

# Technische documentatie interface

## Inhoud

- prototype ontwikkeling
- Web technologieën
- Versies en interacties interface
- Bijlages

## Inleiding

In het team zijn er specifieke rolverdelingen gemaakt. Rober en Stefan zijn beide gespecialiseerd in trouble shooting en coderen.

Rober en Stefan hebben zich ontfermt over de technische aspecten tijdens het project. Er is nagedacht over alle diverse technische uitwerkingen en mogelijke complicaties. Ook is er nagedacht over nieuwe en moderne technieken die het concept naar een hoger niveau kunnen tillen.

## Prototype

Het prototype bestaat uit een interface van 7 schermen met als doel om erachter te komen welk mediatype een gebruiker bezit door naar videofragmenten te kijken die de gebruiker leuk vindt.

Voor het bouwen van de interface zijn in totaal drie verschillende technologieën toegepast. Deze technologieën zijn:

- Eye Tracking
- Web Technologieën
- Face Recognition

### Eye Tracking

Om het kijkgedrag van een gebruiker te kunnen meten moesten we eerst een simpele manier uitzoeken om dit in het prototype toe te passen. Daarom hadden we eerst gebruikt gemaakt van

de ingebouwde webcam van een laptop in combinatie met een Google Chrome extensie (xLabs) die het mogelijk maakt om de ogen van een gebruiker te traceren. Deze methode was helaas niet efficiënt, omdat de gebruiker zijn hoofd mee moest bewegen en de bedoeling is dat alleen de ogen hoeven te bewegen voor een optimale kalibratie. Vanwege dit pijnpunt zijn we gaan kijken naar een betere oplossing. Voor een betere en accurate oplossing hebben we gebruikt gemaakt van de Tobii Eye Tracker hardware. Dit is een hardware die je aan een laptop/pc moet aansluiten. Tobii is voorzien van zijn eigen software. Het voordeel van de Tobii Eye Tracker is dat het heel accuraat werkt, maar heeft één nadeel en dat is dat het steeds gekalibreerd moet worden voordat iemand het wilt gebruiken. We hebben geprobeerd om een eigen kalibratie

interface te bouwen, maar om de kalibratie interface van de Tobii aan te kunnen passen, moet je een licentie aanvragen en in het programmeertaal C++ kunnen programmeren. Helaas kan niemand in ons team in C++ programmeren, hierdoor hebben we besloten om de ingebouwde kalibratie interface van de Tobii in ons prototype te gebruiken.

## Web Technologieën

De interface waarin het prototype is gebouwd werd in CSS, HTML en Javascript geprogrammeerd. We hebben voor deze programmeertalen gekozen, omdat Rober en Stefan (developers) bekend zijn met deze web technologieën. Het prototype werkt daarom in de vorm van een website, maar de kern van het prototype kan op andere platforms toegepast worden om er toch een echte en professionele software van te maken.

### Face Recognition

In het prototype maken we gebruik van Face Recognition, zodat de gebruiker weet dat zijn/ haar ogen getraceerd worden. We hebben een Javascript library (clmtrackr) gebruikt die dit mogelijk maakt. Deze Javascript library biedt meerdere mogelijkheden, maar de belangrijkste hiervan was de gezichtsherkenning. Deze methode herkent het gezicht van de gebruiker via de webcam en er worden vervolgens op het scherm groene lijnen

op het gezicht van de gebruiker getoond die daadwerkelijk laat zien dat het gezicht herkend wordt.

### Interface

#### *Versie 1*

De eerste versie van de interface bestond alleen uit één scherm met 12 bewegende gifjes. Op het moment dat er naar één bewegend gifje werd gekeken kreeg dat gifje een rode overlay, zodat de gebruiker weet waar hij/zij naar kijkt.

Deze eerste versie hebben we vervolgens met gebruikers getest en uit de test bleek dat de gebruikers alle bewegende gifjes te chaotisch vonden. De meeste zeiden dat je je niet op één gif kon focussen en dat het beter zou zijn als het gifje pas beweegt op het moment dat ernaar eentje wordt gekeken. De resultaten zijn vervolgens naar de volgende versie meegenomen.

#### *Versie 2*

De tweede versie van de interface bestond ook uit één scherm met 12 schermen, maar met de aangepaste punt van versie één. In deze versie bewogen de gifjes pas op het moment dat er specifiek naar eentje werd gekeken en de rode overlay hebben we ook weggehaald, omdat de gebruiker meteen weet waar hij/ zij naar kijkt op het moment dat een gif beweegt.

Deze versie werd vervolgens getest en uit de resultaten bleek dat de bewegende

gifjes vanaf het begin de gebruiker inderdaad beter laat focussen.

### ***Versie 3***

In versie 3 begonnen we de interface te bouwen zoals het op de wireframe stond. Deze wireframe bestaat uit 7 schermen. In deze versie was het alleen van belang de flow van de interface goed in kaart te brengen, dus functionaliteiten zoals Face Recognition en de logica van de aanbevelingen werden in een latere versie geïmplementeerd.

### ***Versie 4***

In deze versie begonnen we met het implementeren van de Face Recognition in scherm 3. De bedoeling van dit scherm is dat de gebruiker weet dat zijn/ haar ogen getraceerd worden, zodat de tijdmachine bestuurd kan worden. Verder heeft dit scherm geen andere functionaliteiten.

### ***Versie 5***

In deze versie zijn we begonnen aan de aanbevelingen logica die de interface moet uitvoeren op het moment dat de gebruiker naar videofragmenten kijkt. Het uiteindelijke doel van de interface is om de gebruiker aanbevelingen te geven door het mediatype van de gebruiken te bepalen. Op het moment dat het systeem genoeg informatie heeft, wordt er vervolgens berekend welke mediatype het meest bij de gebruiker past.

Deze logica gebeurt allemaal in de backend van de interface, maar wordt nooit visueel aan de gebruiker getoond. De reden voor dit is om de gebruiker nieuwsgierig te maken en door de ticket die op het einde wordt verkregen de gebruiker te motiveren naar het museum te gaan om daar erachter te komen welke archetype de bezoeker bezit. Verder is er in scherm 6 de opmaak veranderd naar een animatie die de gebruiker laat zien dat zijn/ haar data wordt verwerkt om het archetype te bepalen. (zie scherm 6)

### ***Versie 6***

In deze versie zijn de designs van alle schermen aangepast, behalve scherm 5.

We hebben vervolgens de flow van de interface met gebruikers getest en die waren positief. We konden vanaf dat moment focussen op de videofragmenten van scherm 5.

### ***Versie 7***

Nadat de flow van alle schermen goed waren begonnen we te werken aan scherm 5. Scherm 5 is de belangrijkste scherm, omdat de interface het kijkgedrag van de gebruiker moet volgen om zo een archetype voor de bezoeker te bepalen. Dit scherm is in 3 andere schermen/ fases verdeeld, omdat we van de gebruiker willen weten wat hij/ zij in zijn/ haar kindertijd en pubertijd keek en wat hij/ zij tot heden nog kijkt. Op deze manier krijgt de interface meer

informatie over de gebruiker om zo een beter archetype te bepalen.

We mochten we helaas de database van museum Beeld en Geluid niet gebruiken vanwege copyright. Hierdoor hebben we zelf korte videos op youtube gezocht en gedownload om in scherm 5 van de interface te implementeren. We hebben voor het prototype uiteindelijk ervoor gekozen om per fase 4 categorieën te gebruiken en elke categorie met 3 videofragmenten.

Op het moment dat een gebruiker naar een videofragment kijkt begint het af te spelen en na 10 seconden stopt het, verdwijnt het van het scherm, verschijnt een nieuwe video en die wordt vervolgens afgespeeld.

De interface houdt bij op welke categorieën en videos het meest wordt gekeken om later met deze data het archetype te bepalen.

We hebben vervolgens deze 3 schermen met gebruikers getest en uit de test is gebleken dat 10 seconden per video te lang was. Ook kregen we als feedback dat het handig zou zijn om de gebruiker de optie te geven of een video wel of niet leuk is.

### ***Versie 8***

In deze versie hebben we de user experience van de 3 fasen aangepast door de aantal secondes naar 6 te verlagen in plaats van 10 en in deze versie krijgt de gebruiker nu de optie om te kiezen of een video wel of niet leuk is. Deze opties

verschijnt alleen pas op het moment dat de gebruiker 6 seconden naar een videofragment kijkt.

### ***Versie 9***

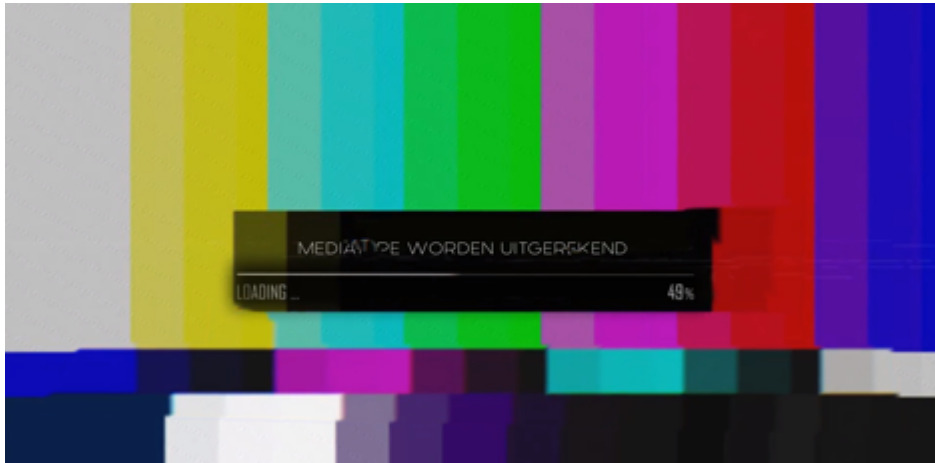
Deze versie is de laatste versie van de interface. In deze versie hebben we de transities tussen de schermen vloeiender gemaakt. Aangezien het prototype in de vorm van een website is gebouwd en uit meerdere schermen bestaat was het uitdagend om de transities vloeiender te maken. We hebben daarom ervoor gekozen om de elementen binnen een schermen vlak voordat een nieuw scherm begint te laten verdwijnen door fadeouts en sliderights effecten te gebruiken. Deze effecten komen uit een css library (animate.css) die makkelijk te implementeren is.

### ***Uiteindelijk Versie***

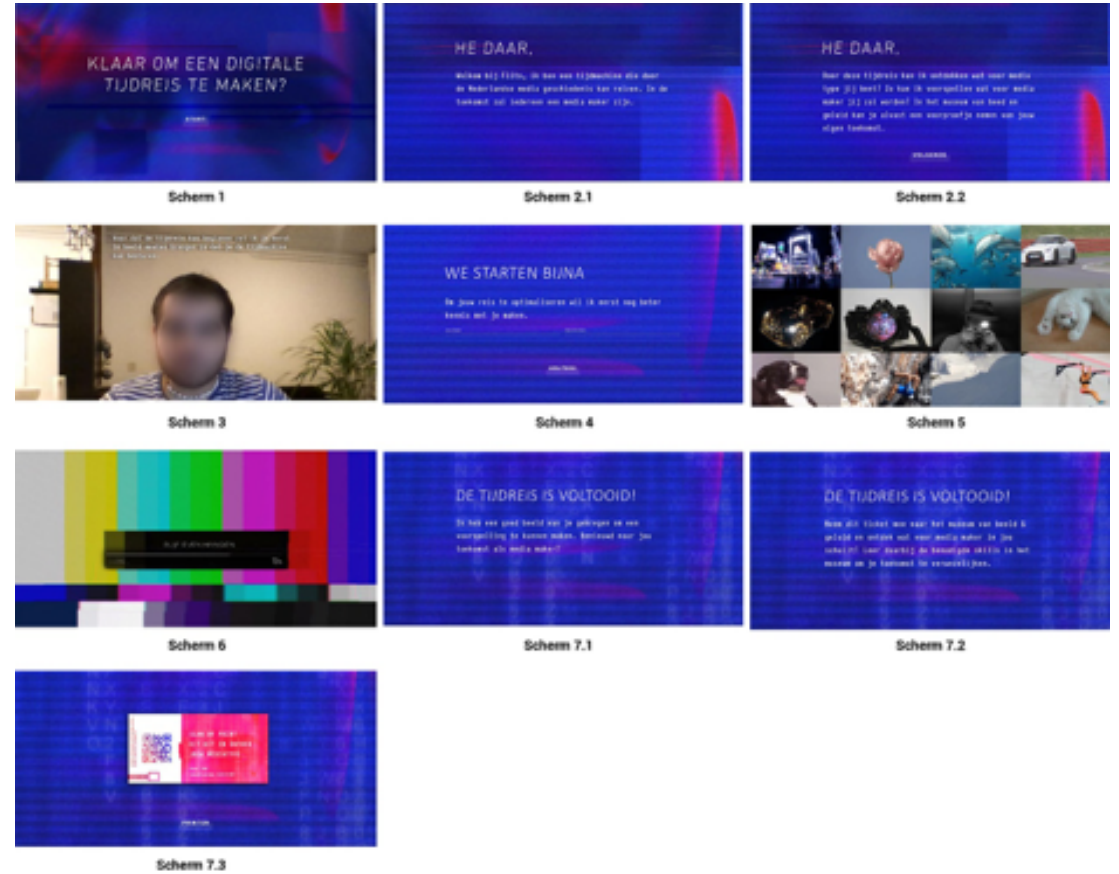
# Bijlagen







ScherM 6



Interface versie 6



Interface versie 7



Interface versie 8





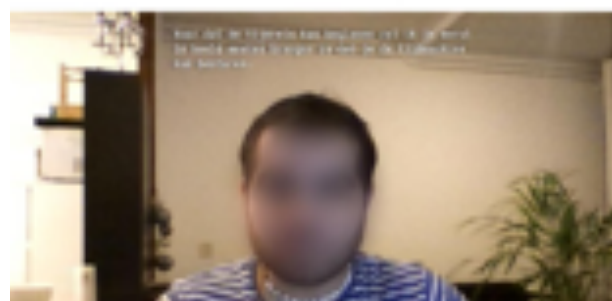
Schem 1



Schem 2.1



Schem 2.2



Schem 3



Schem 4



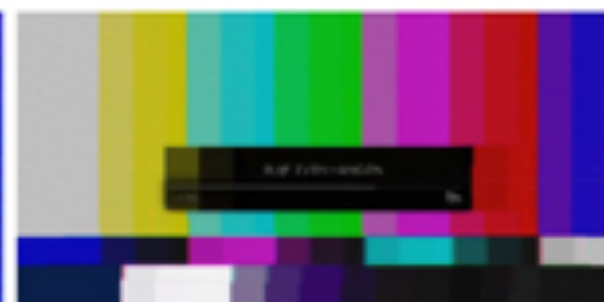
Schem 5.1



Schem 5.2



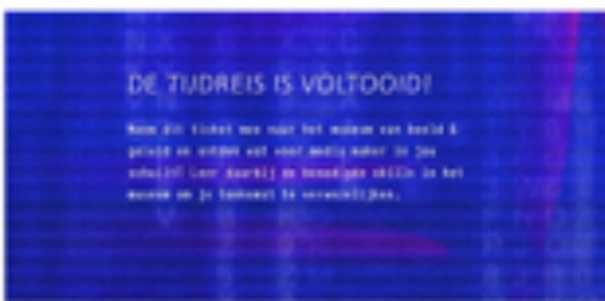
Schem 5.3



Schem 6



Schem 7.1



Schem 7.2



Schem 7.3