutafhandeling
JSON slaat gegevens op in arrays, zodat gegevensoverdracht gemakkelijker wordt	Kan gevaarlijk zijn in combinatie met niet betrouwbare services
Syntaxis is klein en licht	
Support veel browsers	
Opzetten neemt weinig tijd in beslag	

Tabel 5: De voor-en nadelen van JSON



MySQL is een online database dat op een server wordt uitgevoerd. MySQL maakt gebruik van standaard SQL en is momenteel het populairste database systeem welke gebruikt maakt van PHP (W3Schools, z.j.).

De gegevens worden opgeslagen in tabellen. Een tabel is een verzameling van gerelateerde gegevens en bestaat uit kolommen en rijen.

In 2014 onderzochten Mack en Shah de voor-en nadelen van MySQL. Deze voor-en nadelen zijn verwerkt in tabel 21.

Voordelen	Nadelen
Makkelijk te gebruiken en op te zetten	Minder betrouwbaar dan MSSQL
Grote gemeenschap van ontwikkelaars waarbij je terecht kan voor hulp	Werking verminderd naarmate data toeneemt
Gratis open source*	Gratis open source*
Compatibel met vrijwel elk besturingssysteem	Minder kant en klaar functionaliteiten dan andere databases
	Slechts 1 grote release per jaar

Tabel 6: De voor-en nadelen van MySQL

\* Open source is hier zowel een voordeel als een nadeel. Open source is gratis en wordt vaak voorzien van de nieuwste technieken. Het nadeel hieraan is, is dat het wordt onderhouden door vrijwilligers. Als deze groep kleiner wordt, wordt het systeem niet meer goed onderhouden. Wees kritische als je kiest voor Open Source, een slechte ontwikkeling kan het hele systeem onveilig of instabiel maken.



MSSQL: Microsoft SQL Server is een van de drie mark leidende database technologieën, samen met Oracle Database (MySQL) en IBM's DB2. Microsoft SQL Server is gebouwd bovenop SQL.

Farrah en Shah onderzochten de voor-en nadelen van deze database. Deze data is verwerkt in tabel 22.

Voordelen	Nadelen
Gebruiksgemak	Ecosysteem van Microsoft kan alleen deze SQL-server gebruiken
Beschikt over uitstekende hulpprogramma's	Microsoft doet updates pas om de twee jaar
Voldoende online ondersteuning	
Grote hoeveelheden gegevens verwerken	

*Tabel 7: De voor-en nadelen van Microsoft SQL Server*





## F

### *Tijd*

Kijkend naar de tijd die alle drie de opties innemen, neemt het opslaan van data, file-based het minste tijd in beslag. Hierbij hoeft er namelijk alleen een tekst bestand aangemaakt te worden en geen hele online server met database.

### *Hoeveelheid data*

Kijkend naar de beperkte hoeveelheid data die opgeslagen wordt, zou een losse database-engine momenteel overbodig zijn. Door het opslaan in JSON formaat worden eventuele wijzigingen gewaarborgd.

### *Ondersteuning*

Veel technologieën ondersteunen JSON, dus bij een eventuele schaalvergroting kan dit alsnog gemigreerd worden naar een dedicated DB-engine. MySQL is gratis Open Source, waarbij de ondersteuning en onderhoud niet 100% gegarandeerd kan worden.

Om deze redenen is gekozen om bestaande data file-based op te slaan in JSON-formaat.

## G

C# is een taal die gebruikt wordt op verschillende platformen, zowel online als offline. Denk hierbij aan webapplicaties, Windows apps, en smartphone apps. Een taal die breed inzetbaar is en door zijn dynamische opbouw en openheid voor implementatie van ontwikkel patronen, zeer geschikt is om meerdere functionele onderdelen te hebben. .NET is een framework dat een frictieloze implementatie heeft voor C#, dit omdat het door dezelfde ontwikkelaar gebouwd is (Microsoft).

C# is ontwikkeld door Microsoft. Microsoft heeft een integrated development environment (IDE) ontwikkeld: Microsoft Visual Studio. Deze studio biedt ondersteuning voor C# (Visual Studio, z.j.).

Voordelen	Nadelen
Breed inzetbaar (android - ios and windows applicaties)	Veel tijd nodig om volledig te beheersen
Goed werkende integrated development environment	
Snelheid	
Kracht	
Efficiënt	
Visual Studio een krachtige IDE	
Veel documentatie	
Ontwikkeld door Microsoft	

Tabel 8: Voor-en nadelen C#

Bronnenlijst: Visual Studio (z.j) en Quaro (z.j)

Java kan gratis worden gebruikt en werkt op alle platformen. Java is zowel een programmeertaal als een computerplatform. Volgens Java (z.j) is Java snel, betrouwbaar en veilig.

Heller onderzocht in 2016 de verschillende JAVA IDEs: Eclipse, NetBeans en IntelliJ IDEA. Uit het onderzoek bleek IntelliJ de best werkende IDE, deze is vele malen sneller dan de populairste Java IDE: Eclipse. Echter is Eclipse in zijn geheel gratis en heeft IntelliJ een freemium model, waarbij ontwikkelaars op een gegeven moment wel moeten betalen.

Voordelen	Nadelen
Platform onafhankelijk	Langzamer dan C#
Betrouwbaar	Minder krachtige gratis IDE dan C#
Open Source	Open Source
IntelliJ krachtige IDE (wel betaalde service)	
Veel documentatie	
Veilig	

Tabel 9: Voor-en nadelen Java voor Android

Bronnenlijst: Java (z.j) en Quaro (z.j)



Python is ontwikkeld als een makkelijk te lezen taal, met weinig woorden kan veel gezegd worden. Python werkt volgens het zogenaamde duck-typing: het type van een variabele en het type van een argument van een functie hoeft niet worden gedefinieerd. Een Python-programma wordt geschreven met de extensie: py (Nascimento, 2017).

Voordelen	Nadelen
Makkelijker dan C# en Java	Langzamer dan C# en Java
Minder code nodig dan C# en Java	Werkt minder goed op mobiele applicaties
Open Source	Open Source
Veel documentatie	Gelimiteerd database toegang

*Tabel 10: Voor-en nadelen Python*

Bronnenlijst: Nascimento (z.j) en Quaro (z.j)



## H

C# de krachtiger en sneller dan Java en Python. Bij het omzetten van het prototype naar het echte concept, is het van belang dat de schrijftaal krachtig en snel genoeg is. Een minder snelle en krachtige schrijftaal levert een minder goed werkend concept op. Door meteen de krachtigste en snelste schrijftaal te kiezen, voorkom je dat je op later tijdstip de gehele code moet herschrijven. Dit omdat deze schrijftaal al de beste resultaten (snelheid en kracht) oplevert bij het omzetten van prototype naar concept.

Daarnaast beschikt Microsoft over een SDK en een tutorial over gezichtsherkenning geschreven in C#. Om de leercurve klein te houden en niet te lang bezig te zijn met het leren van een nieuwe schrijftaal, is het verstandig om code te gebruiken die Microsoft aanbiedt. Mocht de leercurve voor een bepaalde taal heel klein zijn zou deze in overweging kunnen worden genomen ondanks dat deze niet beschikt over een SDK. In dit geval is hier geen sprake van, vandaar dat C# de beste keuze is.

## I

Volgens Microsoft (z.j) maakt de Microsoft LifeCam HD-300 het eenvoudig om berichten op je netwerk te plaatsen. De TrueColor technologie zorgt automatische voor heldere, kleurrijke videobeelden, onder vrijwel alle lichtomstandigheden. De camera is geoptimaliseerd voor Microsoft Photo Gallery en kost €32,90.

Alternate (z.j) schrijft dat de 16:9 beeldverhouding het mogelijk maakt om bioscoopwaardige video-opnames te maken. De camera bevat een uiterst snelle USB die USB 2.0 ondersteund en heeft een video resolutie van 1280 x 720 pixels.

In 2016 deed Ansaldo onderzoek naar deze werking van deze camera. Uit dit onderzoek kwam naar voren dat elke keer als Ansaldo zijn hoofd bewoog het beeld vervaagd. Zelf de kleinste bewegingen zorgden voor merkbaar vervagen van het beeld.

Voordelen	Nadelen
Kosten €32,90	Beeld vervaagt bij beweging
TrueColor technologie	Film resolutie van 720p
Geoptimaliseerd voor Microsoft Photo Gallery	Foto resolutie van 720p
USB 2.0	
HD	
Geoptimaliseerd voor Windows	

Tabel 11: Voor-en nadelen Microsoft LifeCam HD-3000

Bronnenlijst: Alternate (z.j) en Microsoft (z.j)

Microsoft (z.j) geeft aan dat de LifeCam Studio beschikt over superieure HD-videokwaliteit, met een HD-sensor van 1080p. Met autofocus wordt scherpte gegarandeerd. Daarnaast beschikt de camera over TrueColor technologie en een precisie lens met glaselement. De LifeCam Studio heeft een draaiing van 180 graden voor een absoluut totaalbeeld en beschikt over een groot lenshoek. De prijs van deze camera bedraagt €89,90.

Uit onderzoek (Humphries, z.j.) blijkt dat deze camera de beste camera is uit de LifeCam series. Echter filmt deze camera in 720p, het maken van foto's wordt alleen ondersteund in 1080p. Deze resoluties komen overeen met die van de HD-300. Het beeld wordt wazig op moment van beweging, alleen wordt dit bij deze camera sneller hersteld, dan als bij de Lifecam HD\_3000.



Voordelen	Nadelen
Kosten €89,90	Vervaagt bij beweging
Groot lenshoek	Film resolutie van 720p
TrueColor	
HD	
Geoptimaliseerd voor Windows	
Foto resolutie van 1080p	

Tabel 12: Voor-en nadelen Microsoft LifeCam Studio

Bronnenlijst: Humphries (z.j) en Microsoft (z.j)


De Logitech C922 Pro maakt levendige video's in HD 1080p. Snelle, vloeiende uploads met H.624: een geavanceerde compressietechnologie. De camera beschikt over automatische lichtcorrectie en is geoptimaliseerd voor Windows. Deze camera maakt het mogelijk om te streamen in 1080p of vloeiende 720p bij 60fps. De kosten van deze camera bedragen €119,00 (Logitech, z.j.).

In 2016 onderzocht Devine de werking van de C922. Het is mogelijk om zowel 720p als 1080p te filmen, waarbij beide resoluties over een uitstekende film kwaliteit beschikken. De automatische lichtcorrectie werkt daarnaast ook precies zoals het hoort te werken.

Voordelen	Nadelen
HD	Kosten €119,00
Film en foto resolutie van 1080p	
Automatische lichtcorrectie	
Geoptimaliseerd voor Windows	

Tabel 13: Voor-en nadelen Logitech C922 Pro

Bronnenlijst: Devine (z.j) en Logitech (z.j)



Specialisten distributeur L2Tek brengt een nieuwe reeks USB 3.0 camera's uit. De See\_Cam30, een uit twee panelen vervaardigd camera is volgens E-con Systems (z.j) geoptimaliseerd voor gezichtsdetectie en kost 149,00 pond en alleen verkrijgbaar in England.

Met een 3.4 MP autofocus USE Video Class (UVC) = compatible camera met Varioptics lens, is de SeeCam \_30 een kleurencamera. De standaard UVC-ondersteuningen zijn: regeling van helderheid, witbalans, versterking, digitale zoom, gammacontrole, contrast, scherpst en verzadigingsregeling. De camera wordt eenvoudig bediend met behulp van een generieke camerasoftware stuurprogramma, welke op zowel Microsoft als Linux werkt (L2tek, 2017).

De SeeCam\_30 kleuren vloeistoflens verbruikt minder energie in vergelijking met traditionele op VCM gebaseerde autofocus oplossingen. Er is geen mechanische beweging, zoals bij een VCM motor voor autofocus, dus de levensduur van de lens neemt toe (Prophet, 2017).

Voordelen	Nadelen
USB 3.0	Geen reviews
Geoptimaliseerd voor gezichtsdetectie	Alleen beschikbaar in England
Geoptimaliseerd voor Windows	Kosten \$149,00
	Weinig overige informatie

Tabel 14: Voor-en nadelen SeeCam\_30

Bronnenlijst: Econ Systems (z.j), L2Tek (2017) en Prophet (2017)

J

Tegenwoordig spenderen mensen meer tijd dan ooit kijkend naar schattige dingen op het internet.

- minions op het internet;
- pikachu dansend op YouTube;
- knuffelende hamsters met hoedjes op;
- mickey Mouse;
- hello Kitty;
- miniature M&Ms.



Figure 1: Schattige dingen Bronnen: <http://universalstudios.wikia.com/wiki/Minions>,  
<http://pokemon.wikia.com/wiki/Pikachu>, en [http://vsbattles.wikia.com/wiki/Mickey\\_Mouse\\_\(Composite\)](http://vsbattles.wikia.com/wiki/Mickey_Mouse_(Composite))





De aantrekkingskracht van deze voorbeelden, zijn de schattige kenmerken die ze bevatten. Schattigheid is een bijzonder kracht in de digitale wereld, omdat het iets is dat in snelle, kleine doses, in een gif of foto kan worden geconsumeerd (Swanson , 2016).

Volgens L Glocker et al. (2009) begint schattigheid met baby's . Baby's hebben:

- grote ogen;
- grote hoofden;
- zachte mollige lichamen;
- knoopneuzen.

Baby's hebben een sterk effect op mensen. Uit onderzoek (L Glocker et al., 2012) blijkt dat foto's van schattige baby's de afgifte van dopamine veroorzaken, dezelfde stof die vrijkomt wanneer mensen verliefd worden, seks hebben of drugs gebruiken: het gelukshormoon. In onderzoek (L Glocker et al., 2012) werden foto's van baby's gemanipuleerd om ze minder uitgesproken baby eigenschappen te geven. Mensen die de foto's bekeken beoordelen de baby's met meer baby eigenschappen als schattiger en voelden een sterkere drang om voor hen te zorgen, dan baby's met minder baby eigenschappen. De baby's die als meer schattig werden beschouwd, brachten een positief effect op de gemoedstoestand van de mensen.

## K

Uit het literatuuronderzoek komen de volgende twee filters naar voren:

- Gabor
- grijstint BT 709

Na overleg met Advise is naar voren gekomen dat twee filters niet genoeg zijn. Advise zou graag meerdere relevante filters getest willen hebben. Op deze manier verdiept Advise zich meer in de werking van filters en het effect hiervan op de gezichtsdetectie. Hierbij wilt Advise dat minimaal gekeken wordt naar de filter: helderheid. De reden hierom is dat Advise wilt weten of de helderheid van de afbeelding een rol speelt bij het detecteren van gezichten. Helderheid is namelijk een variabele waarop goed in te spelen is, bij het uitzetten van het prototype. De Logitech C922 beschikt namelijk over licht bijstelling, waarmee de helderheid kan worden beïnvloed (CoolBlue, z.d).

- helderheid

.NET is een framework voor C#. Accord.NET is een .NET framework welk beeldverwerking voor .NET-toepassingen aanbiedt. Dit framework is onderverdeeld in bibliotheken, die bibliotheken zijn verdeeld in drie hoofdfunctionaliteiten (Accord-framework, z.d):

- scientific computing;
- signal and image processing;
- support libraries.

Deze libraries bevatten verschillende image filters (filters voor afbeeldingen). In verband met tijdsplanning worden alleen filters gebruikt die Accord.NET aanbiedt.

Naast het gabor filter, helderheid, en de grijstint BT 709 bevat deze librarie nog twee andere grijstint algoritmes (Accord-framework, z.d):

- grayscale RMY
- grayscale Y

De grijstinten hebben verschillende conversiecoëfficiënten waarden.

Grijstinten	Rood	Groen	Blauw
BT 709	0.2125	0.7154	0.0721
RMY	0.5	0.419	0.081
Y	0.299	0.587	0.114

Tabel 15: conversiecoëfficiënten grijswaarden



Het Gabor filter is gebaseerd op een Gaussian functie; een normale verdeling. Accord.NET biedt het filter: Gaussian sharpen aan. Aangezien het Gabor filter gebaseerd is een normale verdeling (Gaussian), wordt dit filter ook behandeld. Op deze manier kunnen twee filters die gebaseerd zijn op elkaar vergeleken worden.

- gaussian sharpen

Naast het Gaussian sharpen filter, biedt Accord.NET ook een het sharpen filter opzich aan. In overleg met Adwise wordt deze filter ook meegenomen in de testen. Dit, zodat het verschil tussen Gaussian Sharpen en normaal Sharpen hiermee getest kan worden.

- sharpen



## L

Groepsgesprekken Jeroen Kolkman professionele animator en San Mulder student aan het Saxion in Enschede.

Roze en oranje zijn de huisstijlkleuren van Adwise. Jeroen raad aan ondanks dat roze niet duidelijk zichtbaar is in de kleurencirkel toch te gebruiken in combinatie met oranje. Dit geeft een stuk herkenning aan Adwise. De twee andere kleurencombinaties (blauw-oranje en geel-oranje) worden daarnaast ook gebruikt. Dit omdat deze kleuren volgens Jeroen de juiste uitstraling hebben en het juiste effect kunnen bereiken.

Jeroen geeft als advies mee dat de kenmerken van de animatie cliché mogen zijn, als het werkt waarom zou je er dan van afstappen.

Door samen de kenmerken van schattigheid te bekijken komt uit het gesprek de volgende drie animatie naar voren:

- kat
- muis
- kanarie

Met de volgende cliché kenmerken:

- grote ogen
- grote hoofden
- zachte mollige lichamen
- knoopneuzen

Jeroen raad aan om het gezicht zo goed als hetzelfde te houden, waarbij het lichaam wel veranderd. Op deze manier kan gecontroleerd worden welk onderlichaam de respondenten het populairste vinden.



## M

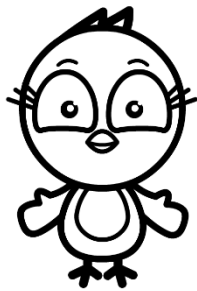


Figure 2: Zwart-wit kanarie



Figure 3: Geel-oranje kanarie



Figure 4: Blauw-oranje kanarie



Figure 5: Roze-oranje kanarie

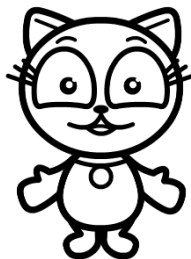


Figure 6: Zwart-wit kat



Figure 7: Geel-oranje kat



Figure 8: Blauw-oranje kat

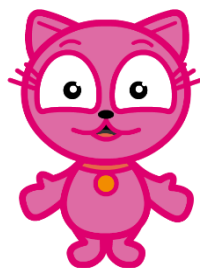


Figure 9: Roze-oranje kat

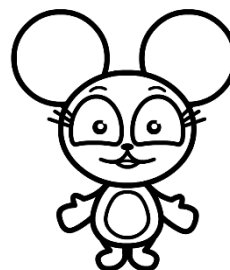


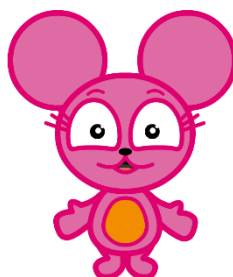
Figure 10: Zwart-wit muis



*Figure 11: Geel-oranje muis*



*Figure 12: Blauw-oranje muis*



*Figure 13: Roze-oranje muis*



# 1

Door *FaceServiceClient* eenmaal als protected variabelen te definiëren in *ValuesController* heeft alleen deze klasse toegang tot deze variabelen. Dit is noodzakelijk omdat de buitenwereld niet zomaar toegang mag krijgen tot deze key. Deze key is namelijk persoonlijk en eventuele kosten zijn verbonden aan deze key. Mochten andere toegang hebben tot deze key, kan deze voor andere doeleinden gebruikt worden en komt de eigenaar van de key met de kosten te zitten. Het is daarnaast ook noodzakelijk dat deze key niet kan worden gemanipuleerd, zonder correcte key kunnen er namelijk geen request worden gedaan op de verschillende API's.



## 2

De class *DetectServices* bevat de methoden om gezichten te detecteren, waarbij bij elk gedetecteerd gezicht de crop methoden wordt aangeroepen. Hierbij worden de coördinaten van het gezicht meegestuurd aan de hand van *FaceRectangle*. *FaceRectangle* is een class welke de get/set methoden: width,height,left, en top van het gedetecteerde gezicht bevat. De crop methoden kan aan de hand van deze coördinaten de afbeelding croppen en opslaan als nieuw .jpg bestand. Op deze manier worden uit afbeeldingen met meerdere personen, elke gedetecteerde gezicht opgeslagen als los bestand. Deze afbeeldingen kunnen worden gebruikt in de class: *CreateAndAddPersonServices*.

De class *CreateAndAddPersonServices* bevat de methoden om een personen groep aan te maken, dit is de groep waarin alle personen worden toegevoegd. Deze groep wordt eenmaal aangemaakt maar wordt wel elke keer meegegeven, zodat de API weet in welke groep gezocht moet worden. Net als de *FaceServiceClient* wordt deze variabelen eenmaal aangemaakt in de *ValuesController*.

```
protected string personGroupld = "adwise";
```

Pas wanneer deze groep is aangemaakt kunnen personen worden toegevoegd aan deze groep. Het aanmaken van een persoon gaat aan de hand van de *CreatePersonAsny* methoden. De klasse *CreatePersonResult* bevat een *Guid* (Globally Unique Identifier) *Personld*. Op dit moment is er een persoon aangemaakt, welke een unique *personld* bevat, welke gekoppeld is aan de *nameOfPerson* in de *personGroup* *Adwise*.

```
CreatePersonResult createPersulResult = await _faceServiceClient.CreatePersonAsny(  
personGroupld, nameOfPerson);
```

Vervolgens moet het referentiemateriaal aan deze persoon worden toegevoegd. Hiervoor is de methoden *DetectAndAddPerson* geschreven. Deze methoden bevat een foreach die elke afbeelding in het meegegeven path afgaat. Vervolgens wordt de *await AddPersonAsync* (*personGroupld*, *createPersonResulat.Personld*, *afbeelding*) aangeroepen. Deze methoden voegt de afbeeldingen uit de opgegeven map toe aan de zojuist gecreëerde persoon.

```
await _faceServiceClient.AddPersonFaceAsnyc(  
    personGroupld, createPersonResult.Personld, img  
);
```





Correct referentiemateriaal kan verkregen worden door afbeeldingen waarbij de betreffende persoon met meerdere personen op staat door de DetectServices te halen. Vervolgens wordt het gedetecteerde hoofd van de betreffende persoon in de juiste map geplaatst. Afbeelding waar alleen het gezicht van de betreffende persoon op staan, hoeven niet eerst door de detectie heen. Hoe meer referentiemateriaal, hoe beter de herkenning.

De class *TrainPersonGroupService* bevat de methoden *TrainPersonGroupAsync*, welke de personen groep traint aan de hand van de opgegeven *peronGroupId*. Elke keer als de *personGroupId* een verandering ondergaat, zoals het toevoegen of verwijderen van een persoon moet de *peronGroupId* opnieuw getraind worden.

```
await _faceServiceClient.TrainPersonGroupAsny(personGroupId);
```

De class *IdentifyServices* detecteert en herkent de gezichten van de ingevoerde afbeelding. Dit wordt gedaan aan de hand van:

```
var faces = await _faceServiceClient.DetectAsync(afbeelding, true, true);
```

Elk gedetecteerde gezicht is nu gekoppeld aan de faces Array. De methode *DetectAsync* bevat *FaceAttributes*, attributes die de detectie aan het gezicht heeft gekoppeld. Bijvoorbeeld, percentage van glimlach en leeftijd. Elke gedetecteerde gezicht bevat een tijdelijke *personId* welke de detectie API aan de hand van *Guid* heeft gecreëerd.

Het koppelen van de *FaceAttributes* aan het zojuist gedetecteerde gezicht gaat aan de hand van een foreach, welke door elke gezicht loopt. Om de *FaceAttributes* te koppelen aan de *faceId* is gebruik gemaakt van een *Dictionary*.

```
Dictionary<Guid, string> faceinfo = new Dictionary<Guid, string>();
```

Vervolgens wordt als key de *faceId* (Guid) en als value *faceAttributes* (string) aan *faceinfo* toegevoegd. Dit wordt gedaan in de foreach, zodat elk gezicht de juiste *FaceAttributes* bevat.

```
faceinfo.Add(face.FaceId, faceAttributes);
```

Vervolgens wordt er gecontroleerd of de zojuist gedetecteerde gezichten overeenkomen met een van de personen in het model. Dit wordt gedaan aan de hand van:

```
IdentifyAsync(string personGroupId, Guid[] faceIds, int MaxNumOfCandidatesReturend = 1);
```

De *faceIds* van de zojuist gedetecteerde gezichten moeten hiervoor eerst in een array geplaatst worden. Dit omdat *IdentifyAsync* een Array van *faceIds* verwacht.



```
var facelds = faces.Select(face => face.faceId).ToArray();
```

```
IdentifyResult results = await _faceServiceClient.IdentifyAsync(personGroupld, facelds, 5);
```

Elk toegevoegde persoon wordt nu vergeleken met de gezichten die zojuist zijn gedetecteerd. Om vervolgens te controleren of er een gezicht is herkend en welk gezicht wordt een foreach geschreven die loopt door de results.

```
foreach( var identifyResults in results){  
    if ( identifyResults.Candidates.Length == 0 ) {  
        // nobody is identified  
    }  
    else {  
        // somebody had been identified  
    }  
}
```

Als er geen mogelijk kandidaten zijn gevonden in de klasse *IdentifyResults* zou deze 0 teruggeven. Er heeft dan geen herkenning plaatsgevonden. Als de lengte niet gelijk aan 0 is, heeft er herkenning plaatsgevonden.

*identifyResults* heeft betrekking op het zojuist gedetecteerde gezicht.  
*identifyResults.Candidates* heeft betrekking op de persoon in het getrainde model.

Op het moment dat er herkenning heeft plaatsgevonden, kan het Id wat bij deze persoon in het model hoort aan de hand van de volgende manier worden opgehaald.

```
var candidateld = identityResult.Candidates[0].PersonId;
```

Op het moment dat het Id van de persoon bekend is, kan deze persoon uit het model worden gehaald.

```
var person = await _faceServiceClient.GetPersonAsync(personGroupld, candidateld);  
var name = person.Name;
```



Figure 14: Personen groep met personen bron: <https://docs.microsoft.com/en-us/azure/cognitive-services/face/face-api-how-to-topics/howtoidentifyfacesinimage>



### 3

De presentatielaag uit fase 2 bestaat uit een *index.cshtml* en een *HomeController.cs*.

De opbouw van de *index.cshtml* is onderverdeeld in een stuk html met daarin de h2, video, canvas en img tags en een groot stuk script.

De script section bestaat uit verschillende methoden.


```
function InitSuccess() {  
    DiffCamEngine.start();  
}  
  
DiffCamEngine.init({  
    video: video,  
    motionCanvas: canvas,  
    initSuccessCallback: initSuccess,  
    initErrorCallback: initError,  
    captureCallback: capture,  
    scoreThreshold: 200  
});
```

Deze methode roept het starten van de geïmplementeerde *diff-cam-engine.js* aan. Hierbij wordt de camera gestart en bij een bepaalde mate van beweging (scoreTreshhold) wordt er een afbeelding gemaakt van het beeld dat op dat moment zichtbaar is uit de camera. Op deze manier worden er niet overmatig veel afbeeldingen gemaakt. Pas op het moment dat er een bepaalde mate van beweging voordoet voor de camera wordt er een afbeelding gemaakt en teruggegeven vanuit de *diff-cam-engine.js*.

Deze afbeelding wordt als *captureCallblack: capture* ontvangen.

Aan de hand van de informatie te vinden op de volgende website is het realiseren van het maken van een afbeelding bij een bepaalde mate van beweging gerealiseerd : <http://diffcam.com/>

Vervolgens is de functie *capture* geschreven. Op het moment dat er een beweging plaatsvindt, wordt een afbeelding gemaakt en teruggegeven. Om serverload te voorkomen en overmatig versturen van afbeeldingen naar de API, is een timestamp gekoppeld aan de afbeelding. Pas wanneer er een bepaalde tijd zit tussen de gemaakte afbeeldingen wordt deze verder werkt. In dit geval om de 2 seconden.



```
var lastMotionTimeStamp = new Date().getTime() / 1000;
```

```
function capture(payload) {  
    var currentTimeStamp = new Date().getTime() / 1000;  
    if (payload.hasMotion && currentTimeStamp - lastMotionTimeStamp > 2) {  
        lastMotionTimeStamp = currentTimeStamp;  
    }  
}
```

De ontvangen afbeelding wordt aan de hand van een Ajax call verstuurd naar *HomeController*. Hiervoor dient de afbeelding die momenteel aan een URL zit gekoppeld eerst als file worden gecomprimeerd. Vervolgens kan deze file gekoppeld worden aan *FormData*, welke verstuurd kan worden aan de hand van de Ajax call.

De *HomeController* staat in verbinding met het model, welke bij een request op het model als response de verkregen informatie uit het JSON-bestand ontvangt. Vervolgens wordt de response via een return teruggestuurd naar de Ajax call uit *index.cshtml*.

Aan de hand van de response komt de begroeting voort.

De functie *rectangels()* is geschreven voor het tonen van een camera waarbij gedetecteerde gezichten omringt zijn met een rood vierkant. Deze functie is geschreven aan de hand van de informatie te vinden op de volgende website: <https://trackingjs.com/>.



## 4

Na overleg met Adwise werd duidelijk dat Adwise wilt dat klanten zien dat er meer data dan alleen het herkennen van gezichten kan worden verschaft met gezichtsherkenningsoftware. Deze data willen ze op een mooie manier als interface laten zien.

Adwise wilt aan de hand van een lijngrafiek de mate van beweging weergeven die plaatsvindt voor de camera. Daarna wilt Adwise dat personen die voor de camera staan getriggerd worden om ervoor te zorgen dat de afbeelding verstuurd wordt. Door het gebruik van een gauge, lijkt het alsof de personen zelf invloed hebben op wanneer een afbeelding wordt verstuurd.

Het registreren van de gemoedstoestand wilt Adwise daarnaast weergeven. Deze gegevens willen ze weergeven in de vorm van een tabel, welke de naam van de herkende persoon met daarbij de mate van de glimlach en het tijdstip weergeeft. Op deze manier kan Adwise in een database bijhouden op welke tijdstip haar collega's het meeste en minste vrolijk zijn, en hierop inspelen.

Ten slot wilt Adwise de begroeting, die afhankelijk is aan persoonlijke gegevens van de voorbijganger in beeld hebben. Op het moment dat deze voorbijganger jarig is of er een andere speciale dag is, wordt de begroeting aangepast aan deze omstandigheden.

Het design is gemaakt door Sarina Meester, UX & UI designer bij Innovadis BV.

## 5

De volgende resultaten zijn verkregen uit het voorafgaand normaliseren met de verschillende filters. Waarbij een X voor detectie staat, bij deze afbeeldingen is er een correcte detectie geweest van alle gezichten aanwezig op de ingevoerde afbeelding.

De vier gebruikte afbeeldingen zonder filter een positieve detectie ondergaan, waarbij alle gezichten op elke afbeelding correct waren gedetecteerd.

Helderheid	Detection 1	Detection 2	Detection 3	Detection 4
0	x	x		x
50	x	x	x	x
100	x		x	x
150			x	x
-50	x	x	x	x
-100	x	x	x	x
-150				x

Tabel 16: Resultaten filter: helderheid

Grijstinten	Detection 1	Detection 2	Detection 3	Detection 4
BT709	x	x	x	x
RMY	x	x	x	x
Y	x	x	x	x

Tabel 17: Resultaten filter: grijstinten



Gabor	Detection 1	Detection 2	Detection 3	Detection 4
Default				
Gamma 0				
Gamma 0.6				
Psi 10				
Psi 12				
Sigma 0				
Sigma 1			x	
Sigma 3				
Sigma 4				
Size 0		x	x	x
Size 6				
Lambda 0				
Lambda 8.0				
Theta 0				
Theta 1.2				
Theta 1.8				

Tabel 18: Resultaten filter: gabor





Sharpen	Detection 1	Detection 2	Detection 3	Detection 4
default	x	x	x	x
Sigma 0	x	x	x	x
Sigma 5	x	x	x	x
Sigma 10	x	x	x	x
Sigma 15	x	x	x	x
Sigma 20	x	x	x	x
Sigma 25	x	x	x	x
SharpenDynamic=true	x	x	x	x

Tabel 19: Resultaten filter: sharpen

## 6

### Welke animatie spreekt je het meeste aan?

Answered: 35 Skipped: 0

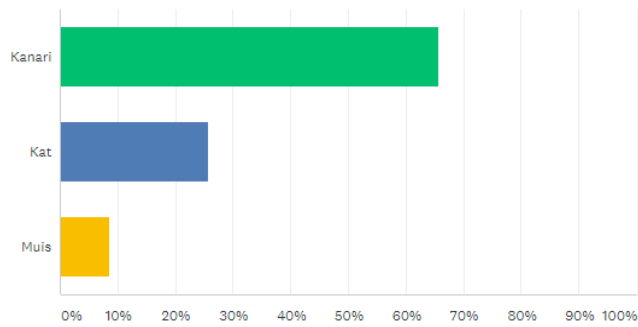


Figure 15: Resultaten keuze animatie

### Welke kanari spreekt je het meeste aan?

Answered: 35 Skipped: 0

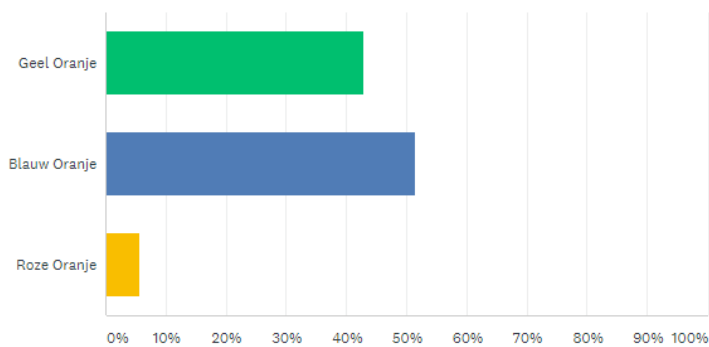
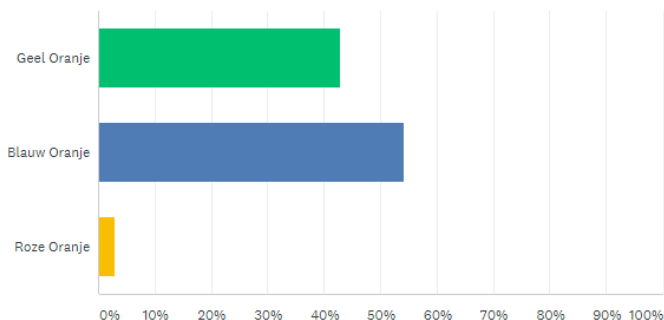


Figure 16: Resultaten kanarie



### Welke kat spreekt je het meeste aan?

Answered: 35 Skipped: 0

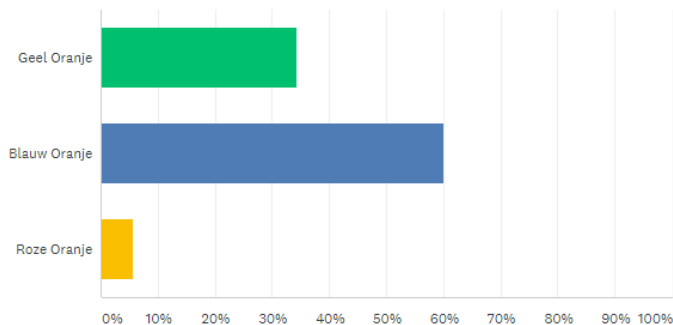


ANSWER CHOICES	RESPONSES
Geel Oranje	42.86%15
Blauw Oranje	54.29%19
Roze Oranje	2.86%1
TOTAL	35

Figure 17: Resultaten kat

### Welke muis spreekt je het meeste aan?

Answered: 35 Skipped: 0



ANSWER CHOICES	RESPONSES
Geel Oranje	34.29%12
Blauw Oranje	60.00%21
Roze Oranje	5.71%2
TOTAL	35

Figure 18: Resultaten muis



Figure 19: Achterzijde respondent camera rechterscherm.

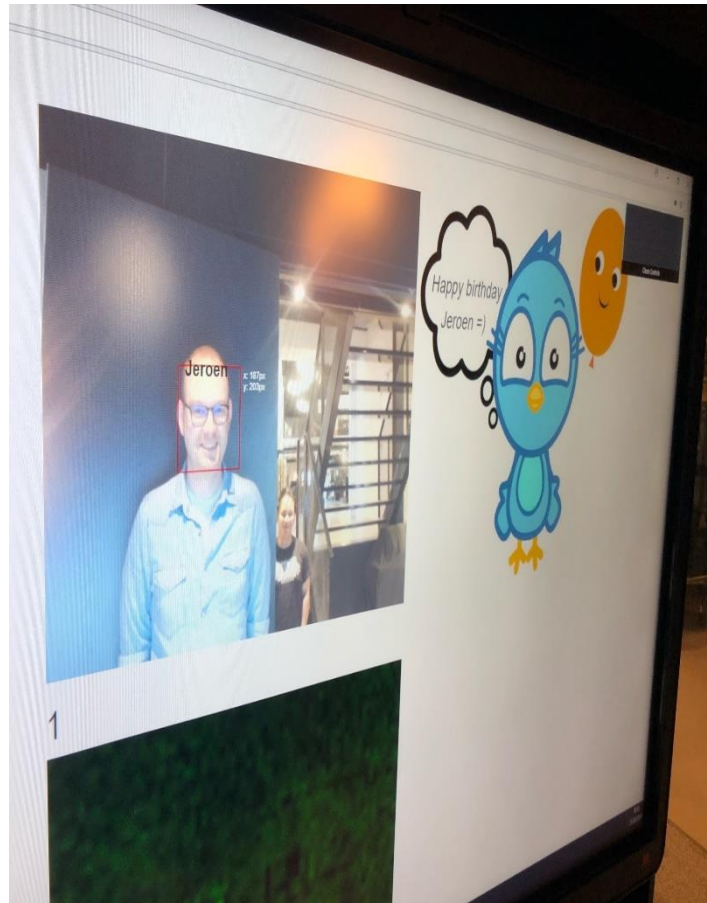
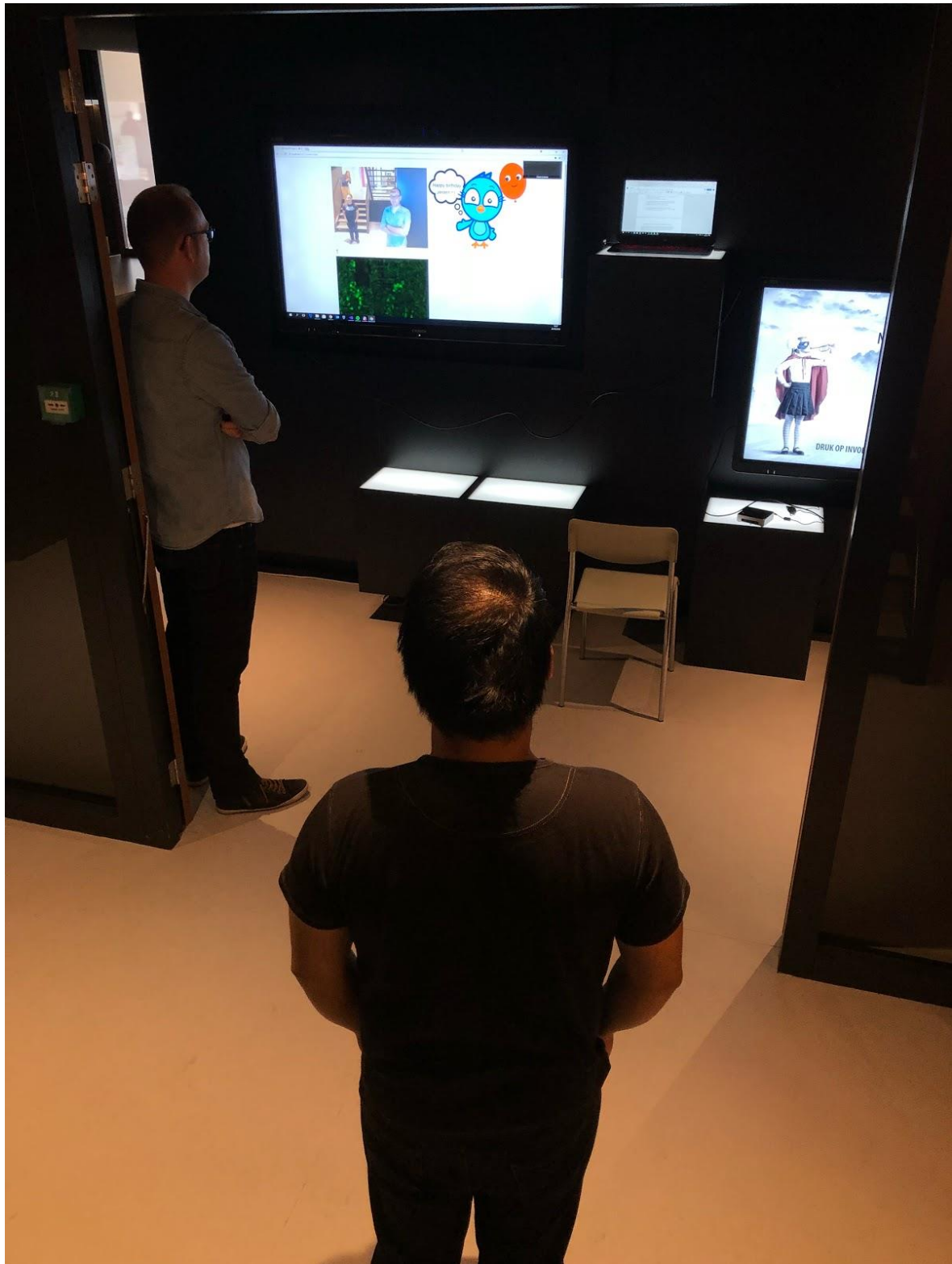


Figure 20: Beeld camera rechterscherm



*Figure 21: Achterzijde respondenten camera linkerscherm*

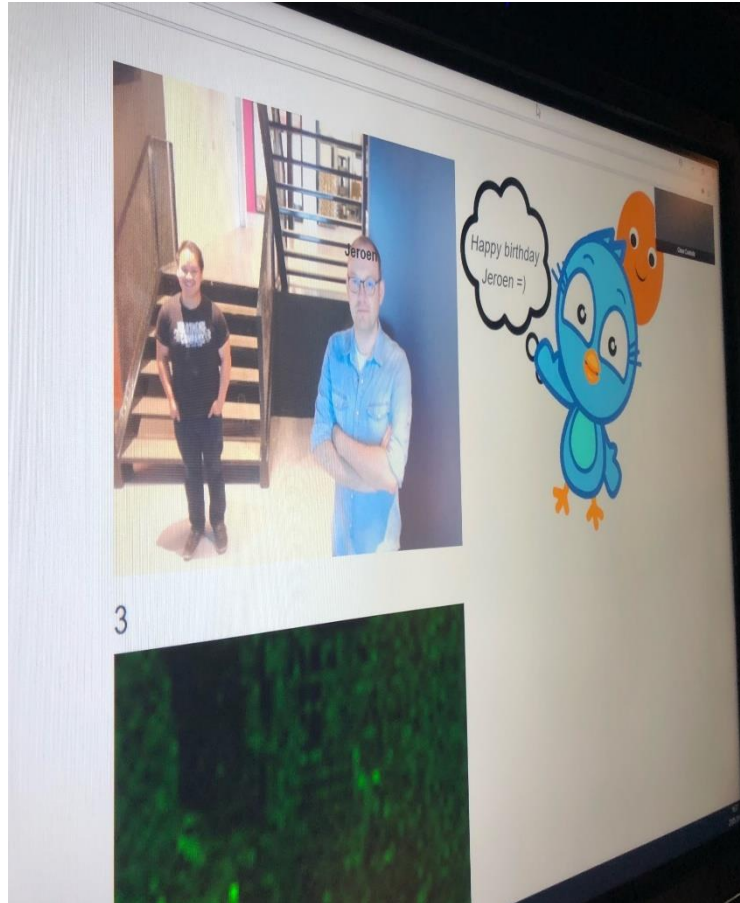


Figure 22: Beeld camera linkerscherf

## 8

In onderstaande tabellen staat een X voor detectie.

Helderheid	100	125	150	175	200	225
Detectie						

Tabel 20: Resultaten plaatsen camera met achtergrondverlichting: 0 trap

Helderheid	100	125	150	175	200	225
Detectie		x	x	x		

Tabel 21: Resultaten plaatsen camera met achtergrondverlichting: 0 deur

Helderheid	100	125	150	175	200	225
Detectie	x	x	x	x	x	x

Tabel 22: Resultaten plaatsen camera met achtergrondverlichting: 0 scherm

Helderheid	100	125	150	175	200	225
Detectie		x	x	x		

Tabel 23: Resultaten plaatsen camera met achtergrondverlichting: 1 trap

Helderheid	100	125	150	175	200	225
Detectie		x	x	x		

Tabel 24: Resultaten plaatsen camera met achtergrondverlichting: 1 deur

Helderheid	100	125	150	175	200	225
Detectie	x	x	x	x	x	x

Tabel 25: Resultaten plaatsen camera met achtergrondverlichting: 1 scherm



## 9

```

▼ (2) [{...}, {...}] ⓘ
  ► 0: {Name: "Jeroen", Birthday: false, Rect: {...}, Smile: 0.864}
  ► 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
    length: 2
  ► __proto__: Array(0)
Motion!
REMOVE
▼ (2) [{...}, {...}] ⓘ
  ► 0: {Name: "Jeroen", Birthday: false, Rect: {...}, Smile: 0}
  ► 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
    length: 2
  ► __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
  ► 0: {Name: "Jeroen", Birthday: false, Rect: {...}, Smile: 0.005}
  ► 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
    length: 2
  ► __proto__: Array(0)
Motion!
null
REMOVE
② Motion!
▼ [{...}] ⓘ
  ► 0: {Name: "Jeroen", Birthday: false, Rect: {...}, Smile: 0}
    length: 1
  ► __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] ⓘ
  ► 0: {Name: "Jeroen", Birthday: false, Rect: {...}, Smile: 0.001}
    length: 1
  ► __proto__: Array(0)
② Motion!
▼ [{...}] ⓘ
  ► 0: {Name: "Jeroen", Birthday: false, Rect: {...}, Smile: 0}
    length: 1
  ► __proto__: Array(0)
REMOVE

```

Figure 23: logs Jeroen / onbekend persoon

```

...
▼ Array(1) 1
  ► 0: {Name: "Lotte", Birthday: false, Rect: {...}, Smile: 1}
    length: 1
    ► __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] 1
  ► 0: {Name: "Lotte", Birthday: false, Rect: {...}, Smile: 1}
    length: 1
    ► __proto__: Array(0)
Motion!
REMOVE
▼ [{...}] 1
  ► 0: {Name: "Lotte", Birthday: false, Rect: {...}, Smile: 0.624}
    length: 1
    ► __proto__: Array(0)
REMOVE
2 Motion!
▼ [{...}] 1
  ► 0: {Name: "Lotte", Birthday: false, Rect: {...}, Smile: 1}
    length: 1
    ► __proto__: Array(0)
REMOVE
2 Motion!
▼ (2) [{...}, {...}] 1
  ► 0: {Name: "Lotte", Birthday: false, Rect: {...}, Smile: 1}
  ► 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
    length: 2
    ► __proto__: Array(0)
Motion!
null
REMOVE
Motion!
null
2 Motion!
▼ [{...}] 1
  ► 0: {Name: "Lotte", Birthday: false, Rect: {...}, Smile: 1}
    length: 1
    ► __proto__: Array(0)
REMOVE
Motion!

```

Figure 24: logs Lotte / onbekend persoon

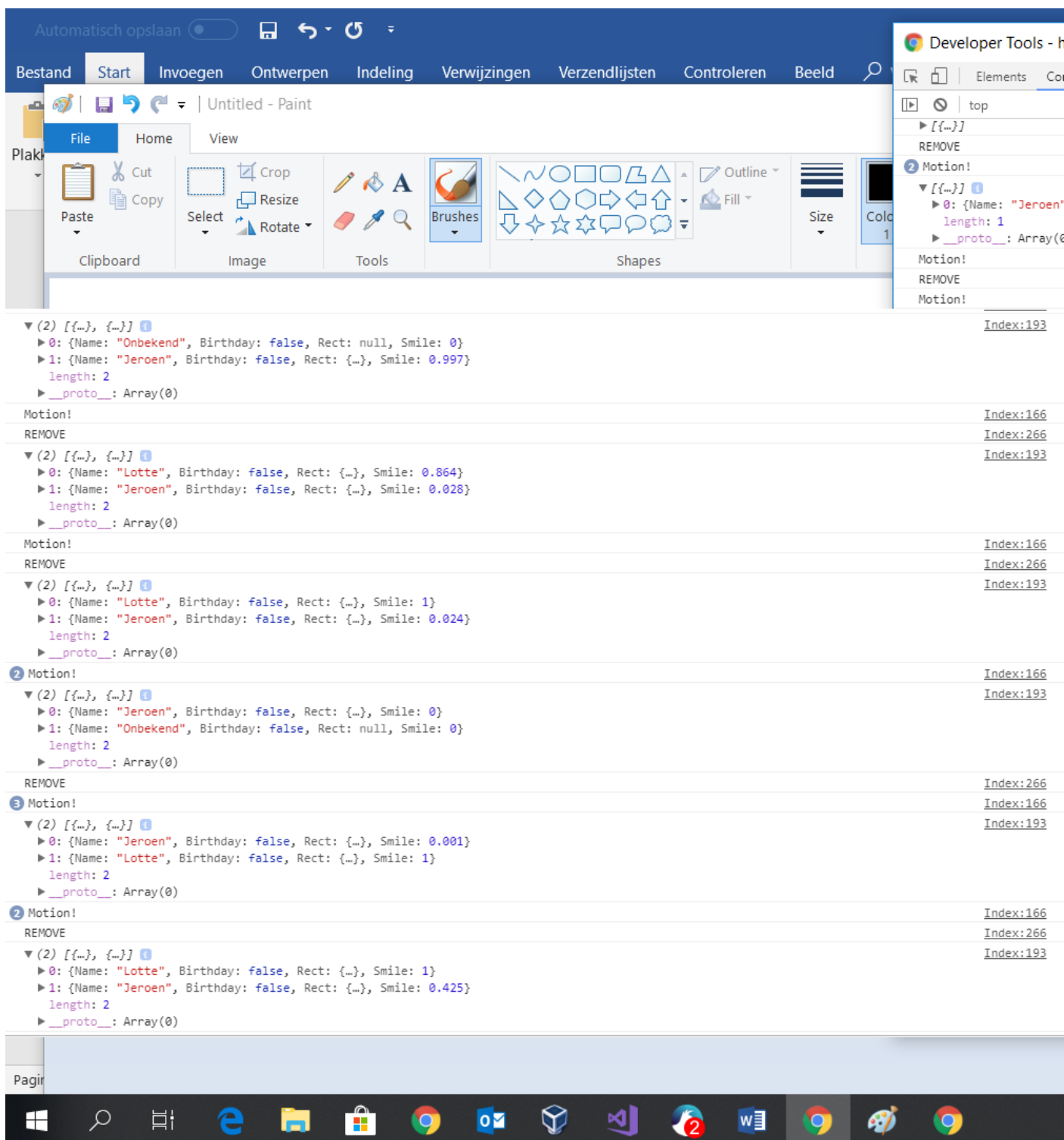


Figure 25: logs Jeroen / Lotte

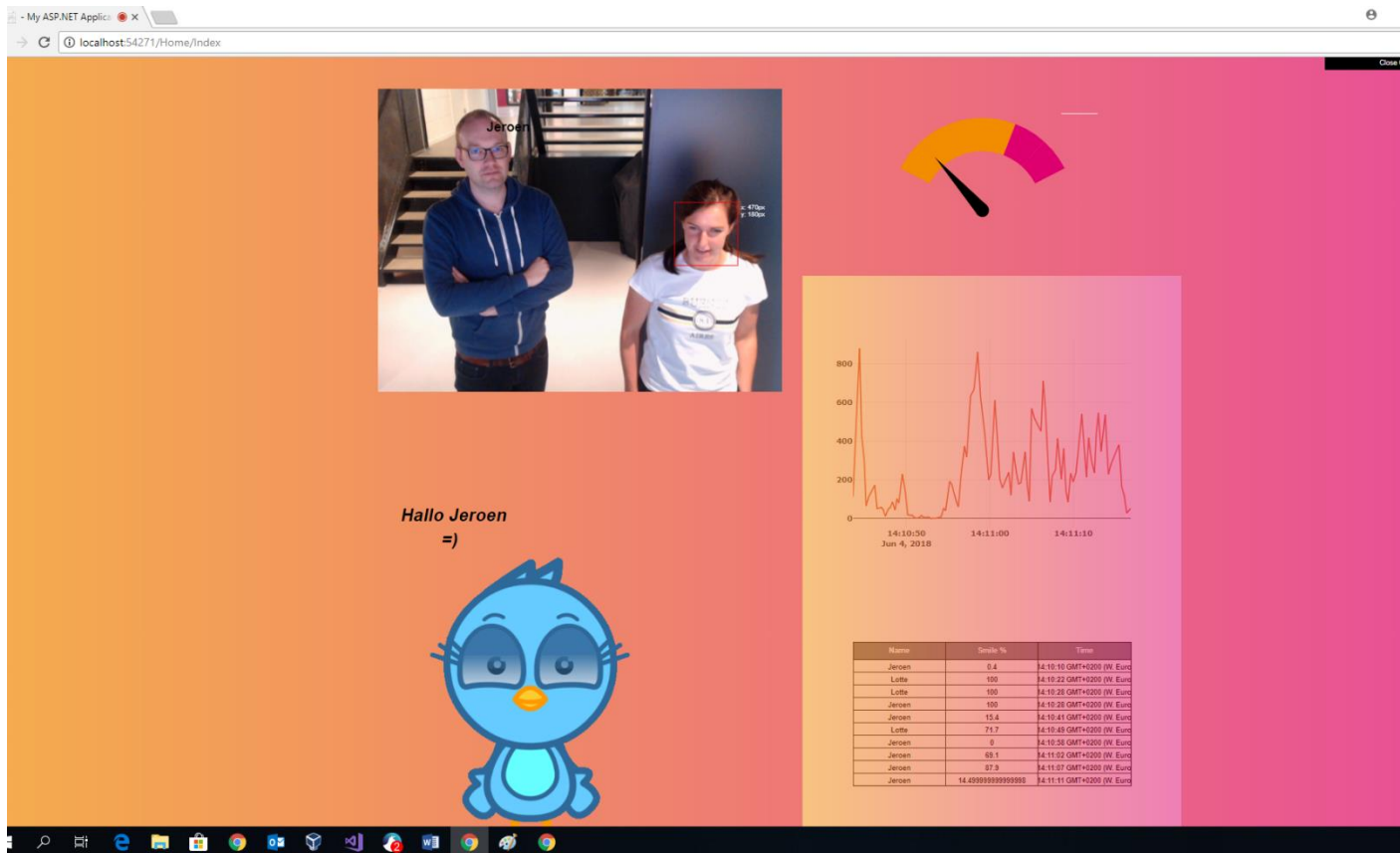


Figure 26: Eerste log resultaat Jeroen / Lotte - Geen herkenning Lotte

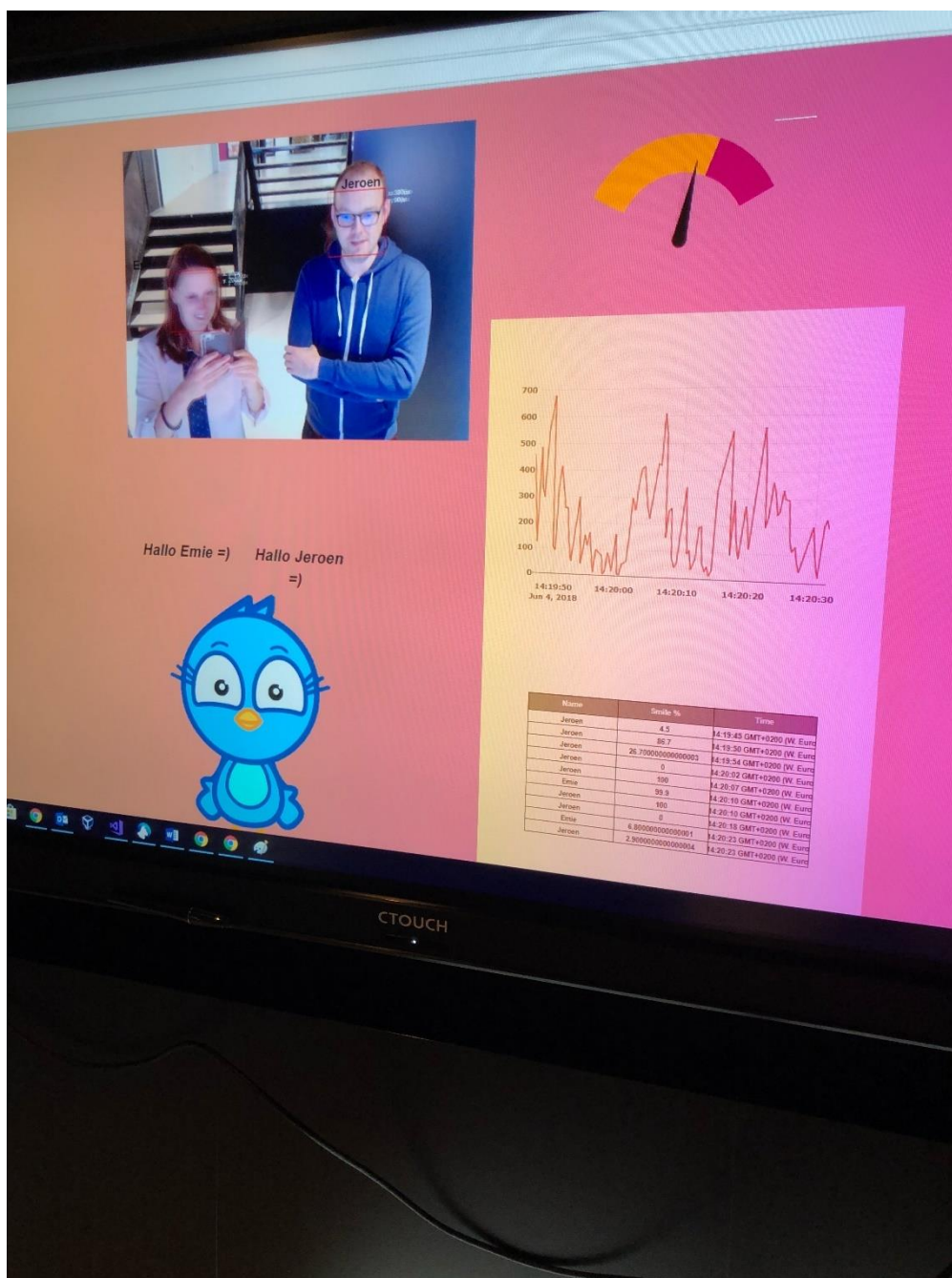


Figure 27: Herkenning Jeroen en Emie



Motion!
▼ [{...}] 1
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{...}] 1
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] 1
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.001}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] 1
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] 1
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.001}
length: 1
▶ __proto__: Array(0)
Motion!
▼ [{...}] 1
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.047}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{...}] 1
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.003}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{...}] 1
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0}
length: 1
▶ __proto__: Array(0)
REMOVE

Figure 28: Logs Emie





```
▼ (2) [{...}, {...}] ⓘ  
  ▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0}  
  ▶ 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}  
    length: 2  
  ▶ __proto__: Array(0)  
REMOVE  
2 Motion!  
▼ (2) [{...}, {...}] ⓘ  
  ▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.377}  
  ▶ 1: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.002}  
    length: 2  
  ▶ __proto__: Array(0)  
Motion!  
REMOVE  
Motion!  
▼ (2) [{...}, {...}] ⓘ  
  ▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.002}  
  ▶ 1: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.003}  
    length: 2  
  ▶ __proto__: Array(0)  
REMOVE  
2 Motion!  
▼ [{...}] ⓘ  
  ▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 1}  
    length: 1  
  ▶ __proto__: Array(0)  
REMOVE  
2 Motion!  
▼ (2) [{...}, {...}] ⓘ  
  ▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0}  
  ▶ 1: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.004}  
    length: 2  
  ▶ __proto__: Array(0)
```

Figure 29: Logs Emie / Ken




▼ [{...}] ⓘ
▶ 0: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.026}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{...}] ⓘ
▶ 0: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.979}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{...}] ⓘ
▶ 0: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.634}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{...}] ⓘ
▶ 0: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.204}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.27}
▶ 1: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.27}

Figure 30: Logs Ken







```

▼ (2) [{...}, {...}] ⓘ
  ► 0: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.27}
  ► 1: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 0.669}
    length: 2
  ► __proto__: Array(0)
Motion!
REMOVE
▼ (2) [{...}, {...}] ⓘ
  ► 0: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.005}
  ► 1: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 1}
    length: 2
  ► __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] ⓘ
  ► 0: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.942}
    length: 1
  ► __proto__: Array(0)
REMOVE
2 Motion!
▼ [{...}] ⓘ
  ► 0: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 1}
    length: 1
  ► __proto__: Array(0)

```

Figure 31: Logs Ken-Bobby samen en apart

```

length: 1
  ▶ __proto__: Array(0)
Motion!
REMOVE
▼ (2) [{...}, {...}] ⓘ
  ▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.973}
  ▶ 1: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 1}
    length: 2
    ▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] ⓘ
  ▶ 0: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 0.814}
    length: 1
    ▶ __proto__: Array(0)
REMOVE
2 Motion!
▼ [{...}] ⓘ
  ▶ 0: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 0.588}
    length: 1
    ▶ __proto__: Array(0)
Motion!
REMOVE
▼ (2) [{...}, {...}] ⓘ
  ▶ 0: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 0.744}
  ▶ 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
    length: 2
    ▶ __proto__: Array(0)
Motion!
REMOVE
▼ (2) [{...}, {...}] ⓘ
  ▶ 0: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 1}
  ▶ 1: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0.012}
    length: 2
    ▶ __proto__: Array(0)
2 Motion!
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
  ▶ 0: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 0.269}
  ▶ 1: {Name: "Ken", Birthday: false, Rect: {...}, Smile: 0}
    length: 2
    ▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
  ▶ 0: {Name: "Bobby", Birthday: false, Rect: {...}, Smile: 0.997}

```

Figure 32: Logs Bobby



52





▼ [{}]
▶ 0: {Name: "Britt", Birthday: false, Rect: {}, Smile: 1}
length: 1
▶ __proto__: Array(0)
Motion!
▼ [{}]
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ [{}]
▶ 0: {Name: "Britt", Birthday: false, Rect: {}, Smile: 0.007}
length: 1
▶ __proto__: Array(0)
Motion!
▼ [{}]
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ [{}]
▶ 0: {Name: "Britt", Birthday: false, Rect: {}, Smile: 1}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{}]
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{}]
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
▼ [{}]
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
null
REMOVE
2 Motion!
▼ [{}]
▶ 0: {Name: "Britt", Birthday: false, Rect: {}, Smile: 1}
length: 1
▶ __proto__: Array(0)
Motion!
▼ [{}]
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!

Figure 34: Logs Britt





▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 1}
length: 2
▶ __proto__: Array(0)
REMOVE
2 Motion!
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Britt", Birthday: false, Rect: {...}, Smile: 0.226}
▶ 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 2
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.007}
length: 2
▶ __proto__: Array(0)
Motion!
REMOVE
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.495}
length: 2
▶ __proto__: Array(0)
REMOVE
2 Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.751}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 1}
length: 2
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Britt", Birthday: false, Rect: {...}, Smile: 0.989}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0}
length: 2
▶ __proto__: Array(0)

Figure 35: Logs Britt / Emie





▼ [{}]
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ [{}]
▶ 0: {Name: "San", Birthday: false, Rect: {}, Smile: 0.008}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{}]
▶ 0: {Name: "San", Birthday: false, Rect: {}, Smile: 0.465}
length: 1
▶ __proto__: Array(0)
Motion!
▼ [{}]
▶ 0: {Name: "San", Birthday: false, Rect: {}, Smile: 0.151}
length: 1
▶ __proto__: Array(0)
REMOVE
2 Motion!
▼ [{}]
▶ 0: {Name: "San", Birthday: false, Rect: {}, Smile: 0.955}
length: 1
▶ __proto__: Array(0)
REMOVE
2 Motion!
▼ (2) [ {}, {}]
▶ 0: {Name: "San", Birthday: false, Rect: {}, Smile: 0.574}
▶ 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 2
▶ __proto__: Array(0)
REMOVE
Motion!
▼ (2) [ {}, {}]
▶ 0: {Name: "San", Birthday: false, Rect: {}, Smile: 0.248}
▶ 1: {Name: "Emie", Birthday: false, Rect: {}, Smile: 0.996}
length: 2
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{}]
▶ 0: {Name: "San", Birthday: false, Rect: {}, Smile: 1}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{}]
▶ 0: {Name: "San", Birthday: false, Rect: {}, Smile: 0.015}
length: 1
▶ __proto__: Array(0)

Figure 36: Logs San



▼ [{...}] ⓘ	▶ 0: {Name: "San", Birthday: false, Rect: {...}, Smile: 0.932}
	length: 1
	▶ __proto__: Array(0)
REMOVE	
Motion!	
▼ (2) [{...}, {...}] ⓘ	▶ 0: {Name: "San", Birthday: false, Rect: {...}, Smile: 0.168}
	▶ 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
	length: 2
	▶ __proto__: Array(0)
REMOVE	
Motion!	
▼ [{...}] ⓘ	▶ 0: {Name: "San", Birthday: false, Rect: {...}, Smile: 0.999}
	length: 1
	▶ __proto__: Array(0)
REMOVE	
Motion!	
▼ (2) [{...}, {...}] ⓘ	▶ 0: {Name: "San", Birthday: false, Rect: {...}, Smile: 0.994}
	▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.758}
	length: 2
	▶ __proto__: Array(0)
Motion!	
REMOVE	
▼ (2) [{...}, {...}] ⓘ	▶ 0: {Name: "San", Birthday: false, Rect: {...}, Smile: 1}
	▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 1}
	length: 2
	▶ __proto__: Array(0)
REMOVE	
2 Motion!	
▼ (2) [{...}, {...}] ⓘ	▶ 0: {Name: "San", Birthday: false, Rect: {...}, Smile: 1}
	▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.893}
	length: 2
	▶ __proto__: Array(0)
2 Motion!	
REMOVE	
▼ (2) [{...}, {...}] ⓘ	▶ 0: {Name: "San", Birthday: false, Rect: {...}, Smile: 0.999}
	▶ 1: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
	length: 2
	▶ __proto__: Array(0)

Figure 37: Logs San - Emie en onbekend



REMOVE
▼ Array(1) ⓘ ▶ 0: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0.958} length: 1 ▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ Array(1) ⓘ ▶ 0: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0.999} length: 1 ▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ Array(1) ⓘ ▶ 0: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0.001} length: 1 ▶ __proto__: Array(0)
Motion!
REMOVE
② Motion!
▼ Array(2) ⓘ ▶ 0: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0} ▶ 1: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 0.999} length: 2 ▶ __proto__: Array(0)
② Motion!
REMOVE
▼ Array(1) ⓘ ▶ 0: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 1} length: 1 ▶ __proto__: Array(0)
② Motion!
REMOVE
▼ Array(1) ⓘ ▶ 0: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0.994} length: 1 ▶ __proto__: Array(0)
② Motion!
REMOVE
▼ Array(1) ⓘ ▶ 0: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 1} length: 1 ▶ __proto__: Array(0)
Motion!

Figure 38: logs Maartje





▼ [{...}] ⓘ
▶ 0: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 0.041}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 0.001}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
null
Motion!
null
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Onbekend", Birthday: false, Rect: null, Smile: 0}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
▼ [{...}] ⓘ
▶ 0: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 1}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 0}
length: 1
▶ __proto__: Array(0)
REMOVE

Figure 39: Logs Kim ogen open/dicht

2	Motion!
▼ (2) [{...}, {...}] 1	
▶ 0:	Birthday: false Name: "Kim" ▶ Rect: {Width: 107, Height: 107, Left: 361, Top: 232} Smile: 1 ▶ __proto__: Object
▶ 1: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 1}	length: 2
▶ __proto__: Array(0)	
Motion!	
REMOVE	
2	Motion!
▼ (2) [{...}, {...}] 1	
▶ 0: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 0.569}	
▶ 1: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 1}	length: 2
▶ __proto__: Array(0)	
2	Motion!
REMOVE	
Motion!	
▼ (2) [{...}, {...}] 1	
▶ 0: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 0.947}	
▶ 1: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0.003}	length: 2
▶ __proto__: Array(0)	
Motion!	
REMOVE	
2	Motion!
▼ (2) [{...}, {...}] 1	
▶ 0: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0.307}	
▶ 1: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 1}	length: 2
▶ __proto__: Array(0)	
2	Motion!
REMOVE	
Motion!	
▼ (2) [{...}, {...}] 1	
▶ 0: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 0.004}	
▶ 1: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0.219}	length: 2
▶ __proto__: Array(0)	
Motion!	
REMOVE	
▼ (2) [{...}, {...}] 1	
▶ 0: {Name: "Kim", Birthday: false, Rect: {...}, Smile: 0.124}	
▶ 1: {Name: "Maartje", Birthday: false, Rect: {...}, Smile: 0.004}	length: 2
▶ __proto__: Array(0)	
2	Motion!

Figure 40: Logs Maartje / Kim



▼ [{...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.089}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.565}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.002}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.747}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.999}
length: 1
▶ __proto__: Array(0)
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.004}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.054}
length: 1
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 1}
length: 1
▶ __proto__: Array(0)
REMOVE

Figure 41: Logs Remco





▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 1}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.979}
length: 2
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.663}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.001}
length: 2
▶ __proto__: Array(0)
Motion!
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.001}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.001}
length: 2
▶ __proto__: Array(0)
REMOVE
② Motion!
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.001}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0}
length: 2
▶ __proto__: Array(0)
Motion!
REMOVE
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.001}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.001}
length: 2
▶ __proto__: Array(0)
② Motion!
▼ [{...}] ⓘ
▶ 0: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.006}
length: 1
▶ __proto__: Array(0)
REMOVE
Motion!
▼ (2) [{...}, {...}] ⓘ
▶ 0: {Name: "Remco", Birthday: false, Rect: {...}, Smile: 0.166}
▶ 1: {Name: "Emie", Birthday: false, Rect: {...}, Smile: 0.056}
length: 2
▶ __proto__: Array(0)
REMOVE

Figure 42: Logs Remco / Emie



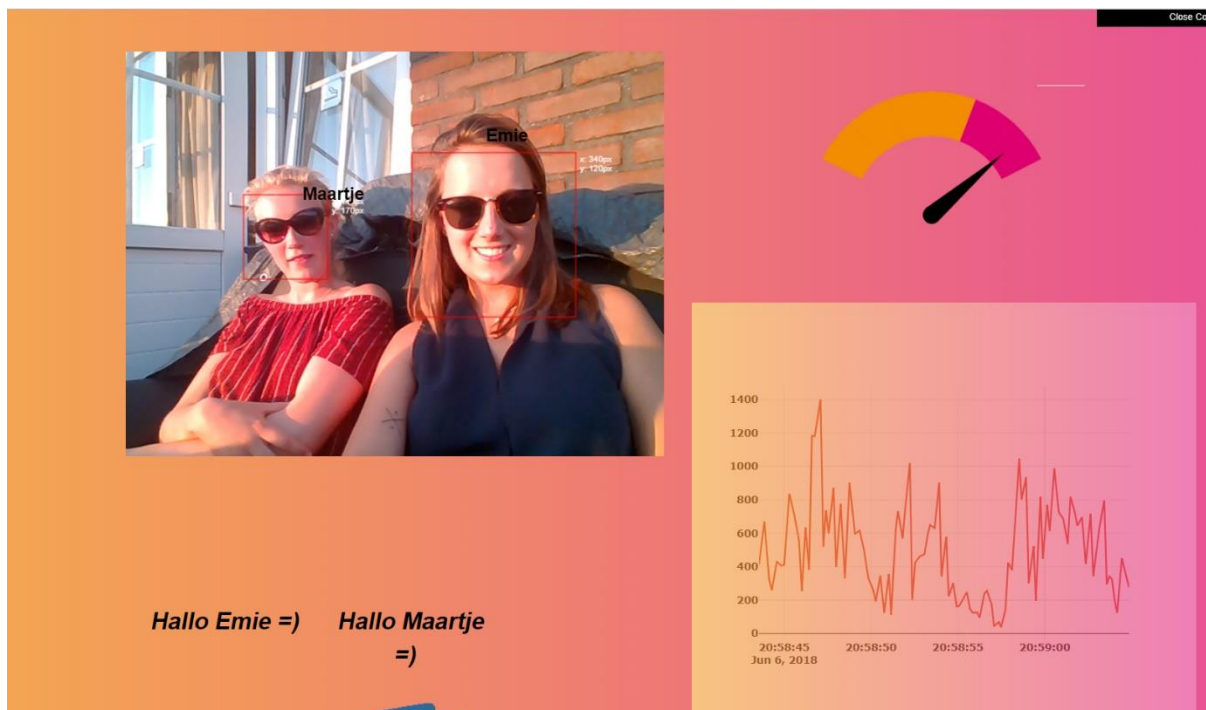


Figure 43: Extra controle Maartje / Emie met zonnebril buiten

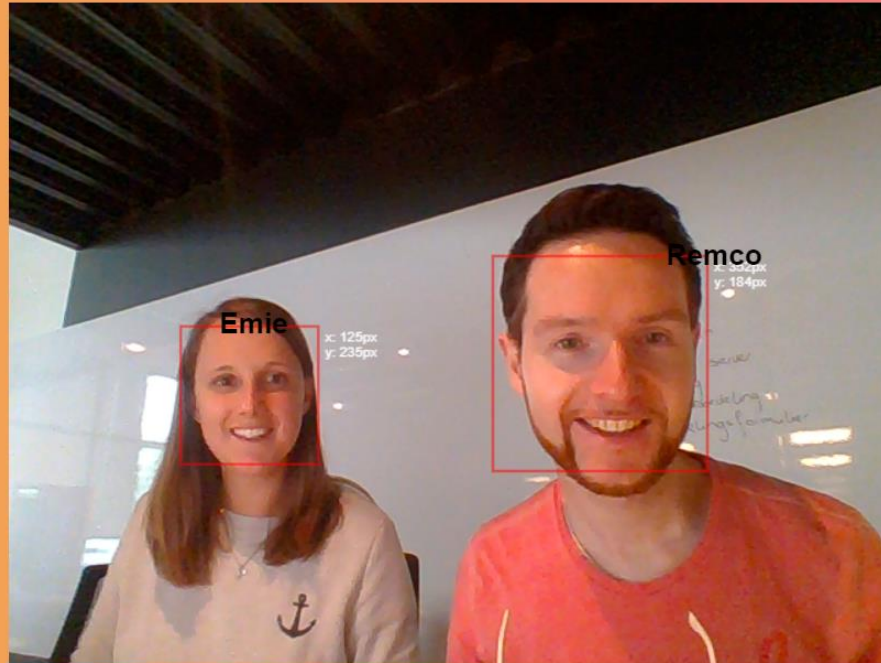


Figure 44: Extra controle Emie / Remco zonder bril andere locatie

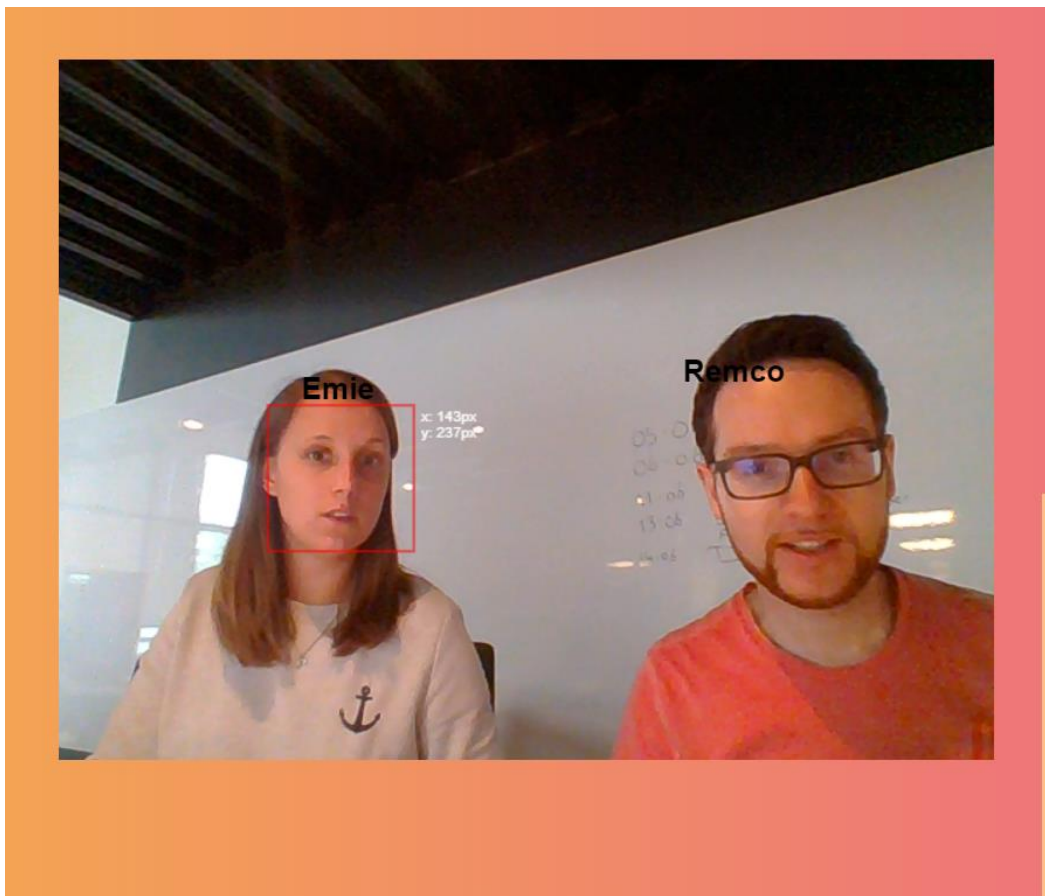
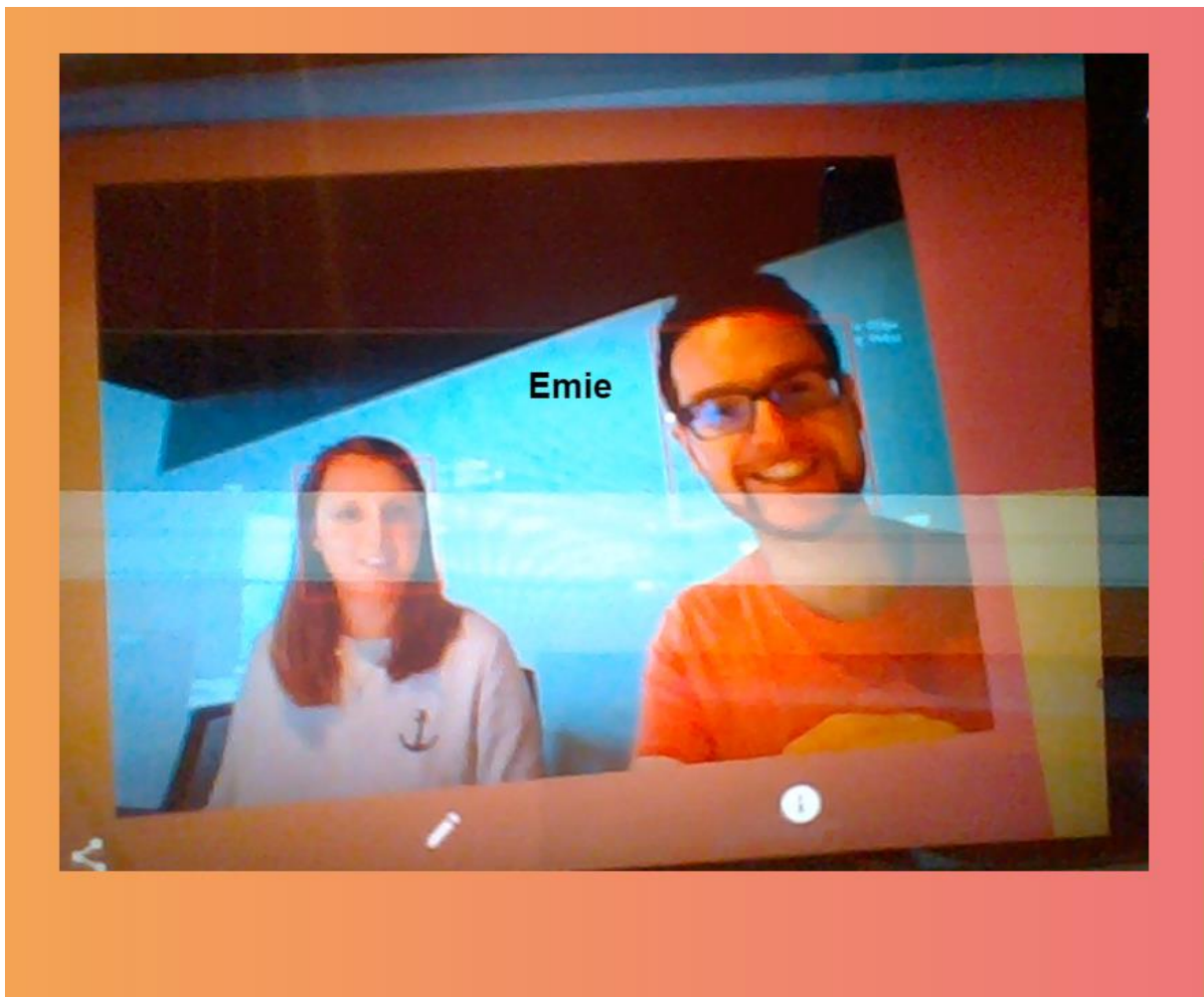


Figure 45: Extra Emie / Remco met bril andere locatie



*Figure 46: Extra Emie / Remco foto op mobiel*





# 10

## Interview 1

### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Ja, de app herkende direct mijn gezicht en toonde mijn naam. De percentages waren voor mij niet direct duidelijk. Na uitleg hoorde ik dat dit gaat over hoeveel je lacht.

### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Ja, ik houd van nieuwe (technologische) dingen dus ik vind het zeker interessant. Wel zou ik het alleen toepassen bij mensen die je (al) goed kent. Dus geen anonieme personen.

### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Prima, het is vriendelijk. Je hebt nog wel het gevoel dat het een computer is, dus in een later stadium zou ik het nog meer uitbreiden.

### Vraag 4:

Wat vind je van de twee manier van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Daar heb ik weinig opgelet 😊 Te veel gefocust op de status van mijn glimlach.

### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Grappig, maar ik zou niet exact weten wat ik hier gelijk mee kan?

### Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

Hoe fit je bent?

### Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Vette applicatie! Zie ik zeker toekomst in. Ik zou het veel testen en continu blijven aanpassen/verbeteren.



Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinder gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Als interactieve paskamer? Voor brillen, kapsels etc.



## Interview 2

### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Ja, het was gelijk duidelijk dat het vogeltje je groet doordat het je gezicht herkent. Het enige waarbij mensen misschien verwarring over kan staan is dat ze misschien nog en spelelement verwachten aan de hand van je bewegingen op camera, maar dat is uiteraard weer een heel andere case dan deze gezichtsherkenningcase.

### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Ja, doordat je op het beeld allerlei metertjes, waarden en het poppetje ziet reageren trekt gelijk de aandacht. Daarnaast is het ook een beetje mens-eigen om te gaan bewegen en "het beeld te gaan spelen", zodra ze zichzelf op live camerabeeld zien.

### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Je zou eventueel ook feestdagen toe kunnen voegen. Of het poppetje op de feestdagen aan kunnen passen. Een pietenmutsje met Sinterklaas, een kerstmuts met kerst, eten bij het suikerfeest. Of – een heel ander idee - motiverende woorden wanneer iemand in de buurt van deadlines komt (gekoppeld aan de agenda van diegene)

### Vraag 4:

Wat vind je van de twee manier van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Het is leuk om de percentages van je lach te zien. Eventueel zou je bij de bewegingsmeter nog een titeltje toe kunnen voegen. Ik wist al waar het voor was, maar misschien dat het voor andere mensen niet gelijk duidelijk is waar het uitslaande pijltje voor is.

### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Ja, het is juist erg leuk om te ontdekken en experimenteren met die waarden. Het laat je juist langer met de applicatie bezig zijn.



Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

Ik denk dat het een coole toevoeging zou zijn als er elementen het beeld in kwam die je bijvoorbeeld met je handen het beeld uit kunt slaan. (Kinect-achtig) Of wanneer je met je handen bijvoorbeeld een brillettje of iets dergelijks op het hoofd van degene die naast je staat kan plakken en die de rest van de dag ook telkens weer voorbij komt als die persoon langsloopt. Net zo lang tot 'ie digitaal weer is afgedaan.

Daarnaast is het goed in te zetten in winkels of bij evenementen. Zo kun je klanten bijvoorbeeld laten springen voor korting door hun beweging o.i.d. te meten.

Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Eventueel zou je ook audio toe kunnen voegen. Er is nu een kans dat iemand voorbij loopt en de tekst op het scherm niet opmerkt. Audio triggert gelijk, zeker als je je naam hoort.

Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinder gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Je zou het bijvoorbeeld kunnen gebruiken om ergens in te checken. Bij een restaurant bijvoorbeeld of het OV. Zodra je binnenloopt wordt je geregistreerd en begeleid een ober je naar je tafel.



### Interview 3

#### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Ja ik zag dat hij mijn hoofd herkende en ik zag mijn naam in beeld.

#### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Ik vond het leuk dat het vogeltje mijn naam zei en als ik niet zou weten dat mijn data gebruikt zou zijn dan zou ik zeker nog meer interesse in de applicatie hebben.

#### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Ja leuk. Misschien nog meer kenmerken wat betrekking heeft op het uiterlijk. Bijvoorbeeld als je lacht, dat de applicatie dan aangeeft dat je een leuke glimlach hebt. Of dat je juist niet zo boos moet kijken wanneer chagrijnige mensen langslopen.

#### Vraag 4:

Wat vind je van de twee manier van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Ik vindt het duidelijk, je kunt aan de meter zien dat er beweging is.

#### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Ja leuk! Het gaf aan dat ik vrolijk was en veel lach. Het zou wel meer interactie roepen wanneer er wat meer mee gedaan zou worden(bijvoorbeeld dus dat het vogeltje wat zegt over je emotie).

#### Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

Misschien als er een stem bij zou zijn die daadwerkelijk je naam roept. En dus iets meer interactie op het gebied van wat het vogeltje zegt. Maar ik vond het al wel een leuke applicatie en interactie.



Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Nee, die heb ik hierboven al benoemt.

Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinder gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Je kan het ook echt gaan verkopen als product. Dus bijvoorbeeld bij Greetz, dat je gepersonaliseerde verjaardagscadeaus maakt op basis van gezichtsherkenning. Dat als je een verjaardagskaart opent, of een doos chocola opent, je naam wordt genoemd en er bijvoorbeeld een gepersonaliseerde boodschap wordt afgespeeld op het moment dat je gezicht wordt herkent. Dat is dus meer dat je een gepersonaliseerd cadeau kunt maken als consument.



#### Interview 4

##### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Ja en nee. In het begin had ik niet door waar de gauge voor was, pas nadat het werkt ugelegd werd mij dit meer duidelijk. De rest van de applicatie was wel meteen duidelijk voor mij.

##### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Ja, bewegende dingen op schermen trekken mijn aandacht. Daarnaast houdt ik van nieuwe technologieën.

##### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Vrolijk heel vrolijk. Het kan aangepast worden op eventuele feestdagen per jaar zoals pasen, waarbij er een paas animatie wordt getoond.

##### Vraag 4:

Wat vind je van de twee manier van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Het zou fijn zijn als er in de interface komt te staan waarvoor het precies dient. Dan is het meer een uitdaging om een bepaalde score van beweging te halen.

##### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Leuk, ik ging meteen kijken of het werkte of niet door boos en blij te kijken.

##### Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

Het poppetje zou als ware op de mate van beweging kunnen reageren, of dat de begroeting als spraak wordt gedaan.

##### Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Ik zou de interface iets anders maken, duidelijker.



Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinder gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

In winkels betalen met je gezicht, openen van gesloten deuren aan de hand van gezichtsherkenning software.





## Interview 5

### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Ja

### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Ja, dat komt doordat het kruist met mijn vakgebied.

### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Ja, meer notificaties, agenda afspraken, weers voorspellingen, interne wist-je-datjes, nieuws berichten.

### Vraag 4:

Wat vind je van de twee manier van weergeven van de mate van beweging, van de interactie die er is met de applicatie?



### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Cijfer is onduidelijk, zou eerder 3 / 4 categorieën willen zien met bepaalde emojis.

### Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

Ja, game op je verjaardag om cadeau's te verdienen.

### Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Meer interactie, meer persoonlijke info op t scherm.

### Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinden gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Deur ontgrendeling, aanwezigheidscheck.



## Interview 6

### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Jazeker, het heeft weinig uitleg nodig.

### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Ja, vooral als je de functie er van weet ga je het uitproberen.

### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Misschien een leuke verwelkoming.

### Vraag 4:

Wat vind je van de twee manieren van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Cool en interessant.

### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Leuke bijkomstigheid. Dan kan je mensen extra motiveren vaker te lachen als dat het geval is.

### Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

Misschien een soort van Siri of Alexa mee gecombineerd. Maar dat is misschien te ambitieus.

### Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Zou super gaaf zijn als dit daadwerkelijk ook verder ontwikkeld gaat worden tot een echt product.

### Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinder gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Niet zo direct nee.



## Interview 7

### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

De Applicatie is absoluut duidelijk zodra je er voor het eerst mee in aanraking komt. Je wordt gelijk begroet door een leuke animatie en je ziet dat je herkend word.

### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Zeker weten. Het is aantrekkelijk, omdat je bewegende beelden ziet en hier ook direct ziet dat er iets gebeurd als je langs loopt. Dit trekt mensen aan!

### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Misschien dat het voor een latere versie een goeie toevoeging zou zijn, om goedemorgen, middag en avond te zeggen aan de hand van de tijd. Dan is er wat meer variatie in de begroetingen.

### Vraag 4:

Wat vind je van de twee manieren van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Het is leuk om het terug te zien in de grafieken en de snelheidsmeter. Door de beweging wordt het aantrekkelijker om te doen en ben je toch sneller aangetrokken.

### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Ik vind dit een super toevoeging in de applicatie, vooral omdat dit later ook gebruikt zou kunnen worden voor analyses. Denk hierbij aan een tevredenheidsonderzoek o.i.d.

### Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

Zelf ben ik fan van een "cleane" applicatie en vind ik het zoals het nu is juist perfect.



Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Misschien is het voor een latere versie een mooie toevoeging om wanneer je iemand herkent, hier een foto van te maken en deze toe te voegen in het proces, zodat je automatisch meer data verzamelt.

Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinden gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Zoals in vraag 5 benoemd, lijkt het mij heel interessant om dit in te zetten voor tevredenheidsonderzoeken of om dit te gebruiken voor analyses bij winkels: "wanneer kopen mensen sneller iets?"



## Interview 8

### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Ja helemaal, alles sprak voor zich.

### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Ja, want het ziet er interessant uit. Ook leuk dat je hoeveelheid lach kan herkennen.

### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Ja heel leuk, koppeling aan agenda, animatie aanpassen aan feestdagen.

### Vraag 4:

Wat vind je van de twee manier van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Goed, wordt genoeg mee gedaan.

### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Heel leuk, dat maakt het stuk interessant.

### Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

Genoeg interactie moet niet te veel worden.

### Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Vind ik lastig, dus zou het niet weten.

### Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinder gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Interactie met spelletjes.



## Interview 9

### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Niet meteen, titels en uitleg zouden de applicatie duidelijker maken.

### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

Ja, een camera waarop je jezelf ziet is altijd interessant. Daarnaast bewegen er veel dingen op de interface, dat is iets wat mijn aandacht trekt.

### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Leuk, zou alleen leuker zijn als die echt alleen reageert op het moment dat je herkent wordt. Agenda koppeling, aanpassen aan feestdagen.

### Vraag 4:

Wat vind je van de twee manier van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Leuk, alleen was bij de gauge niet helemaal duidelijk wat deze deed. Een titel/omschrijving zou fijner zijn geweest.

### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Dat maakt het wel of je wilt kijken of het werkt of niet, het triggert om boos en blij te gaan kijken. Alleen zijn de nummers beetje saai, misschien dat je daar wat anders op kan verzinnen.

### Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

De interactie is voor een prototype genoeg, het moet niet te veel worden.

### Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

Ik zou de interface mooier maken. Wat er nu staat kan, maar als je er even dagje aan gaat zitten kan je er zoveel meer uithalen.



Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinder gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Gepersonaliseerde reclame.



## Interview 10

### Vraag 1:

Was de applicatie duidelijk zonder uitleg van de onderzoeker?

Ik heb al wat ervaring met dit concept, maar zonder die ervaring is het duidelijk dat hij jou moet zien en jou ziet wanneer hij een hokje plaats en herkend als er een naam bij komt te staan.

### Vraag 2:

Mocht u deze applicatie tegenkomen, wordt je dan getriggerd om erna toe te gaan? Wekt de applicatie interesse op?

De meeste mensen worden aangetrokken door dit soort apps, vooral als ze wat extra's tonen, een hokje om het hoofd of een 3D hoedje op het hoofd.  
– leuk grappig en herkenbaar (jij bent het)

### Vraag 3:

Wat vind je van de begroeting van de applicatie? Zou je nog andere begroetingen willen zien dan hallo en gefeliciteerd met je verjaardag, wanneer het je verjaardag is?

Hoe meer hoe beter als het maar duidelijk is wat de app toont, een voorbeeld uit je app, de 'smile' detectie en hoeveel.

### Vraag 4:

Wat vind je van de twee manier van weergeven van de mate van beweging, van de interactie die er is met de applicatie?

Daar heb ik niet zoveel naar gekeken, ik wist dat die onderdelen er waren, maar hier kijk je eigenlijk niet naar wanneer je gebruikt van de app, evt. een icoontje met waardes over de video plaatsen, zodat het meer opvalt, maar evt. minder uitgebreid kan zijn.

### Vraag 5:

Wat vind je van de tabel die de mate van je glimlach weergeeft? Nodigt dit uit om meer interactie met de applicatie hebben om verschillende waardes te krijgen, of voegt het weinig toe?

Geeft leuke resultaten.





Vraag 6:

Zou je nog meer vormen van interactie willen zien? Zo ja, welke?

2D balletjes wegslaan als je er met hoofd tegenaan komt of random pruiken of ander soortige lenzen

Vraag 7:

Heeft u nog enige tips en/of aanmerking met betrekking op de applicatie?

De basis lijkt goed te werken, door enkele bekende beperkingen laat hij niet direct de naam zien, dit zou wel leuk zijn (snellere response).

Vraag 8:

De data die verzameld wordt met deze applicatie kan voor verschillende doeleinder gebruikt worden. Als voorbeeld "Gepersonaliseerde reclame". Heeft u nog andere ideeën waarvoor de verzamelde data gebruikt kan worden?

Checken in welke ruimte een bepaald persoon bevind, dit d.m.v. meerdere van dit soort camera stacks, algemene humeur checker en de verandering daar in, wanneer een persoon dagelijks op de foto komt.