

**TOUCHTRIBE**

SCRIPTIE

# SAMENWERKING DESIGNERS EN DEVELOPERS VERBETEREN DOOR DESIGNSYSTEEM

Naam : Robin van Vliet

Studentnummer : S1113824

Specialisatie : Informatica, Interactie Technologie (IAT) Instelling : Hogeschool Leiden

Stagedocent: Jaap Kanbier

Periode : 1 februari 2022 t/m 1 juli 2022

## Samenvatting

Touchtribe gaf aan dat er een designsysteem ontworpen moest worden om de samenwerking tussen ontwerpers en developers beter te laten verlopen. Zij nemen aan dat een designsysteem mogelijk zou helpen om de pijnpunten uit de weg te gaan. Developers zijn duur, lastig te vinden en nog moeilijker om te behouden. Bedrijven richten zich vaak op het verbeteren van de werkervaring van ontwikkelteams, maar richten zich vaak minder op het meest belangrijke aspect: kunnen ontwikkelaars effectief werken en hebben ze plezier in hun werk?

Het doel van dit onderzoek is om de developer en designer experience te verbeteren door de pijnpunten tijd, geld en irritaties kaart te brengen. Hiervoor wordt een designsysteem gemaakt met de wensen en behoeften van de doelgroep om deze punten zo goed mogelijk te verbeteren. Het designsysteem moet uiteindelijk ook de teamsfeer binnen de samenwerking verbeteren. Hiervoor is de volgende hoofdvraag opgesteld: ***“Wat zijn de wensen en behoeften van de doelgroep de ontwerpers en developers van Touchtribe m.b.t. het ontwikkelen van een designsysteem?”***

Om antwoord te kunnen geven op deze hoofdvraag is er gebruik gemaakt van een literatuuronderzoek en een praktisch onderzoek. Er is uit het literatuuronderzoek en praktisch onderzoek een token-transformer gekomen die getest is met respondenten; de developers en designers. Er is een nul- en éénmeting uitgevoerd. Hier is naar voren gekomen dat de huidige situatie waarbij developers handmatig de tokens eruit moest halen gemiddeld een 6 kreeg. De nieuwe situatie waar door middel van de token-transformer de tokens geleverd werden een 8.2 als gemiddelde kreeg. Dit is een duidelijke stijging. Daarnaast zijn er richtlijnen van het designsysteem opgesteld.

Op basis hiervan wordt geconstateerd dat de token-transformer wel degelijk verbetering geeft in de developer en designer experience. Het eventuele vervolgonderzoek zou zich kunnen richten op het uitbreiden van deze token-transformer, als de plug-in bijvoorbeeld zich uitbreidt.

## Voorwoord

In deze scriptie wordt er gekeken naar het verbeteren van de samenwerking tussen designers en developers. Het onderzoek is uitgevoerd bij het bedrijf Touchtribe met meerdere designers en developers. Deze scriptie is geschreven in het kader van mijn afstuderen aan de opleiding Informatica aan Hogeschool Leiden en ik opdracht van stageplek Touchtribe. In de periode februari 2022 tot juli 2022 ben ik bezig geweest met het onderzoek en schrijven van de scriptie. Daarnaast is er ook nog gebouwd.

Mijn stagebegeleider Maarten Jongerius kwam met een probleem wat al een periode bij Touchtribe in de kast lag. Touchtribe is een groot groeiend bedrijf, dat geen tijd had voor het onderzoeken van dit probleem. Het onderzoek dat ik heb uitgevoerd was erg leuk en interessant, maar het bouwen van een oplossing vond ik lastig. Na uitvoering van kwalitatieve onderzoek, ofwel mijn literatuuronderzoek, kon ik de eerste deelvraag beantwoorden. Maarten en mijn stagedocent, Jaap Kanbier, stonden altijd voor mij klaar bij vragen of onduidelijkheden. Hierdoor ben ik wel erg gerustgesteld gedurende mijn stage.

Bij deze wil ik mijn begeleiders graag bedanken voor de fijne en goede begeleiding gedurende mijn stage. Ook wil ik de respondenten bedanken die mee hebben geholpen bij mijn onderzoek. Zonder de medewerking van deze respondenten kwam ik niet tot dit mooie resultaat.

Tot slot wil ik mijn collega's van Touchtribe bedanken voor de gezellige en vooral leerzame tijd bij Touchtribe. Ik kon altijd aan mijn collega's hulp vragen en vooral was het erg gezellig tussen het werken door.

Ik wens u veel leesplezier toe,

Robin van Vliet

Alphen aan den Rijn, 14 juni 2022

## Inhoudsopgave

<b>SAMENVATTING .....</b>	<b>1</b>
<b>VOORWOORD .....</b>	<b>2</b>
<b>HOOFDSTUK 1 INLEIDING .....</b>	<b>6</b>
<b>1.1 PROBLEEMANALYSE.....</b>	<b>7</b>
1.1.1 ACHTERGROND .....	7
1.1.2 AANLEIDING EN CONTEXT.....	7
1.1.3 PROBLEEMFORMULERING .....	7
1.1.4 DOELSTELLING .....	8
1.1.5 AFBAKENING DOELGROEP .....	8
<b>HOOFDSTUK 2 HOOFD- EN DEELVRAGEN .....</b>	<b>8</b>
<b>2.1 HOOFDVRAAG.....</b>	<b>8</b>
2.1.1 VERANTWOORDING HOOFDVRAAG .....	9
<b>2.2 DEELVRAGEN .....</b>	<b>9</b>
2.2.1 VERANTWOORDING DEELVRAGEN .....	9
<b>2.3 GEKOZEN AANPAK EN EIGENVERWACHTINGEN.....</b>	<b>9</b>
<b>HOOFDSTUK 3 GESERREERD LITERATUURONDERZOEK .....</b>	<b>10</b>
<b>3.1 EEN DESIGNSYSTEEM .....</b>	<b>11</b>
<b>3.2 GEBRUIK VAN DESIGNSYSTEEM .....</b>	<b>12</b>
<b>3.3 TOOLS .....</b>	<b>13</b>
3.3.1 VERSCHILLENDE TOOLS .....	13
3.3.2 STORYBOOK.....	14
3.3.3 FIGMA .....	14
3.3.4 VERGELIJKING FIGMA TOKENS VS DESIGN TOKENS.....	16
3.3.5 VOOR- EN NADELEN FIGMA .....	17
<b>3.4 INHOUD GOED DESIGNSYSTEEM.....</b>	<b>17</b>
3.4.1 RICHTLIJNEN .....	17
3.4.2 TYPOGRAFIE .....	18
3.4.3 KLEUREN .....	18
<b>3.5 VOORBEELDEN DESIGNSYSTEMEN .....</b>	<b>19</b>
<b>HOOFDSTUK 4 PRAKTISCH ONDERZOEK .....</b>	<b>20</b>
<b>4.1 ONDERZOEKSOPZET.....</b>	<b>21</b>
<b>4.2 HUIDIGE MANIER VAN WERKEN.....</b>	<b>21</b>
4.2.1 DEVELOPERS.....	21
4.2.2 DESIGNERS .....	21
<b>4.3 PROBLEMEN HUIDIGE MANIER VAN WERKEN .....</b>	<b>22</b>
4.3.1 DEVELOPERS.....	22
4.3.2 DESIGNERS .....	22

4.3.3 SWOT-ANALYSE .....	23
4.3.4 CUSTOMER JOURNEY.....	23
<b>4.4 AL AANWEZIGE WENSEN EN BEHOEFTE.....</b>	<b>24</b>
4.4.1 DEVELOPERS.....	24
4.4.2 DESIGNERS.....	25
<b>4.5 WENSEN EN BEHOEFTE.....</b>	<b>25</b>
4.5.1 DEVELOPERS.....	25
4.5.2 DESIGNERS.....	25
<b><u>HOOFDSTUK 5 METHODEN EN TECHNIEKEN.....</u></b>	<b><u>26</u></b>
5.1 STAKEHOLDERANALYSE .....	26
5.2 NUL- EN ÉÉNMETING .....	27
5.2.1 RESULTATEN METINGEN .....	28
5.2.1.1 NULMETING .....	28
5.2.1.2 ÉÉNMETING.....	29
5.2.2 CONCLUSIE .....	29
5.3 OPZETTEN DESIGNSYSTEEM .....	30
5.4 FIGMA TOKENS; DE TOKEN-TRANSFORMER .....	32
5.5 OPZET USABILITY TEST DESIGNERS.....	35
<b><u>HOOFDSTUK 6 ONDERZOEKSRISULTATEN ANALYSE.....</u></b>	<b><u>37</u></b>
6.1 DESIGNSYSTEEM.....	37
6.2 USABILITY TEST DESIGNERS .....	38
6.3 ADVIESRAPPORT.....	40
<b><u>HOOFDSTUK 7 CONCLUSIE .....</u></b>	<b><u>41</u></b>
7.1 HOOFDVRAAG.....	41
7.2 DEELVRAGEN .....	41
<b><u>HOOFDSTUK 8 A-COMPETENTIES .....</u></b>	<b><u>43</u></b>
8.1 ONDERZOEK .....	43
8.2 LEREN LEREN.....	43
8.3 PROFESSIONEEL WERKEN .....	44
8.4 INNOVATIE .....	45
9.1 GEBRUIKERSINTERACTIE – ANALYSEREN.....	46
9.1.1 GEBRUIKERSONDERZOEK.....	47
9.1.2 METHODEN EN TECHNIEKEN .....	47
9.1.3 PROTOTYPING & DESIGN.....	47
9.1.4 USABILITY .....	47
9.2 GEBRUIKERSINTERACTIE – ADVISEREN .....	47
9.2.1 ONDERZOEK .....	47
9.2.2 RAPPORTAGE .....	47
9.2.3 PROTOTYPING.....	47
9.3 GEBRUIKERSINTERACTIE – REALISEREN .....	47
9.3.1 VERTALING VAN ONTWERP.....	47

DESIGNSYSTEEM.....	48
WEBSITE NABOUWEN .....	48
9.3.2 USABILITY TESTEN.....	51
9.3.3 STAKEHOLDERS .....	51
<b>HOOFDSTUK 10 NAWOORD .....</b>	<b>52</b>
<b>HOOFDSTUK 11 REFLECTIE .....</b>	<b>53</b>
<b>BIJLAGEN .....</b>	<b>54</b>
<b>BIJLAGE 1 THEORETISCH KADER EN LITERATUURONDERZOEK .....</b>	<b>54</b>
<b>HOOFDSTUK 1 EEN DESIGNSYSTEEM.....</b>	<b>55</b>
1.1 ATOMIC DESIGN .....	55
1.2 VOORBEELDEN DESIGNSYSTEEM .....	57
<b>HOOFDSTUK 2 GEBRUIK VAN DESIGNSYSTEEM .....</b>	<b>57</b>
<b>HOOFDSTUK 3 TOOLS.....</b>	<b>58</b>
3.1 VERSCHILLENDE TOOLS.....	58
3.2 STORYBOOK .....	59
3.2.1 INSPECTEREN IN STORYBOOK.....	59
3.2.2 VOOR- EN NADELEN STORYBOOK.....	60
3.3 FIGMA.....	61
3.3.1 JSON-TOKENS; FIGMA TOKENS .....	62
3.3.2 JSON-TOKENS; DESIGN TOKENS.....	65
3.2.3 VERGELIJKING FIGMA TOKENS VS DESIGN TOKENS .....	67
3.2.4 VOOR- EN NADELEN FIGMA .....	67
3.4 INHOUD GOED DESIGNSYSTEEM.....	68
3.4.1 RICHTLIJNEN .....	68
3.4.2 TYPOGRAFIE .....	69
3.4.3 KLEUREN.....	69
3.5 VOORBEELDEN DESIGNSYTEMEN .....	71
<b>BIBLIOGRAFIE.....</b>	<b>73</b>
<b>BIJLAGE 2 INTERVIEWS .....</b>	<b>75</b>
<b>BIJLAGE 3 FIGMA BESTAND.....</b>	<b>75</b>
<b>BIJLAGE 4 NUL- EN ÉÉNMETING .....</b>	<b>75</b>
NULMETING .....	75
ÉÉNMETING.....	75
<b>BIJLAGE 5 PROTOTYPE DESIGNSYSTEEM.....</b>	<b>76</b>
<b>BIJLAGE 6 CASUS DESIGNERS.....</b>	<b>84</b>
<b>BIJLAGE 7 RESULTATEN USABILITY TEST DESIGNERS .....</b>	<b>91</b>
<b>BIJLAGE 8 DESIGNSYSTEEM CERQLAR .....</b>	<b>92</b>

BIJLAGE 9 ADVIESRAPPORT .....	100
<b><u>MANAGEMENTSAMENVATTING.....</u></b>	<b>101</b>
<b><u>INLEIDING .....</u></b>	<b>103</b>
<b><u>ONDERZOEK.....</u></b>	<b>104</b>
<b><u>ALTERNATIEVEN.....</u></b>	<b>106</b>
ALTERNATIEF 1 .....	106
ALTERNATIEF 2 .....	106
<b><u>CONCLUSIE.....</u></b>	<b>107</b>
<b><u>BIBLIOGRAPHY.....</u></b>	<b>108</b>

## Hoofdstuk 1 Inleiding

In het eerste Hoofdstuk wordt het probleem besproken waar Touchtribe tegen aanloopt.

## 1.1 Probleemanalyse

### 1.1.1 Achtergrond

Touchtribe is een web-development bedrijf dat e-commerce oplossingen bouwt die als doel hebben merken te laten groeien. Dit doen ze door gebruik te maken van moderne cloudtechnologieën headless development (Touchtribe, 2022). Zij leveren cloud native headless CMS-oplossingen waarmee de content op ieder gewenst platform uitgeserveerd kan worden. Touchtribe zit aangesloten bij Makerstreet. Dit houdt in dat zij een aparte tak hebben voor front-end, back-end en design. Het front-end team ontwerpt dus niet zelf de websites en apps.

Touchtribe loopt er tegenaan dat ontwerpers en developers vaak onenigheid hebben over het opgeleverde werk. Het opgeleverde design is niet altijd even goed te maken voor een developer. Een voorbeeld hierbij is dat developers een ontwerp krijgen waar niet alle variabelen een naam hebben of afstanden niet kloppen. De developers wensen te zien dat alle basics compleet worden gedefinieerd in een styleguide. Hierdoor wordt er voorkomen dat developers de verkeerde variabelen in de code doorvoeren of dat zij zelf achter een niet geheel ontwerp moeten gaan. Dit kost voor de developer tijd, omdat het weer terug moet naar de designers en het even duurt voordat dit ontwerp verbetert terugkomt. Uiteindelijk komen de developers in de knel met het aantal sprints dat zij voor een project hebben. Voor de designers kost dit tijd, omdat zij sneller dan de developers op een nieuw project worden gezet. Zij moeten dan tijdens een ander project terug aan het ontwerp van een vorig project.

Een ander ergerpunt van de developers is dat zij handmatig designtokens uit het ontwerp moeten halen, meer hierover wordt besproken in Hoofdstuk 4 Praktisch onderzoek. Een designtoken zijn bijvoorbeeld de kleuren en typografie. Deze onenigheden kosten tijd en geld. Een van de ergernissen van de designers is dat de developers onderling niet rond hebben hoe zij een styleguide opgeleverd willen hebben. Om deze ergernissen tegen te gaan, moet er een designsysteem<sup>1</sup> ontworpen worden om alle ontwerpers iets consistentes aan Touchtribe te leveren. Dit designsysteem wordt gedurende mijn stageopdracht ontwikkeld.

### 1.1.2 Aanleiding en context

Om de samenwerking tussen developers en ontwerpers beter te laten verlopen moet er een designsysteem ontwikkeld worden waar alle ontwerpers mee overweg kunnen. Er wordt een literatuuronderzoek gedaan naar de designtokens en er worden interviews gehouden met de stakeholders, de designers en developers. Om een goed designsysteem neer te zetten, moet er eerst een onderzoek uitgevoerd worden.

### 1.1.3 Probleemformulering

Touchtribe gaf aan dat er een designsysteem ontworpen moest worden om de samenwerking tussen ontwerpers en developers beter te laten verlopen. Zij nemen aan dat een designsysteem mogelijk zou helpen om de pijnpunten uit de weg te gaan. Developers zijn duur, lastig te vinden en nog moeilijker om te behouden. Bedrijven richten zich vaak op het verbeteren van de werkervaring van

---

<sup>1</sup> “**Designsysteem**” is een systematische benadering voor het ontwikkelen en onderhouden van consistente gebruikersinterfaces die de merkwaarden communiceren en de gebruikerservaring versterken – op een samenhangende wijze.



ontwikkelteams, maar richten zich vaak minder op het meest belangrijke aspect: kunnen ontwikkelaars effectief werken en hebben ze plezier in hun werk? (Ward, 2020). Meer over de pijnpunten van Touchtribe is te lezen in Hoofdstuk 4 Praktisch onderzoek.

#### 1.1.4 Doelstelling

Mijn doelstelling is om de developer en designer experience te verbeteren door de pijnpunten tijd, geld en irritaties in kaart te brengen. Hiervoor wordt een designsysteem gemaakt met de wensen en behoeften van de doelgroep om deze punten zo goed mogelijk te verbeteren. Het designsysteem moet uiteindelijk ook de teamsfeer binnen de samenwerking verbeteren.

Er wordt gebruik gemaakt van een nul- en éénmeting. Een designer maakt voor dit onderzoek een simpele landingspagina met styleguide in Figma. Eerst ga ik zelf deze homepage namaken zonder het designsysteem met een timer. Hier moeten de designtokens, zoals bijvoorbeeld kleuren er handmatig uitgehaald worden. Daarna wordt dit bestand nogmaals nageemaakt, maar dit keer met designsysteem. Bij deze éénmeting wordt er ook gekeken hoe lang hierover gedaan wordt.

Het nieuwe designsysteem wordt met designers getest, om te kijken of zij de nieuwe richtlijnen begrijpen en of zij nog feedback hebben. Ook wordt het getest met developers. De 3 testpersonen testen net zoals ik door middel van een nul- en éénmeting het nieuwe designsysteem.

Als de samenwerking na het designsysteem niet verbetert, moet er gekeken worden of het dan niet aan andere problemen ligt zoals communicatieproblemen.

#### 1.1.5 Afbakening doelgroep

De doelgroep bestaat uit vier stakeholders, zie Hoofdstuk 5.1 Stakeholderanalyse, ofwel:

- Touchtribe
- Klanten
- Developers
- Designers

## Hoofdstuk 2 Hoofd- en deelvragen

### 2.1 Hoofdvraag

De hoofdvraag luidt als volgt:

***“Wat zijn de wensen en behoeften van de doelgroep de ontwerpers en developers van Touchtribe m.b.t. het ontwikkelen van een designsysteem?”***

### 2.1.1 Verantwoording hoofdvraag

Het uitgangspunt bij de hoofdvraag is het opleveren van een designsysteem, waarbij op meerdere vlakken de developer en designer experience wordt verbeterd. Eerst heb ik onderzoek gedaan naar de richtlijnen van een goed designsysteem, zie Hoofdstuk 3 Geserreerd literatuuronderzoek. Daarna zijn de wensen en behoeften van de doelgroep in kaart heb gebracht, zie Hoofdstuk 4 Praktisch onderzoek. Er wordt getest met het designsysteem.

De resultaten hiervan zijn te lezen in Hoofdstuk 6 Onderzoeksresultaten analyse. De uiteindelijke conclusie wordt beknopt samengevat in Hoofdstuk 7.1 Conclusie hoofdvraag.

## 2.2 Deelvragen

De deelvragen luiden als volgt:

- *“Wat is een designsysteem?”*
- *“Waar loopt de doelgroep tegenaan tijdens het werk?”*
- *“Wat is de huidige manier van werken bij de doelgroep?”*
- *“Welke informatie wenst de doelgroep te zien bij het designsysteem?”*

### 2.2.1 Verantwoording deelvragen

Als eerst moet er gekeken worden naar de kenmerken van een designsysteem. Wat is een designsysteem? Welke richtlijnen bevat een designsysteem? Als tweede moet er gekeken worden waar de doelgroep tegenaan loopt tijdens het werk zoals de ergerpunten van de developers en designers. Dit is eerder kort beschreven in Hoofdstuk 1.1.1 Achtergrond en is verder uitgewerkt in Hoofdstuk 4 Praktisch onderzoek.

## 2.3 Gekozen aanpak en eigenverwachtingen

Mijn eerste verwachting is dat een designsysteem bepaalde richtlijnen bevat en een bepaalde manier van werken is, waardoor er structuur voor een bedrijf is bij het bouwen van websites en apps. Bij deze manier van werken ga ik ervanuit dat dit meer rust geeft als je met meerdere takken werkt, zoals bij Touchtribe het geval is. Daarnaast verwacht ik dat de problemen die bij Hoofdstuk 1.1.1 Achtergrond worden beschreven opgelost kunnen worden met het opstellen van een designsysteem.

De gekozen aanpak om achter mijn eerste verwachting te komen is een Theoretisch kader en literatuuronderzoek uitgevoerd te hebben die te vinden is in *Bijlage 1 Theoretisch kader en literatuuronderzoek*. De beknopte samenvatting van deze bijlage is te vinden in Hoofdstuk 3 Geserreerd literatuuronderzoek.

Mijn tweede verwachting wordt onderzocht door middel van het in kaart brengen van de huidige manier van werken, zie Hoofdstuk 4 Praktisch onderzoek. Daarnaast wordt er, zoals in Hoofdstuk 1.1.4 Doelstelling beschreven, gebruik gemaakt van een nul- en éénmeting om aan te kunnen tonen of het designsysteem daadwerkelijk de manier van werken versnelt. Deze metingen worden na mijn eigeninzichten, ook getest met 3 developers. Er wordt van de developers gevraagd

de tijd en irritaties een cijfer te geven bij beide metingen, om een goede vergelijking na de testen te kunnen schetsen.

Tot slot wordt er een website nagebouwd met het designsysteem om met mijn eigen tokens te werk te gaan en om in de manier van werken van Touchtribe te duiken.

## Hoofdstuk 3 Geserreerd literatuuronderzoek

*In Bijlage 1 Theoretisch kader en literatuuronderzoek is het uitgebreide literatuuronderzoek te vinden met meer visuele afbeeldingen dan het Hoofdstuk 3 Geserreerd literatuuronderzoek. In dit hoofdstuk wordt de beknopte variant omschreven en de eerste deelvraag “Wat is een designsysteem?” beantwoord.*

### 3.1 Een designsysteem

Bij een optimale gebruikerservaring draait het voornamelijk allemaal om consistentie, hier speelt een designsysteem een grote rol. Een designsysteem is een set aan standaarden en richtlijnen over hoe je dingen maakt en ontwerpt, denk hierbij aan een brand- of styleguide. In de guide staat waar je rekening mee moet houden als je met een merk of bedrijf aan de slag gaat (Soda Studio, 2021). Enkele voorbeelden hiervan zijn lettertypes, kleuren, afmetingen enzovoort. Om dieper in te gaan op een designsysteem, wordt het onderverdeeld in verschillende onderdelen: atomen, moleculen en organismes (Woznica, 2018). Deze Atomic Components vormen samen een Atomic Design, zie *Afbeelding 1 Atomic Components*.



*Afbeelding 1 Atomic Components*

Als eerst heb je atomen. Dit zijn de kleinste onderdelen van een project, zoals de buttons, kleuren, titels, labels, lettertypes, paragrafen, enzovoort. Deze atomen kunnen worden toegepast op elke context, globaal of binnen andere componenten, zie in *Bijlage 1 Theoretisch kader en literatuuronderzoek; Afbeelding 2 Voorbeeld atomen*.

Als tweede heb je moleculen. Moleculen zijn een samenstelling van meerdere componenten of atomen. Moleculen worden gezamenlijk meer waard. Elke molecule kan een eigen eigenschap hebben en functionaliteiten creëren door atomen te gebruiken, die op zichzelf geen enkele functie hebben. In *Bijlage 1 Theoretisch kader en literatuuronderzoek; Afbeelding 3 Voorbeeld moleculen* is een visualisatie te vinden.

Het derde onderdeel zijn de organismen. Organismen zijn een combinatie van moleculen die samenwerken of zelfs met atomen die meer uitgebreide grensvlakken vormen, ook wel de bouwstenen. Een voorbeeld hiervan is een header van een website, zie *Bijlage 1 Theoretisch kader en literatuuronderzoek; Afbeelding 4 Voorbeeld organismen*.

Om het volledige Atomic Design door te nemen, worden ook de templates en de pages besproken. Templates zijn een blauwdruk van een website of applicatie. Het template bestaat uit gegroepeerde organismen die samen een pagina vormen, zie *Bijlage 1 Theoretisch kader en literatuuronderzoek; Afbeelding 5 Voorbeeld templates*. Tot slot heb je nog de pages, dit is het template met ingevulde content die van toepassing is voor de organisatie of klant.

Een Atomic Design helpt om dubbele taken te voorkomen, door het veel duidelijker is welke componenten hergebruikt kunnen worden en er geen dubbele code wordt geschreven. Door de opbouw van kleine naar grotere onderdelen is het voor programmeurs ook een stuk eenvoudiger om projecten over te dragen. Naast componenten, houdt een designsysteem ook documenten, artikelen, voorbeelden, codefragmenten, screenshots, ontwerprichtlijnen, filosofieën en andere digitale middelen in (Alli, 2020).

### 3.2 Gebruik van designsysteem

Er zijn vijf redenen waarom er een designsysteem gebruikt moet worden. De eerste reden is dat ontwerp- en ontwikkelingswerk snel en op grote schaal gemaakt kan worden. Bij een ontwerpsysteem staat het vermogen om snel ontwerpen te repliceren door gebruik te maken van vooraf gemaakte UI-componenten centraal. Dit scheelt tijd, doordat ontwerpers steeds dezelfde elementen kunnen gebruiken (Fessenden, 2021). Een manier om deze snelheid te garanderen is geïntegreerde producten en workflows te gebruiken die samenwerking ondersteunen. Dit zal het

team helpen om in een sneller tempo te werken. Als alle feedback op één plek verzameld kan worden, kunnen ontwerpers snel versies herzien en publiceren (Wadsworth, 2019).

De tweede reden is dan ook dat ontwerpers zich minder hoeven te concentreren op ontwerpbronnen te zoals het visuele uiterlijk en kunnen hierdoor de focus leggen op complexere problemen zoals prioriteitstelling van informatie, workflow optimalisatie en reisbeheer (Wadsworth, 2019).

De derde reden houdt in dat het een uniforme taal binnen en tussen cross functionele teams creëert. Als ontwerpverantwoordelijkheden verschuiven of wanneer teams veranderen, wordt er tijd verspild aan ontwerp- of ontwikkelingstijd rondom miscommunicatie. Dit is een struikelpunt van Touchtribe dat in Hoofdstuk 1.1.4 wordt Achtergrond beschreven. Een voorbeeld hiervan is dat er nu geen discussie gevoerd hoeft te worden over het uiterlijk van een component zoals een button, omdat dit in het designsysteem teruggevonden kan worden (Wadsworth, 2019).

De vierde reden is dat het voor virtuele consistentie zorgt tussen producten, kanalen en andere afdelingen. Als teams van verschillende afdelingen onafhankelijk van elkaar opereren, kan het missen van een designsysteem leiden tot discussie (Wadsworth, 2019).

De vijfde reden is dat het kan dienen als een educatief hulpmiddel en naslagwerk voor ontwerpers op juniorniveau en inhoudelijke bijdragers. Nadrukkelijke geschreven gebruikersrichtlijnen en stijlguides helpen individuele bijdragers aan boord te krijgen die nieuw zijn in UI-ontwerp of het maken van inhoud. Ook dienen ze als herinnering voor de rest van de bijdragers (Wadsworth, 2019).

### 3.3 Tools

Er wordt dieper ingegaan op Storybook en Figma, om te kijken op welke van deze tools voortgebouwd kan worden voor een goed designsysteem.

#### 3.3.1 Verschillende tools

Er zijn vier type tools die je kan gebruiken voor je designsysteem; design tools, documentation tools, design documentation tools en custom platforms.

#### **Design tools**

Met design tools zoals Figma, Adobe XD, Sketch en Framer kan je online schermen ontwerpen. Bij deze tools kan je bijvoorbeeld het scherm delen en afbeeldingen of componenten exporteren.

### **Documentations tools**

Documentations tools hosten toegankelijke websites die iedereen kan gebruiken om informatie op te halen. Voorbeelden van documentations tools zijn Gitbook, VuePress, Fractal en Confluence (Swierts, 2020).

### **Design documentation tools**

De tools Frontify, Zeroheight, Nuclino, Lucid en Bynder zijn tools die gericht zijn op design. De layouts die deze tools aanbieden zijn meer ondersteunend voor het tonen van merk en ontwerp (Swierts, 2020).

### **Custom platforms**

Er zijn mogelijkheden om je eigen platform te creëren. Een eigen platform houdt meer vrijheid in met wat mogelijk is. Enkele voorbeelden zijn Google's Material Design en Shopify's Polaris (Swierts, 2020).

## **3.3.2 Storybook**

Storybook is een open source tool voor het bouwen van UI componenten en pagina's. Het stroomlijnt de ontwikkeling, het testen en de documentatie van de gebruikersinterface (Storybook, 2022).

### **Voor- en nadelen Storybook**

#### *Voordelen*

- Enkel component zien; je kan een enkel component te zien krijgen, in plaats van alle componenten van die categorie; zoals Buttons/button-01.
- CSS; als de CSS wordt vergeten neer te zetten in het Figma bestand door de ontwerpers, wordt het in Storybook automatisch neergezet
- Update automatisch; bij aanpassingen vanuit Figma wordt het automatisch geüpdatet in Storybook

#### *Nadelen*

- Traag; Als Storybook helemaal gevuld is met component duurt het inladen erg lang
- Geen JSON-tokens kunnen versturen; dit kan alleen in Figma zelf
- Voorkeur developers; developers vinden de informatie vanuit Figma halen makkelijker

## **3.3.3 Figma**

Figma is een cloudgebaseerde tool voor het ontwerpen en prototypen van de UI waarmee meerdere teams kunnen samenwerken bij het maken, testen en implementeren van interface- of productontwerpen (Figma, 2022). Er zijn verschillende plug-ins die aangeboden worden vanuit Figma waaruit JSON-tokens verstuurd kunnen worden naar GitHub. Dit is een belangrijke eis vanuit Touchtribe. De plug-ins Figma Tokens en Design Tokens worden besproken en vergeleken.

### **JSON-tokens; Figma Tokens**

Bij Figma Tokens kunnen er herbruikbare en semantische tokens gemaakt worden waarnaar vervolgens verwezen kan worden in het ontwerpproces. Enkele voorbeelden van beschikbare token

groepen zijn: sizing, spacing, color, fon-familiy, enzovoort. Zie voor de gehele tabel *Bijlage 1 Theoretisch kader en literatuuronderzoek; 3.3.1 JSON-tokens Figma Tokens*.

Het creëren van een token via de gebruikersinterface is bij Figma Tokens eenvoudig. Een van de manieren is als volgt; je opent de plug-in en klikt op het plus icon. Door middel van deze knop kan je een token toevoegen. Hierna wordt er een ander venster geopend waarin je de gegevens kan plaatsen.

Ook kan in Figma Tokens er in waarde naar andere tokens verwezen worden. Dit betekent dat een merkkleur verwezen kan worden als `colors.red.500`, of een achtergrondtoken hebben die zijn waarde ontleent aan een `colors.black` token, terwijl voorgrondtoken zijn waarde ontleent aan een `colors.white` token. Om een alias in de tokens te gebruiken, worden ze in de volgende notatie geschreven: `{spacing.sm}`.

Daarnaast kan de plug-in automatisch kleur- en typografiestijlen converteren naar tokens. Het beste deel van deze aanpak is dat de plug-in probeert uw basiseenheden te bepalen en hiervoor tokens te maken.

Tot slot kan Figma Tokens gesynchroniseerd worden met de volgende bronnen:

- JSONBin
- GitHub; push en pull naar repository. Gecombineerd met GitHub Actions en style-richtingen creëert dit je eigen pijplijn.
- URL (read-only)

### **JSON-tokens; Design Tokens**

De plug-in Design Tokens is door Luka Oppermann ontwikkelt. Met Design Tokens kan je je Figma ontwerp omzetten in JSON-tokens. Deze JSON-tokens kunnen gegenereerd worden als een zipfile en naar de GitHub worden gestuurd. De plug-in zet de stijlen die worden gedefinieerd in Figma om in ontwerptokens. Dit omvat tekst-, kleur, raster- en effectstijlen.

Bij Design Tokens kan met het standaard token format een alias/referentie voor kleurtokens gedefinieerd worden via het beschrijvingsveld. Een voorbeeld; bij de kleurstijl `danger-color` dat verwijst naar de kleurstijl `colors/red` kan een `ref:regel` worden toegevoegd aan het beschrijvingsveld van de `danger-color`:



*Use this color for destructive actions only.*  
*ref: core-colors.red*

Dit wordt gecreëerd in JSON als:

```
"danger-color": {  
  "description": "Use this color for destructive actions only.",  
  "type": "color",  
  "value": "{core-colors.red}"  
};
```

De plug-in ondersteunt aangepaste tokens voor randen, radii, maten, afstand en beweging. Elk aangepast ontwerptopen moet zich binnen een frame op het hoogste niveau bevinden met een naam die begint met *\_tokens*. Dit betekent dat de structuur als volgt is: pagina > *\_tokens*/sizes > sizes/8. De token zelf moet een naam hebben die begint met sizes, spacing, borders, radii of motion en moet de Main Component of een variant zijn.

Daarnaast biedt Design Tokens een optie om de JSON-tokens in een zipfile te ontvangen. Er kan worden gekozen om een compressie van dit bestand te maken.

Tot slot kan Design Tokens gesynchroniseerd worden met GitHub, maar moeten enkele gegevens worden ingevuld zoals de event type, server URL, accept header, auth type en acces token. De acces token is je personal acces token van GitHub. Naast het opzetten van de repository, moet je enkele bestanden vanaf een andere repository genaamd Design Tokens Transformer halen. Zonder deze bestanden doet de converter het niet.

### 3.3.4 Vergelijking Figma Tokens vs Design Tokens

Doordat het development team niet op de hoogte was van de verdere vooringenomen van het ontwerpteam, had ik mij inmiddels verdiept in een plug-in genaamd Design Tokens. Om te kijken welke plug-in het beste gaat werken voor het team, heb ik de voor- en nadelen onder elkaar gezet.

#### **Design Tokens**

##### *Voordelen*

- Design Tokens kan de tokens in een .zip opleveren die gedownload kan worden op een device
- Kan naar GitHub gestuurd worden

##### *Nadelen*

- Is door een enkele developer gebouwd; zitten nog bugs in

- Verzend af en toe de tokens niet; als ze niet juist gelabeld zijn
- Als developer stopt, houdt de plug-in op
- Aparte bestanden voor gebruik; voor gebruik moet je bijvoorbeeld eerst de package-json van de Design-Tokens-Transformer overnemen in je eigen package-json bestand.

### **Figma Tokens**

#### *Voordelen*

- Plug-in wordt vertegenwoordigd vanuit Figma; waterdicht systeem
- Kan naar GitHub gestuurd worden
- Eenvoudiger; kan je eigen benamingen meegeven

#### *Nadelen*

- Als developers stoppen, houdt de plug-in op

### **3.3.5 Voor- en nadelen Figma**

Er zijn enkele voor- en nadelen aan het gebruiken van Figma om een nieuw designsysteem op te stellen:

#### **Voordelen**

- Momenteel wordt er al meegewerkt door designers en developers
- JSON-tokens; er kunnen JSON-tokens verstuurd worden naar GitHub
- Handige plug-ins; eindeloze handige plug-ins die een designsysteem ondersteunen

#### **Nadelen**

- Verkeerd gelabelde namen; ontwerpers labelen niet altijd de componenten juist, waardoor developers heel erg moeten zoeken naar componenten
- Zien alle componenten; je ziet alle componenten naast elkaar, in plaats van dat je ook enkele kan zien.

## **3.4 Inhoud goed designsysteem**

### **3.4.1 Richtlijnen**

Een designsysteem bestaat uit verschillende delen om zoveel mogelijk structuur te geven:

#### **Content, informatie en doelen**

Bij het opstellen van een designsysteem is het erg belangrijk om over de doelen van de eindgebruikers na te denken. Het is belangrijk om gesprekken te voeren met de eindgebruikers om de wensen en behoeften hierin te verwerken (Hacq, 2018).

#### **Elementen en componenten**

De beste designsystemen bevatten zowel UI-elementen (een catalogus van herbruikbare componenten), als standaard merkgidsen, waaronder het kleurenpalet, typografie en iconografie.

Verder bestaat het uit bijvoorbeeld gegevensvisualisatie en ontwerpprincipes. Ontwerpprincipes zijn leidende zinnen die de teams helpen om het doel van het product te bereiken dankzij het ontwerp. Ze helpen de teams om zinvolle ontwerpbeslissingen te nemen. (Hacq, 2018).

### **Ontwerpers als designsysteem gebruikers**

Het is voor ontwerpers erg belangrijk dat er gedocumenteerd staat hoe en wanneer elk onderdeel gebruikt moet worden. Dit helpt om ontwerpers vrij om hun focus te leggen aan het besteden aan het bedenken van oplossingen van unieke uitdagingen, in plaats zich te concentreren op componenten die al in gebruik zijn opnieuw uit te vinden, (Cunningham, 2022). De elementen moeten de volgende punten bevatten om ontwerpers te helpen begrijpen hoe ze het moeten gebruiken:

- *Beschrijving*: Wat is dit element? Wat is het doel?
- *Gebruiksoptmerkingen*: Wanneer en waar moet dit element gebruikt worden?
- *Verwante elementen*: Als dit element op een ander element lijkt, overweeg dan om naar andere verwante elementen te verwijzen. In veel gevallen kan het handig zijn om te schetsen wanneer het ene of het andere element gebruikt moet worden gebruikt, afhankelijk van de gebruikerssituatie en het doel.
- *UX of visuele opmerkingen* (optioneel): Aanvullende opmerkingen over de interactie of het visuele ontwerp van het element.
- *Opmerkingen voor ontwikkelaars* (optioneel, tenzij ontwikkelaars ook beoogde gebruikers zijn): Aanvullende opmerkingen over hoe een component is gebouwd.

### **Developers als designsysteem gebruikers**

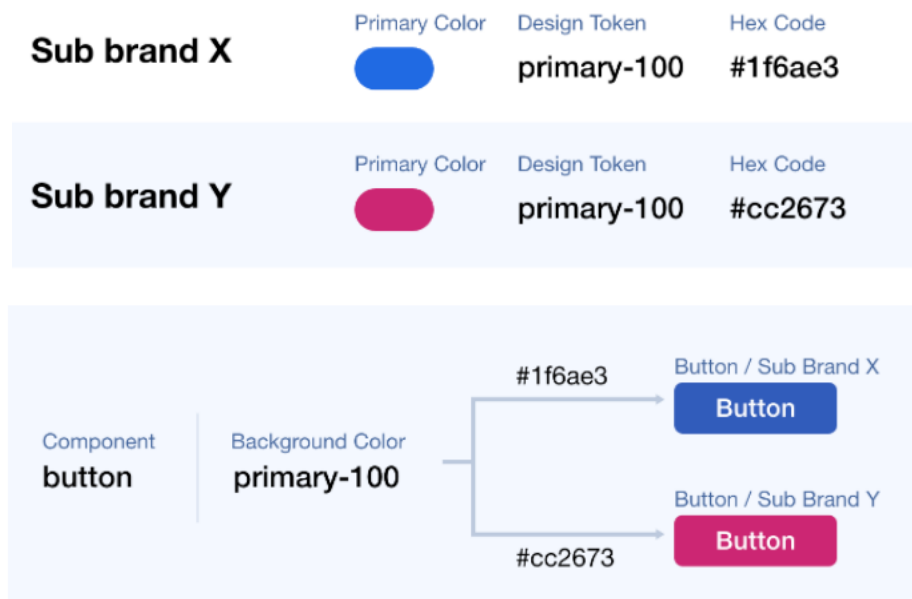
Het hangt van de doelen van het designsysteem af of organisaties het designsysteem gebruiken om ook de code voor elk UI-element te catalogiseren. Hierdoor kunnen ontwikkelaars het designsysteem gebruiken om standaardcomponenten te gebruiken in plaats van ze elke keer anders te bouwen.

## **3.4.2 Typografie**

Typografie is een van de belangrijkste aspecten van een designsysteem om in een vroeg stadium vast te stellen. Door middel van typografie brengt de website of applicatie de informatie over aan de gebruikers (Design Systems, sd).

## **3.4.3 Kleuren**

Naast het opzetten van de typografie is het opzetten van een kleurenpalet ook erg belangrijk om daarna door te kunnen met het verder ontwerpen van componenten, patronen en sjablonen (Choudhary, 2021). Zo wordt er gebruik gemaakt van labels. Door middel van labels is het makkelijker te communiceren dan een hexwaade zoals '#1F6AEE3', zie *Afbeelding 2 Voorbeeld Primary Color*.



Afbeelding 2 Voorbeeld Primary Color

Daarnaast zijn het labelen van verschillende tinten ook van belang, meer over deze labels is te lezen in *Bijlage 1 Theoretisch kader en literatuuronderzoek; 3.4.3 Kleuren*. De schaal weerspiegelt een bekend bereik van donker naar licht, maakt het mogelijk om nieuwe opties ertussen te injecteren. De waarde van de 'XX' hangt af van de volgende drie aspecten:

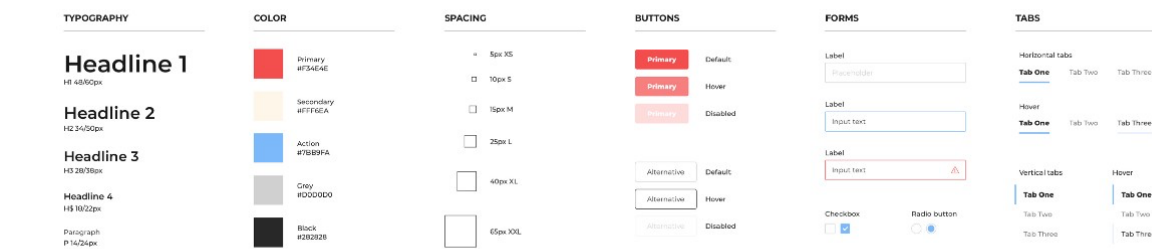
- Basiskleur: 100
- Tinten donkerder dan de basiskleur: > 100
- Tinten lichter dan de basiskleur: < 100



Afbeelding 3 Voorbeeld kleurenlabel

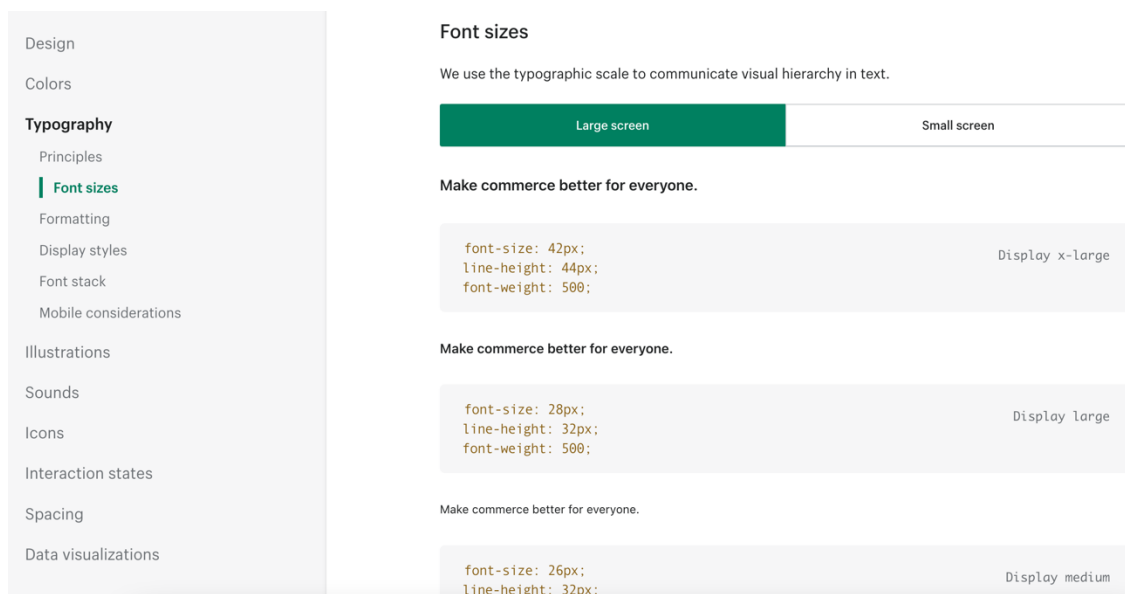
### 3.5 Voorbeelden designsystemen

In *Afbeelding 4 Voorbeeld designsysteem* is een designsysteem te zien. Hier worden de typografie meegegeven, de kleuren met HEX-waarde. De spacing, als een ontwerp die bevat en andere mogelijke variabelen die een designsysteem kan bevatten zoals buttons, forms en tabs.



Afbeelding 4 Voorbeeld designsysteem

In Afbeelding 5 Voorbeeld Shopify is het designsysteem van Shopify te zien. Dit is een website waar alle styleguide van Shopify te vinden is.



Afbeelding 5 Voorbeeld Shopify

## Hoofdstuk 4 Praktisch onderzoek

In dit hoofdstuk wordt het praktisch onderzoek beschreven waarbij de volgende deelvragen worden beantwoord:

- “Waar loopt de doelgroep tegenaan tijdens het werk?”
- “Wat is de huidige manier van werken bij de doelgroep?”

- “Welke informatie wenst de doelgroep te zien bij het nieuwe designsysteem?”

## 4.1 Onderzoekopzet

Voor de deelvragen is er contact opgenomen door middel van een interview met Linda van Dijk uit het front-end development team van Touchtribe en Vincent Koopmans uit het ontwerpteam van Makerstreet. Het audiofragment met Vincent Koopmans is te vinden in *Bijlage 2 Interviews*.

## 4.2 Huidige manier van werken

### 4.2.1 Developers

Er worden enkele stappen doorlopen bij het opzetten van een project bij de developers. De stappen zijn als volgt:

1. Er wordt begonnen met het opzetten van alle standaarden in een project. React, typescript, styled components etc. Hier hebben is een library voor dus gaat dit vrij snel.
2. Vervolgens wordt het gekozen CMS opgezet en wordt er de koppeling gemaakt.
3. Daarna kan er aan de slag gegaan worden met het ‘design system’, in hoeverre Touchtribe die heeft. Eerst wordt de ‘base’ als variabelen: kleuren, typography, layout, icons, shadows, borders geïmplementeerd. Meestal wordt dit aangevuld in de loop van het project. Dit komt omdat nooit alles van tevoren is gedefinieerd door de ontwerpers. Doordat vaak niet alles gedefinieerd is wekt dit irritaties bij de developers, die er dan weer achteraan moeten gaan bij de designers. Ook dit kost extra tijd. Af en toe worden designers dan al op andere projecten gezet, omdat hun werk erop zit bij het huidige project. Hierdoor reageren ze niet zo snel op de developers die zelf achter de variabelen moeten aangaan.
4. Na het opzetten van de basisvariabelen wordt er begonnen met de kleinste componenten die vaker terugkomen op verschillende plekken zoals: button, input, select, checkbox, radio button etc.
5. Dan kan er pas worden begonnen aan de ‘content blocks’ en de header en footer. Een content block is het blok dat de klant uiteindelijk kan selecteren in het CMS en vrij kan gebruiken over de hele website (Orace, 2022). Het is meestal te herkennen aan dat het blok paginabreed is en er margin zit tussen het volgend blok.

### 4.2.2 Designers

De designers ontwerpen een design in Figma en sturen dit naar de front-end developers. De designers hebben geen specifieke werkwijze, zoals het werken vanuit mobile-first. Als er aan een project begonnen wordt, beginnen designers eerst met een ontwerp en achteraf wordt er een styleguide opgezet. Hierdoor lopen dingen nog weleens mis. Daarnaast verwachten de developers af en toe een ander ontwerp, doordat ze bijvoorbeeld een ander framework gebruiken.

Een project kan aangenomen worden door het team van design, maar kan ook vanuit het team van development zijn aangenomen. De designers teken in Figma, niet in code. Er is geen link van een UI-kit die vanuit development komt in Figma kan stoppen. De designers kunnen al een beetje voorsorteren met wat ze gaan doen met bijvoorbeeld de head-spacing en de font-family etc. Designers vergeten soms de state van een button zoals de `::hover` of de `::press`<sup>2</sup>. Designers krijgen hierdoor irritaties naar het designteam, doordat zij terug moeten naar het designteam en er veel tijd overheen gaat zitten. Hier is momenteel geen goede workflow voor. Hier loopt het mis en ontstaat het probleem tussen developers en designers.

### 4.3 Problemen huidige manier van werken

#### 4.3.1 Developers

##### **De inconsistentie in het design**

Het is de bedoeling dat designers ook gebruik maken van ‘variabelen’ of ‘tokens’, maar in de praktijk wanneer developers het design implementeren, zijn elementen niet gekoppeld. Hierdoor moeten developers zelf uitzoeken wat bijvoorbeeld de typografie is. Af en toe worden er ook fouten gemaakt in het ontwerp waardoor de developers verkeerde variabelen in de code meenemen.

##### **De onvolledigheid**

Naast het missen van variabelen en gedefinieerde componenten, ontbreken er ook vaak states van buttons en formulier velden. Er wordt niet genoteerd wat gebeurt er als je focust op een button, dus als je met je toetsenbord navigeert door de website. Daarnaast mist er vaak een Error State bij input's of een bevestiging als een formulier is ingevuld.

##### **Verschillende werkwijzen**

Er wordt geprobeerd wel eens tegelijk aan een project te beginnen, maar vaak moeten developers dan te lang wachten voordat designers iets kunnen opleveren. Designers willen altijd eerst een hele pagina ontwerpen terwijl developers werken in componenten. Daarnaast gebeurt het vaak dat designers al niet meer op het project gepland staan waardoor developers geen feedback meer kunnen geven als er nog wat mist.

##### **Geen tokens**

Developers moeten momenteel uit het Figma bestand de variabelen halen. De developers willen graag dat ze JSON-tokens handmatig uit het Figma bestand kunnen halen of dat het automatisch naar de GitHub wordt verstuurd. Dit scheelt veel tijd.

#### 4.3.2 Designers

##### **Geen designsysteem aanwezig**

Er is geen momenteel geen designsysteem aanwezig, wel richtlijnen. Momenteel zetten de designers de ontwerpen in Figma, waar de developers een link van krijgen om de componenten te kunnen zien. Daarnaast staat in Figma een styleguide waar de developers informatie uit kunnen halen, maar wordt deze styleguide pas achteraf gemaakt waardoor er veel fouten inzitten. De variabelen zijn niet

---

<sup>2</sup> Bepaalde states bij een knop wanneer er overheen “gehangen” of opgedrukt wordt met de muis.

gedefinieerd en vaak missen er delen uit het project waar developers dan achteraan moeten. Ook moeten de tokens uit het bestand gehaald worden en worden deze niet naar de GitHub gepusht.

### Verschillende werkwijzen

Het development team gaf het ook al aan, maar designers en developers hebben verschillende werkwijzen. Designers maken een gaaf ontwerp en sturen dit naar de developers; developers hadden iets anders verwacht, want het ontwerp is niet opgezet hoe zij het in het framework hebben. Er wordt niet mobile-first gedacht vanuit de designers. Daarnaast worden er onnodige fouten gemaakt door laksheid of gebrek aan kennis over het opbouwen van een goed design. Er zijn geen concrete afspraken die wel nodig zijn bij het werken met verschillende teams.

### 4.3.3 SWOT-analyse

Uit Hoofdstuk 4.3.1 en 4.3.2 is naar voren gekomen welke punten er belangrijk zijn om terug te laten komen in de manier van werken. Hiermee is een SWOT-analyse opgesteld met de kansen, bedreigingen, sterktes en zwaktes. Als we alles onder elkaar zetten komt het erop neer dat het probleem is dat er geen designsysteem aanwezig is en dat er verschillende werkwijzen zijn doordat beide bedrijven bij Makerstreet zijn aangesloten, maar geen directe collega's zijn. De designers willen bij het ontwerpen hun vrijheid behouden, een automatische tokenflow gegenereerd krijgen en gerichte afspraken met de developers maken. Developers moeten duidelijk rond hebben wat zij willen ontvangen van de designers. Momenteel zijn developers het vaak onderling nog oneens over het ontvangen van een design. Developers willen graag tokens van het design ontvangen. Ook willen ze dat alle variabelen gelabeld zijn en van naam zijn voorzien. Momenteel is er geen designsysteem, alleen richtlijnen die niet altijd nagestreefd worden

<b>Sterkten</b> <ul style="list-style-type: none"><li>• Enkele weten de basis van het opstellen van een style guide</li></ul>	<b>Zwakten</b> <ul style="list-style-type: none"><li>• Onderlinge communicatie tussen developers en designers</li><li>• Bijhouden van SCRUM</li><li>• Geen gebruik maken van juiste naamgeving</li></ul>
<b>Bedreigingen</b> <ul style="list-style-type: none"><li>• Te lang over een project doen</li><li>• Opleveren onjuiste componenten</li></ul>	<b>Kansen</b> <ul style="list-style-type: none"><li>• Opstellen duidelijke regels</li><li>• Variabelen definiëren en consistent koppelen</li><li>• Automatisch geëxporteerd en gekoppeld worden aan develop omgeving</li></ul>

### 4.3.4 Customer Journey

Er is een Customer Journey opgesteld over de developers van Touchtribe. Als eerst is de lifecycle stage te vinden. Deze is opgebouwd uit het opzetten van een project, het ontvangen van een ontwerp, de terugkoppeling met ontwerpers en het uitwerken van het project. Bij het opzetten van het project is het een uitdaging om de verschillende werkwijzen van de designers en developers te laten samenwerken. In het begin van het project zijn er nog geen tot weinig ergernissen. Er wordt



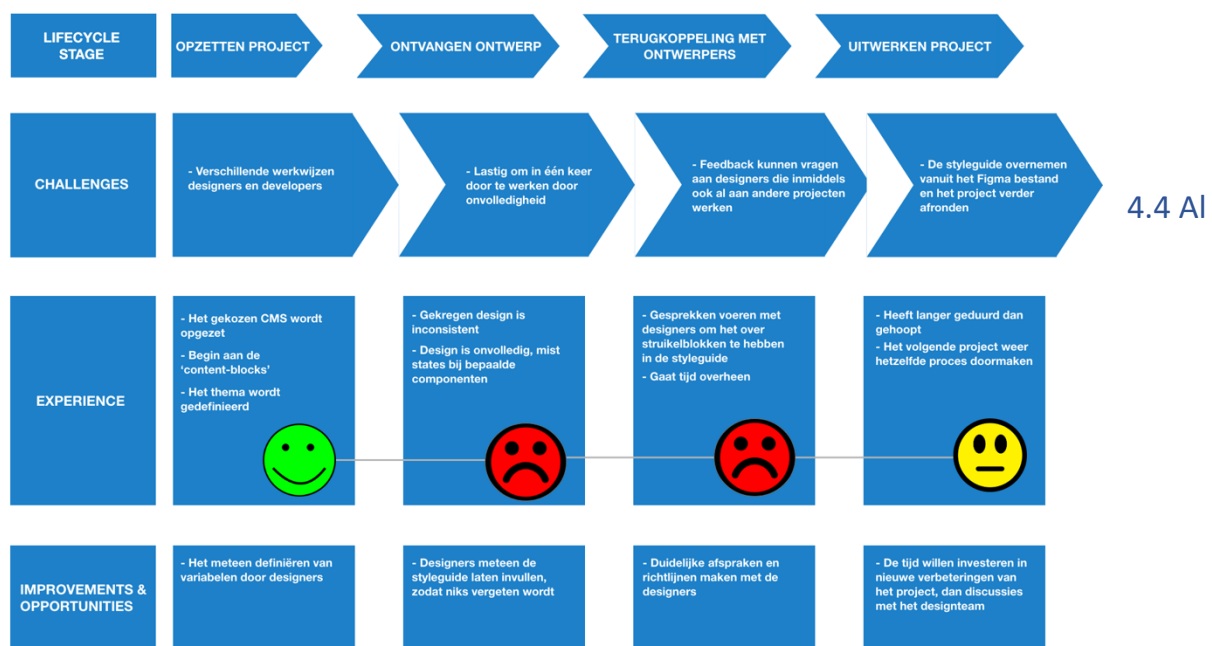
een CMS gekozen en opgezet. Vervolgens wordt er begonnen aan de 'content-blocks' en wordt het thema gedefinieerd. De kansen die bij het opzetten van een project worden gezien zijn het meteen definiëren van variabelen door designers.

De tweede fase is het ontvangen van het ontwerp. De uitdaging is daar dat het lastig is om in één keer door te kunnen werken door de onvolledigheid van het design. Hier beginnen de irritatie op te komen. Het gekregen design is inconsistent. Het design is onvolledig, zoals het missen van states bij bepaalde componenten. Dit kan opgelost worden door de designers meteen de styleguide te laten invullen, zodat er niks vergeten wordt.

In de derde fase wordt er teruggekoppeld met ontwerpers. Feedback vragen aan designers is hier vaak niet meer mogelijk, door dat de designers al aan een ander project werken. Het kost veel tijd om contact te houden met de designers over een onvolledig design. Door duidelijke afspraken en richtlijnen kan dit voorkomen worden.

De laatste fase is het uitwerken van het project. De styleguide wordt overgenomen vanuit het Figma bestand en het project wordt afgerond. Het project heeft langer geduurd dan gehoopt. Het volgende project gaat misschien wel net zo. De verloren tijd kan beter geïnvesteerd worden in nieuwe verbeteringen van het project, dan discussies over het design.

#### CUSTOMER JOURNEY: DEVELOPERS TOUCHTRIBE



4.4 AI

## aanwezige wensen en behoeften

### 4.4.1 Developers

#### Figma

De designers werken met de tool Figma die het mogelijk maakt om design systems op te zetten. Er wordt een styleguide opgezet met daarin de basis zoals kleuren, typografie, layout, icons en de basiscomponenten. Deze styleguide wordt niet meteen, maar pas achteraf gemaakt. Hierdoor krijg je het gevaar dat veel niet goed gekoppeld wordt.

### **Betrekken developers**

Er wordt geprobeerd om developers steeds vroeger bij het designproces te betrekken zodat ze tijdig feedback kunnen geven, maar dit gaat niet altijd even goed doordat de communicatie niet helemaal soepel verloopt.

### 4.4.2 Designers

#### **Inhoud**

Er wordt per project gekeken naar de inhoud van het project. Bij het toepassen van kleuren en typografie lukt het allemaal nog wel, maar bij het opmaken van componenten en pagina's gaat het mis; niet consequent de componenten gebruiken die in design al gebruikt worden of nog helemaal niet gemaakt zijn.

### 4.5 Wensen en behoeften

#### 4.5.1 Developers

##### **Gedefinieerde basics**

Een van de wensen van de developers zijn dat alle basics compleet worden gedefinieerd in een styleguide. Op deze manier wordt er voorkomen dat developers verkeerde variabelen in de code doorvoeren of dat zij zelf achter een niet geheel ontwerp moeten gaan. Dit kost beide veel tijd.

##### **Variabelen gekoppeld**

Een andere wens van de developers is dat alle variabelen consistent worden toegepast en gekoppeld in Figma. Alle variabelen hebben een correcte naam en bevatten de juiste gegevens.

##### **Componenten voorzien naamgeving**

Componenten en variabelen worden voorzien van naamgeving. Vaak ontbreken er namen, doordat de designers deze grofweg niet toevoegen.

##### **Automatisch geëxporteerd naar development omgeving**

Het designsysteem moet automatisch geëxporteerd en gekoppeld worden met de development omgeving, ofwel het moet verstuurd kunnen worden naar GitHub. De JSON-tokens kunnen ook handmatig uit het Figma bestand kunnen worden gehaald, maar het liefst wil development het naar de GitHub geëxporteerd krijgen. Dit is een erg belangrijke wens en behoeften vanuit Touchtribe.

### 4.5.2 Designers

#### **Vrijheid**

Een designer wilt vrij zijn om te kunnen ontwerpen wat hij wilt. Als er een lijst is met allerlei buttons, moeten designers deze nog op hun eigenmanier kunnen ontwerpen.

##### **Automatische tokenflow**

Designers zouden het op prijs stellen als er een automatische tokenflow komt, wat veel wrijving tussen designers en development zou schelen.

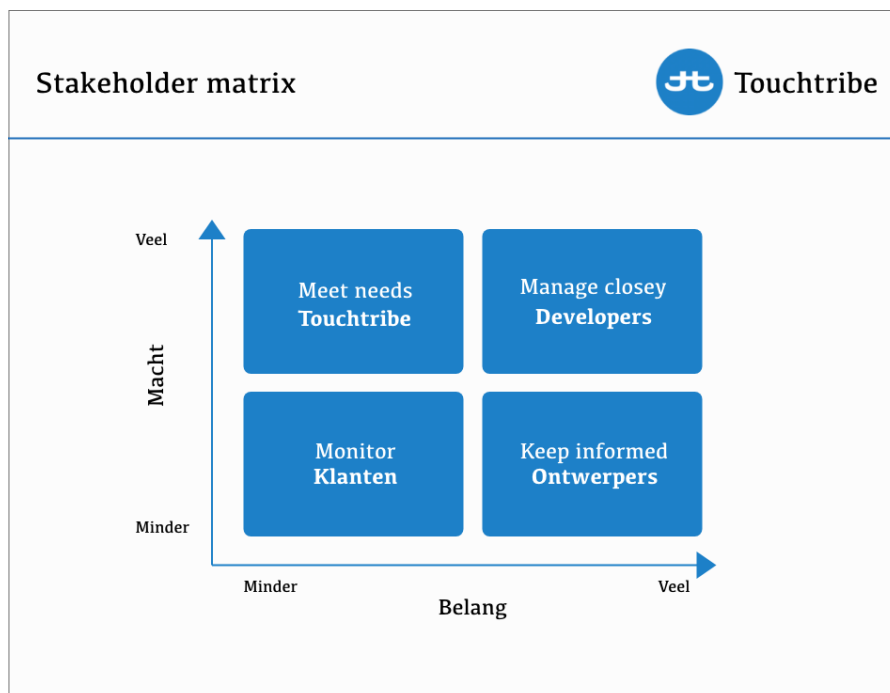
### Gerichte afspraken

Er moeten afspraken gemaakt worden wanneer bijvoorbeeld er een button ontworpen wordt, dat het altijd diezelfde border-radius definieert, ofwel een soort boilerplate. Een boilerplate is een korte tekst die zonder veel aanpassingen kan worden hergebruikt (Kuipers, 2022). De boilerplate is in dit geval bedoelt voor design dat altijd gebruikt wordt. Op deze manier hoeft er vanuit de ontwerpkant geen rekening meer gehouden te worden of het een Shopify project of een Wordpress project wordt. Daarnaast heeft het development team vaak nog niet rond wat ze precies willen bij het geven van benamingen en afmetingen, zoals de breakpoints. Breakpoints zijn punten waar de inhoud van de website reageert om de beste lay-out voor de gebruiker te bieden om informatie te verkrijgen (Responsive Design, sd).

## Hoofdstuk 5 Methoden en technieken

### 5.1 Stakeholderanalyse

Voorafgaand is een stakeholderanalyse gemaakt. Touchtribe heeft veel macht, maar weinig belang doordat het een organisatie is die mensen in dienst hebben die bij het probleem betrokken zijn. Developers hebben veel macht en veel belang doordat zij bij het bedrijf Touchtribe horen en dagelijks met dit probleem te werk moeten. Klanten hebben weinig macht en weinig belang, omdat zij niet veel meekrijgen van het probleem waar de developers en ontwerpers tegen aanlopen. De ontwerpers hebben minder macht en veel belang, doordat zij aangesloten zijn bij andere organisaties zoals Resoluut en Elements. Touchtribe, Resoluut en Elements zijn allemaal aangesloten Makerstreet. Hierdoor moeten zij samenwerken met Touchtribe, maar zijn het geen directe collega's van elkaar. De ontwerpers hebben dus veel belang, omdat zij ook tegen de problemen aan lopen maar hebben minder macht dan de developers van Touchtribe, zie *Afbeelding 6 Stakeholderanalyse*.



Afbeelding 6 Stakeholderanalyse

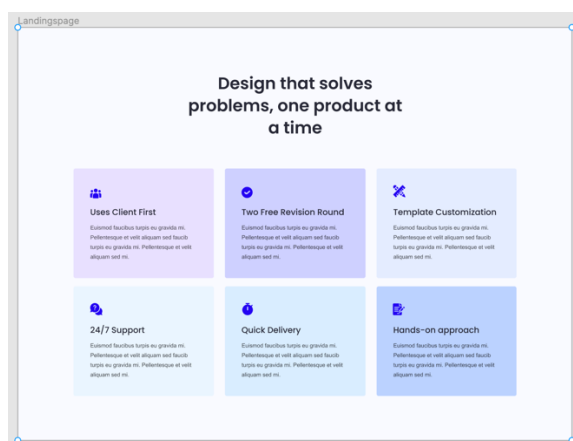
## 5.2 Nul- en éénmeting

Touchtribe heeft aangegeven een automatische tokenflow te willen. Naast deze richtlijn is het ook belangrijk om de developer experience te verbeteren. Om de doelstelling te behalen is er een nul- en éénmeting uitgevoerd. Een nulmeting brengt de huidige situatie in kaart en is de eerste stap in het verbeter- en veranderproces. Met de nulmeting wordt de startpositie bepaald. Om de doelstelling te kunnen behalen is het belangrijk om te weten waar Touchtribe op moment staat. De nulmeting is bedoeld om duidelijke resultaten te creëren en doelgericht te werken aan het behalen van de doelstelling.

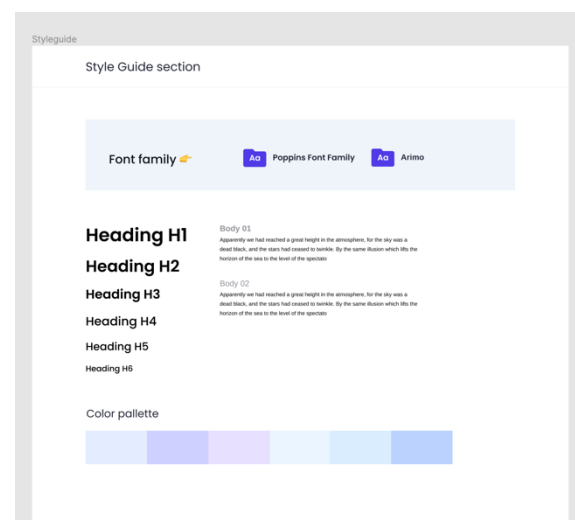
In deze metingen is er gebruik gemaakt van een opgeleverd ontwerp van een van designers. Dit ontwerp bevat alleen een eenvoudige landingspagina, zie *Afbeelding 7 Eenvoudige landingspagina*. Het Figma bestand is te vinden in *Bijlage 3 Figma metingen*. Ik heb dit ontwerp nagebouwd in React om vervolgens de kleur en typografie toe te voegen door middel van de metingen. Bij de nulmeting wordt er gekeken naar de tijd die ik erover doe om de variabelen handmatig uit het Figma bestand

te halen en vervolgens toe te passen in de code, om het Figma bestand identiek aan de website te maken. In *Bijlage 4 Nul- en éénmeting* zijn de videobestanden te vinden. Deze bijlage bevat een video opnamen waar de variabelen kleur en typografie uit het Figma bestand worden gehaald en vervolgens worden toegepast in de code, als bewijs van deze nulmeting. Voor dit Figma bestand is er gebruik gemaakt van een font-family en font-weight.

Bij de éénmeting wordt er gekeken naar de tijd die de testpersonen erover doen om de variabelen uit het Figma bestand te halen. In Figma wordt de plug-in Figma Tokens gebruikt. Vervolgens worden de variabelen door het script de "Token-Transformer" gehaald. Het script maakt er constante variabele van die vervolgens toegepast kunnen worden in de code. In *Bijlage X Éénmeting* is het videobestand te vinden. Dit bevat een video opnamen waar de variabelen kleur en typografie uit het Figma bestand worden gehaald, door het script worden gehaald en vervolgens worden toegepast in de code, als bewijs van deze éénmeting. Voor dit Figma bestand is er gebruik gemaakt van een font-family en font-weight. Er wordt uiteindelijk gekeken of door middel van het script de developer experience wordt verbeterd, dus dat de developer minder tijd kwijt is met de automatische tokenflow.



Afbeelding 7 Eenvoudige landingspagina



## 5.2.1 Resultaten metingen

### 5.2.1.1 Nulmeting

#### Robin van Vliet

De tijd die ik erover deed om de variabelen uit Figma te halen en toe te passen in de code is 10 minuten en 12 seconden. Hiermee haalde ik de variabelen uit het Figma bestand, maakte ik zelf voor elke variabele kleur, font-family en font-weight een constante aan. In deze constante werden handmatig de variabelen toegevoegd. Hierna konden ze worden toegevoegd aan de rest van de code door middel van props.

#### Tim Huijg

De tijd die Tim er bij de nulmeting overdeed was 13 minuten en 46 seconden. Het cijfer wat de nulmeting kreeg was een 6. Tim vond dit werk erg saai, omdat het alleen maar variabelen uit een bestand halen is.

### **Linda van Dijk**

De tijd die Linda er bij de nulmeting over deed was 14 minuten en 39 seconden. Het cijfer wat de nulmeting kreeg was een 6. Het is nu erg standaardwerk dat verricht wordt bij het begin van een project. Het moet gebeuren geeft Linda aan, maar het is op deze manier wel zinlozer met tijd omgaan.

### **Jasper van Zuyderwijk**

De tijd die Tim er bij de nulmeting overdeed was 13 minuten en 3 seconden. Het cijfer wat de nulmeting kreeg was een 6. “Het handmatig variabelen uit Figma halen is een proces wat we allemaal al kennen. Het is iets wat je in de beginfase doet. Je bent constant aan het kopiëren en plakken uit een Figma design. Dit duurt lang, is saai en je hoeft er niet veel bij na te denken. Het is alleen wel vervelend dat je constant repetitief werk moet verrichten.”

#### **5.2.1.2 Éénmeting**

### **Robin van Vliet**

De tijd die ik erover deed om de variabelen uit het script te halen en toe te passen in de code is 6 minuten en 46 seconden. Hiermee haalde ik de variabelen uit het Figma bestand en zette het script deze om naar constante variabelen. Deze variabelen kopieerde ik in de code en werden verder toegepast door middel van props.

### **Tim Huijg**

De tijd die Tim er bij de éénmeting over gedaan had was

### **Linda van Dijk**

De tijd die Linda er bij de éénmeting over deed was 11 minuten en 50 seconden. Linda gaf het gebruik van de Token-Transformer een 8,5. “Deze transformer is erg waardevol. Het geeft consistentie aan de developers. Het is erg gaaf dat dit nu geautomatiseerd is en zeker erg bruikbaar. Een hele vette ontwikkeling voor het developer-team van Touchtribe”.

### **Jasper van Zuyderwijk**

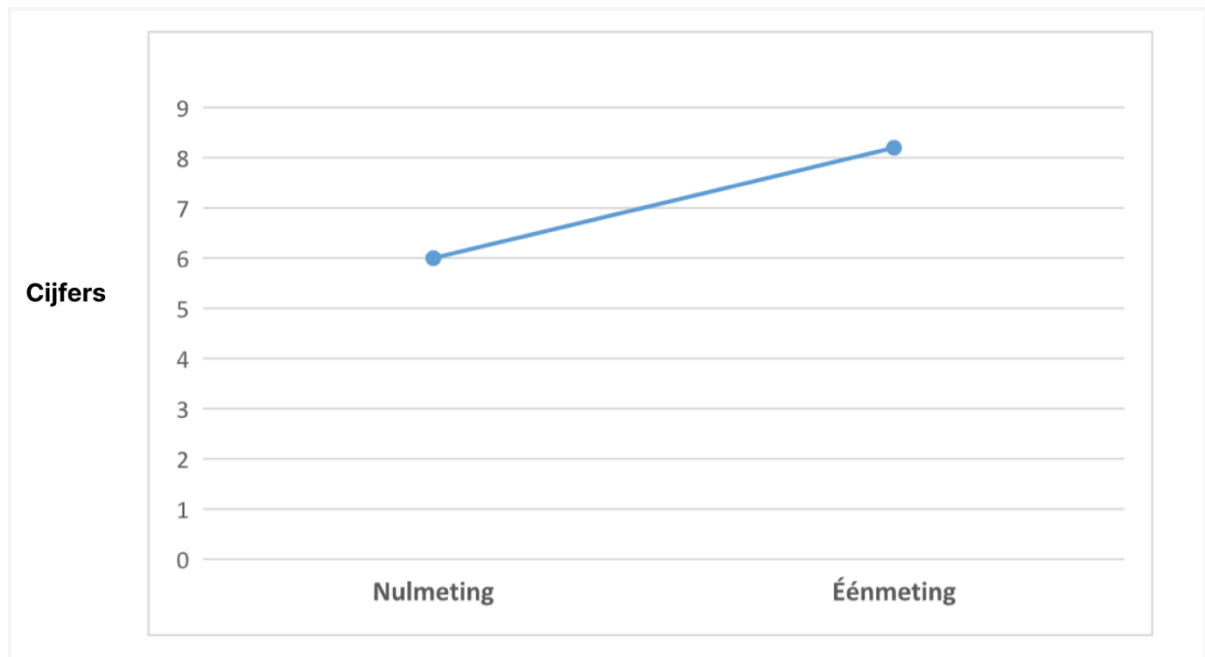
De tijd die Jasper bij de éénmeting nodig had was 9 minuten en 10 seconden. Het cijfer wat de nulmeting kreeg was een 8. “Het repetitieve werk is nu goed geautomatiseerd. Dit is erg fijn werken, omdat het tijd bespaard. Ik vind het erg gaaf.”

#### **5.2.2 Conclusie**

Door gebruik te maken van het script de Token-Transformer heb ik als developer 3 minuten en 27 seconden bespaard. De andere developers zaten ook op een gemiddelde van 3 minuten. Enkele factoren konden meespelen met deze uitkomst:

- Eigenbestand; Doordat ik deze metingen zelf heb opgezet, heb ik dit bestand al een aantal keer doorgenomen en ben ik er mee bekend. Dit kan mij tijd gescheeld hebben met het coderen van de variabelen.
- Nieuw bestand; De testpersonen kregen die bestand voor het eerst te zien en konden niet van tevoren kijken welke manier van coderen zij prettig vonden bij het toepassen van deze casus. Daardoor doen hun er bij beide metingen gemiddeld iets langer over.
- Klein bestand; Dit test bestand is er minimaal. Bij grote projecten zijn er soms wel 20 kleuren of verschillende typografieën. Als ik deze allemaal handmatig erin zou moeten zetten ben ik ook langer bezig en zou het verschil uiteindelijk ook groter zijn.

Zoals in *Afbeelding 8 Verbetering tokenflow* is te zien, is de automatisering van de tokenflow van een gemiddelde 6 naar een gemiddelde 8,2 gegaan. De automatisering van de tokenflow was een van de grootste wensen van Touchtribe.



*Afbeelding 8 Verbetering tokenflow*

### 5.3 Opzetten designsysteem

#### Afspraken binnen Touchtribe

Om te een goed designsysteem neer te kunnen zetten, moest er een aantal afspraken gemaakt worden binnen Touchtribe. Ik ben samen met Linda, Maarten en Daan, een andere developer rond de tafel gaan zitten. We zijn met zijn vieren in gesprek gegaan over wat zij daadwerkelijk wilden van het ontwerpteam, omdat elke developer van Touchtribe iets anders wilde. Er moest bijvoorbeeld besproken worden of er in de JSON-tokens gebruik gemaakt moest worden van breekpunten of hoofdletters. Uiteindelijk is het development na een aantal meetings er niet uitgekomen of zij de breekpunten wilde aangeven of dat de designers dat zelf mochten doen. Wel is er naar voren gekomen dat zij op gestructureerde wijze en volgens de designregels kleuren wilde ontvangen, zoals in Hoofdstuk 3.4.3 Kleuren wordt beschreven. Daarnaast willen de developers de variabelen in de

Screaming Snake Case ontvangen. De Screaming Snake Case houdt in dat er gebruik wordt gemaakt van hoofdletters en een underscore-teken<sup>3</sup> voor woordscheiding (Fission, 2021).

### Opstellen richtlijnen

De richtlijnen die uit de meetings zijn gekomen zijn als volgt:

- Benaming tokens:  
De tokens moeten een benaming in de Screaming Snake Case krijgen. Dit houdt in dat alles in hoofdletter moet en bij woordscheiding een underscore-teken gebruikt wordt. Alles moet gelabeld worden. Dit is in het verleden niet gebeurd.
- Categorie in benaming:  
Er moet gebruik gemaakt worden van categorie in de benaming van de variabelen. Een voorbeeld is: colors/PRIMARY\_COLOR. Op deze manier worden de tokens gefilterd in het script om de tokens om te zetten naar bruikbare design tokens voor de developers.
- Kleuren gerepresenteerd op basiskleur:  
De kleuren moeten gerepresenteerd worden op basiskleur volgens ontwerpprincipes. Hier is meer uitleg over gegeven in Hoofdstuk 3.4.3 Kleuren.
- Figma Tokens:  
Er wordt gebruik gemaakt van de plug-in Figma Tokens om de tokens uiteindelijk in JSON-tokens te kunnen versturen naar de development.
- Mobile first werken:  
Bij het ontwerpen moeten er mobile first gewerkt worden. Eerst de ontwerpen voor mobiel en dan uitwerken voor andere schermen.
- Gebruiken maken van sprints:  
Er moet duidelijk aangegeven worden wanneer een design of een stuk daarvan af is, zodat het development team hier al mee aan de slag kan.

### Opzetten styleguide

Om de designers vrij te laten in hun werk mogen zij zelf de lay-out van de styleguide kiezen. Wel moeten zij zich aan de richtlijnen die opgesteld zijn vanuit het development team. Om een richting te geven hoe Touchtribe het wil heb ik een prototype opgesteld, zie *Bijlage 5 Prototype designsysteem*. In dit prototype zie je dat alle tokens voorzien zijn van naam. Er wordt gebruik gemaakt van de Screaming Snake Case en kleuren krijgen een benaming op basis van basiskleur. De plug-in automatiseert alleen kleuren, typografie en box-shadow. De border-radius moet er handmatig in de plug-in worden toegevoegd.

### Designsysteem verbinden met GitHub

Een van de wensen van de developers was dat het designsysteem automatisch geëxporteerd moest kunnen worden naar de development-omgeving. Hier ben ik mee aan de slag gegaan en had de plug-in Design Tokens voor Figma gevonden. Na enkele testen lukte het om hem te verbinden met GitHub. Ook kon deze plug-in een zipfile van de JSON-tokens leveren. Toen ik hiermee naar het ontwerpsteam ging, gaven zij aan in een project buiten Touchtribe een plug-in te gebruiken genaamd Figma Tokens. Deze plug-in mochten zij niet binnen projecten van Touchtribe gebruiken, doordat een van de developers hierop tegen was. Ik heb uiteindelijk bij Maarten voorgelegd dat het gebruiken van een van deze tools wel degelijk handig is te gebruiken. Als eerst moesten de plug-ins onder de loep genomen worden en moest er bekeken worden welke van de twee kwalitatief beter was.

---

<sup>3</sup> Een “\_” teken.



### **Figma Tokens**

Figma tokens is een Figma Plug-in waarmee je tokens kunt integreren in een Figma-ontwerp. Het geeft herbruikbare tokens die kunnen worden gebruikt voor een reeks ontwerpopties zoals ontwerpopties, afstandseenheden en typografische stijlen. Hiermee kunnen de tokens gewijzigd worden en worden deze wijzigen in het hele document aangepast (Figma, 2022). Zie Hoofdstuk 5.4 Figma Tokens; de Token-Transformer voor uitleg over het script de Token-Transformer.

### **5.4 Figma Tokens; de Token-Transformer**

Een grote wens van Touchtribe is het automatiseren van de tokenflow. Met het script de Token-Transformer, hoeven de developers niet meer handmatig de tokens uit het Figma bestand te halen. Deze tokens hoeven nu alleen maar opgehaald te worden met de plug-in Figma Tokens en door het script gehaald te worden.

#### **Script omzetten naar bruikbare code**

Touchtribe wilt de tokens op de Screaming Snake Case manier ontvangen. Daarnaast willen ze het als Javascript variabelen ontvangen. In *Bijlage 1 Prototype* is te zien hoe een designsysteem er in de toekomst uit moet gaan zien. Dit geeft de volgende code uit de plug-in, zie *Afbeelding 9 Designtokens*.

```
{
  "colors": {
    "SECONDARY_TAP": {
      "value": "#c95501",
      "type": "color"
    },
    "SECONDARY_DEFAULT": {
      "value": "#7d0000",
      "type": "color"
    }
  },
```

```
"boxshadow": {
  "CARD": {
    "value": {
      "color": "#0000001a",
      "type": "dropShadow",
      "x": 0,
      "y": 4,
      "blur": 20,
      "spread": 0
    },
    "type": "boxShadow"
  }
},
```

```
"typography": {
  "HEADING_01": {
    "value": {
      "fontFamily":
"$fontFamilies.barlow",
      "fontWeight":
"$fontWeights.barlow-0",
      "lineHeight":
"$lineHeights.0",
      "fontSize":
"$fontSize.4",
      "letterSpacing":
"$letterSpacing.0",
      "paragraphSpacing":
"$paragraphSpacing.0",
      "textCase":
"$textCase.none",
      "textDecoration":
"$textDecoration.none"
    },
    "type": "typography"
  },
```

```
"borderradius": {
  "X-SMALL": {
    "value": "0px",
    "type": "borderRadius"
  },
  "SMALL": {
    "value": "4px",
    "type": "borderRadius"
  },
```

Afbeelding 9 Designtokens

Deze code wordt door een script gehaald genaamd, design.js, die gebruik maakt van Style Dictionary en Token-Transformer. Style Dictionary is een systeem waarmee stijlen in één keer gedefinieerd kunnen worden op een manier door elk platform of taal gebruikt kan worden. Ze kunnen geëxporteerd worden naar iOS, Android, CSS, JS, HTML, schetsbestanden, stijldocumentatie etc. Het is beschikbaar via CLI naar npm (Amzn GitHub, sd). Token-Transformer converteert tokens van Figma Tokens naar iets dat Style Dictionary kan lezen, waarbij alle wiskundige bewerkingen of aliassen worden verwijderd. Dit resulteert in onbewerkte waarden (Npm, 2022).

Style Dictionary heeft meerdere formats die te vinden zijn op GitHub waar er al bestaande formats gemaakt zijn. Het format ES6 leek erg op wat Touchtribe wilde, maar er ontbrak nog het een en ander. Zo bracht dit script uiteindelijk wel een constante variabele wat Touchtribe graag wilt, zie *Afbeelding 10 ES6 module*. Nu is uiteindelijk dit script zo omgeschreven dat er een constante variabele wordt aangemaakt met het type (color, typography, border-radius of border-shadow). Deze types worden omgezet in hoofdletters waar naar ze bij meerdere values van één type onder elkaar worden gezet.

```

332 * Creates a ES6 module of the style dictionary.
333 *
334 * ```` json
335 * {
336 *   "platforms": {
337 *     "js": {
338 *       "transformGroup": "js",
339 *       "files": [
340 *         {
341 *           "format": "javascript/es6",
342 *           "destination": "colors.js",
343 *           "filter": {
344 *             "attributes": {
345 *               "category": "color"
346 *             }
347 *           }
348 *         }
349 *       ]
350 *     }
351 *   }
352 * }
353 * ````
354 *
355 * @memberof Formats
356 * @kind member
357 * @example
358 * ```` js
359 * export const ColorBackgroundBase = '#ffffff';
360 * export const ColorBackgroundAlt = '#fcfcfc';
361 * ````
362 */
363 'javascript/es6': function({dictionary, file}) {
364   return fileHeader({file}) +
365     dictionary.allTokens.map(function(token) {
366       var to_ret = 'export const ' + token.name + ' = ' + JSON.stringify(token.value) + ' ';
367       if (token.comment)
368         to_ret = to_ret.concat(' // ' + token.comment);
369       return to_ret;
370     }).join('\n');
371 },
372
373 // TypeScript declarations

```

### Afbeelding 10 ES6 module

Er moeten twee bestanden naast het design.js bestand gemaakt worden, namelijk; input.tokens.json en output.tokens.json. In de input.tokens.json moeten de tokens volgens *Afbeelding 9 Designtokens* geplaatst worden. In de terminal moet dan het commando ‘npx token-transformer input.tokens.json output.tokens.json’ worden uitgevoerd. Het resultaat is te zien in output.tokens.json. Deze transformer filtert alle categorieën of ook wel types. Vervolgens moet er in de terminal het commando ‘node design.js’ worden uitgevoerd. Het script design.js wordt dan uitgevoerd. Zie *Afbeelding 11 Design.js* voor een deel van de code uit het script, waarnaar de variabelen de Screaming Snake Case meekrijgen en onder elkaar gezet worden. Na het uitvoeren van de node design.js worden alle tokens in het bestand allTokens.js worden geplaatst op de gewenste manier, zie *Afbeelding 12 allTokens.js*

```

StyleDictionaryPackage.registerFormat({
  name: 'javascript/color-const',
  formatter: function({dictionary, file}) {
    var styles = {};

    dictionary.allTokens.forEach(function(token) {
      var to_ret = ' ' + snakeCase(token.name).toUpperCase() + ': ' + JSON.stringify(token.value, null, 2) + ',';
      if (token.comment)
        to_ret = to_ret.concat(' // ' + token.comment);

      if (!styles[token.type])
        styles[token.type] = [];

      styles[token.type].push(to_ret);
    });

    var output = fileHeader({file});
    for (const [type, values] of Object.entries(styles)) {
      output += `const ${type.toUpperCase()} = {\n${values.join("\n")}\n}\n\n`;
    }

    return output;
  }
});

```

### Afbeelding 11 Design.js

```
const COLOR = {
  SECONDARY_TAP: "#c95501",
  SECONDARY_DEFAULT: "#7d0000",
  SECONDARY_SELECT: "#eaa81b",
  PRIMARY_TAP: "#0a3c15",
  NEUTRAL_WHITE: "#ffffff",
  DISCOUNT_100: "#faaa00",
  DISCOUNT_110: "#ffc446",
  PRIMARY_PINK: "#ff6262",
  NEUTRAL_90: "#3c6e57",
  NEUTRAL_100: "#469b4c",
  NEUTRAL_110: "#52c25a",
  NEUTRAL_120: "#4bcc54",
  FEEDBACK_SUCCES_LIGHT: "#95e3a7",
  FEEDBACK_SUCCES: "#2fbc4e",
  FEEDBACK_SUCCES_DARK: "#074e17",
  FEEDBACK_WARNING_LIGHT: "#ffdc5f",
  FEEDBACK_WARNING: "#ffc90c",
  FEEDBACK_WARNING_DARK: "#c88a07",
  FEEDBACK_ERROR_LIGHT: "#ff9696",
  FEEDBACK_ERROR: "#fa5e5e",
  FEEDBACK_ERROR_DARK: "#fe0000",
  PRIMARY_DEFAULT: "#012c0b",
  PRIMARY_SELECT: "#33613e",
}
```

```
const TYPOGRAPHY = {
  HEADING_01: {
    "fontFamily": "$fontFamilies.barlow",
    "fontWeight": "$fontWeights.barlow-0",
    "lineHeight": "$lineHeights.0",
    "fontSize": "$fontSize.4",
    "letterSpacing": "$letterSpacing.0",
    "paragraphSpacing": "$paragraphSpacing.0",
    "textCase": "$textCase.none",
    "textDecoration": "$textDecoration.none"
  },
  HEADING_02: {
    "fontFamily": "$fontFamilies.barlow",
    "fontWeight": "$fontWeights.barlow-0",
    "lineHeight": "$lineHeights.1",
    "fontSize": "$fontSize.3",
    "letterSpacing": "$letterSpacing.1",
    "paragraphSpacing": "$paragraphSpacing.0",
    "textCase": "$textCase.none",
    "textDecoration": "$textDecoration.none"
  },
}
```

```
const FONTSIZES = {
  0: 12,
  1: 16,
  2: 18,
  3: 24,
  4: 32,
  5: 40,
  6: 56,
  7: 160,
}
```

Afbeelding 12 allTokens.js

## 5.5 Opzet usability test designers

### Opzet

Voor de usability testen had ik met Vince van het designteam overlegt dat ik 5 testpersonen nodig had. Ik wilde graag een gevarieerde groep van junior designers tot senior designers, in hoeverre dit mogelijk was. Uiteindelijk zijn het 2 vrouwen en 3 mannen geworden, waarbij er 3 senior waren en 2 junior. Na de agenda's van de testpersonen naast elkaar gehouden te hebben had ik een indeling gemaakt. Er was een e-mail naar de testpersonen gestuurd en de dag daarna kon ik met alle testpersonen testen. Bijna iedereen werkte vanaf huis, dus zijn de testen online gedaan via Google Meetings en opgenomen, zie *Bijlage 7 Resultaten usability test designers* voor de opnames.

### Casus

Ik heb voor deze testen een prototype gemaakt met de richtlijnen vanuit Touchtribe. Voordat de testen begonnen ben ik deze met de testpersonen gaan doornemen, zodat ze een indicatie kregen over de richtlijnen. Toen ik de casus voorlegde, hebben de testpersonen voordat de opdrachten

begonnen een korte introductie over de nieuwe onderwerpen gekregen, zie *Bijlage 6 Casus designers*. Ik wist niet of deze onderwerpen wellicht nieuw waren voor de testpersonen. De onderwerpen waren: kleurentinten, Screaming Snake Case, layers labelen en Figma Tokens. Hiervoor zijn afbeeldingen en tekstuele uitleg gebruikt. De kern van de testen ging voornamelijk over het opstellen van een duidelijke styleguide. Ze moesten hiervoor kleine opdrachten doen door middel van een stappenplan om te kijken of ze begrepen wat ze deden en waarom ze het deden. Het ging in de casus niet om het opbouwen van frames en componenten, maar om het labelen van de juiste elementen en het uiteindelijk kunnen versturen naar development. De verdere opmaak van de styleguide was aan de testpersonen zelf. Na de casus werden er nog enkele vragen gesteld wat ik mee kon nemen in het verbeteren van de richtlijnen. De casus was opgedeeld in de onderwerpen: colors, typography, box-shadow, borderradius en Figma Tokens. Als voorbeeld pak ik even colors;

### **Stap 1**

De eerste stap is een styleguide maken met 3 willekeurige kleuren. Het moet het volgende bevatten:

- Geef ze de juiste categorie (colors/)
- Noem ze primary default, primary tap en primary select. Hiermee geef je ze meteen de juiste variabelen.
- Voeg deze styles hierna toe: (voorbeeld) colors/PRIMARY\_DEFAULT

### **Stap 2**

De tweede stap is een styleguide maken met 3 tinten (je mag ze zelf kiezen) en geef ze meteen de juiste variabelen:

- Geef ze de juiste categorie (colors/)
- Noem ze neutral.90, neutral.100 en neutral.110. Hiermee geef je ze meteen de juiste variabelen.
- Voeg deze styles hierna toe: (voorbeeld) colors/NEUTRAL\_100

Ik legde uit dat deze categorieën de belangrijkste voor development waren. Als deze richtlijnen worden doorgevoerd, moeten de testpersonen met de nieuwe richtlijnen de categorie meenemen, doordat in het script voor het omzetten van de JSON-tokens naar bruikbare variabelen voor Touchtribe, wordt er gefilterd op categorie. Doordat Figma automatisch ook nog variabelen apart toevoegt die al in typography worden meegegeven als één geheel, moet deze er handmatig uitgehaald worden. Het gaat bijvoorbeeld om de lineHeights, fontSize, letterSpacing, fontWeights, textCase en textDecoration. Gelukkig kan dit gemakkelijk in Figma worden verwijderd. Als de testpersonen alle categorieën had toegevoegd, kwamen we bij Figma Tokens uit. Daar werd de styleguide toegevoegd en omgezet in JSON-tokens.

### **Resultaten**

De resultaten van de usability testen met de designers is te lezen in Hoofdstuk 6.2 Usability test designers. Een designer maakt het ontwerp en de developer programmeert het. Mijn opdracht zit hier tussenin. Ik maak de voorwaarden om een designsysteem te bouwen. Het prototype wat hieruit

is gekomen is het script de Token-Transformer, zie Hoofdstuk 5.4 Figma Tokens; de Token-Transformer.

## Hoofdstuk 6 Onderzoeksresultaten analyse

### 6.1 Designsysteem

Een goed designsysteem bevat volgens Hoofdstuk 3.4.1 Richtlijnen verschillende richtlijnen. De richtlijnen zijn als volgt toegepast in het designsysteem van Touchtribe:

#### **Content, informatie en doelen**

Er wordt gekeken naar welke informatie relevant is voor in het opgeleverde designsysteem. De kleuren worden opgedeeld in bijvoorbeeld basiskleuren, neutrale kleuren en feedback kleuren om het overzichtelijk te houden voor de eindgebruikers.

#### **Elementen en componenten**

Het designsysteem bevat een kleurenpalet en typografie. Iconen worden door designers zelf gemaakt en later toegevoegd in het designsysteem onder een ander kopje. Momenteel is het probleem niet de iconografie, maar de onvolledige styleguide die aan de developers wordt

opgeleverd. In het nieuwe designsysteem moeten alle elementen en componenten gelabeld worden. Er moet gebruik gemaakt worden van de Screaming Snake Case bij het benoemen van de elementen en componenten. Daarnaast moet de categorie benoemd worden in de naam, zoals colors/PRIMARY\_COLOR. Ook moeten kleuren benoemd worden volgens de ontwerpprincipes, meer te lezen in Hoofdstuk 3.4.3 Kleuren. In *Bijlage 5 Prototype* is te zien dat het prototype is opgebouwd in frames en componenten. Er wordt van de designers verwacht na het doorvoeren van het nieuwe designsysteem dat zij mobile first te werk gaan. Uiteindelijk worden alle elementen in de plug-in Figma Tokens gezet, waarbij het omgezet kan worden in design tokens. Hierna moet het door het script speciaal voor deze tokens gehaald worden, waarnaar het naar wens van Touchtribe Javascript variabelen worden die zij kunnen gebruiken tijdens het werk.

### **Ontwerpers als designsysteem gebruikers**

Er moet voor ontwerpers gedocumenteerd staan hoe en wanneer elk onderdeel gebruik moet worden. Het moet een beschrijving, gebruikersopmerkingen, verwante elementen bevatten. UX of visuele opmerkingen en opmerkingen voor ontwikkelaars zijn optioneel.

In *Bijlage 5 Prototype* is te zien dat elke categorie groot is uitgedrukt, zodat snel te zien is waar welk onderdeel staat. Hier kan nog een korte uitleg geplaatst worden over de kleur, typografie of gewenste categorie. Omdat dit een prototype was zonder echte organisatie, is deze voor nu weggelaten maar kan dit zeker worden toegevoegd. Ook zijn er namen gegeven zoals base color, neutral color en feedback color om de verschillende kleuren van elkaar te scheiden en het gemakkelijk te maken voor designers en developers om informatie terug te vinden. Doordat alles gelabeld is kan er makkelijk bij het uiteindelijke design gekeken worden welke kleur bij welk object hoort en hoeft er geen omschrijving te komen wanneer en waar dit element gebruik wordt. Dit is alleen van toepassing bij het opstellen van moleculen, maar dat was niet waar het gegeven probleem vanuit Touchtribe lag. Voor de verwante elementen is de oplossing in de styleguide om gebruik te maken van de ontwerpprincipes voor kleuren. Dit brengt logica voor de designer en developer, omdat je weet hoeveel tinten de kleur van de basiskleur afzit. Aanvullende opmerkingen over het ontwerp kunnen designers in een opmerking in het Figma bestand zetten.

### **Developers als designsysteem gebruikers**

Om de developers alle code te leveren uit de styleguide, wordt het in de plug-in Figma Tokens gezet. Deze transformeert het tot JSON-tokens die uiteindelijk door een zelfgemaakt script om worden gezet in Javascript variabelen naar wens van Touchtribe. De nieuwe richtlijnen van het designsysteem houden ook in dat designers zich wel aan de sprints moeten houden. Als een sprint niet up-to-date wordt gehouden wordt het voor een designer erg moeilijk om op tijd stukken code af te krijgen.

## **6.2 Usability test designers**

De casus waar in dit hoofdstuk over wordt gesproken is te vinden in *Bijlage 6 Casus designers*. Voordat de casus van de designers begon heb ik duidelijk aan de testpersonen uitgelegd dat niks fout kan gaan en ze zich vooral geen zorgen hoeven te maken of ze niet professioneel overkomen. De test ging voornamelijk over begrijpen en feedback ontvangen vanuit de testpersonen. De junior designers hadden wat moeite met überhaupt een styleguide opstellen. Bij de senior designers gingen dit gemakkelijker. Gedurende de test hadden sommige testpersonen vragen over waarom ze bepaalde dingen moesten doen. Ondanks de uitleg over de onderwerpen die ik van tevoren met ze was doorgelopen, begrepen enkele testpersonen bijvoorbeeld niet waarom ze hoofdletters en underscores moesten gebruiken. Ik vertelde ze dat dit uiteindelijk nodig is voor de tokens en liet op het einde ook de JSON-tokens vanuit Figma Tokens zien. Daaruit zag je dat door de Screaming Snake Case je heel makkelijk de onderwerpen kon onderscheiden van elkaar dan dat ze allemaal

lowercase<sup>4</sup> waren. Het toevoegen van een categorie zoals 'colors/', was ook voor een testpersoon niet geheel duidelijk. Zij dacht dat die onderwerpen er ook wel zonder de 'colors/' bijstonden, omdat dat in Figma zelf gedaan werd. Ik liet haar zien bij de laatste stap dat in de JSON-tokens de categorie er wel degelijk bijgezet moest worden. Hierna begreep ze waarom dit nodig was voor het development team. Het volgende liet ik haar zien:

```
{
  "colors": {
    "SECONDARY_TAP": {
      "value": "#c95501",
      "type": "color"
    },
    {
      "SECONDARY_TAP": {
        "value": "#c95501",
        "type": "color"
      },

```

Een enkele testpersoon merkte op dat ze het minder prettig vindt om een underscore-teken te gebruiken bij het onderscheiden van namen. Ze zei dat dit hun manier van mappen onderscheiden PRIMARY/DEFAULT is. Uiteindelijk is zij de enige geweest die dit opmerkte. Daarnaast voelde het voor haar gevoel dubbel om tweemaal 'colors' neer te moeten zetten, dus in het Figma bestand zelf en om het als categorie 'colors/' toe te voegen. Sommige testpersonen vonden de Screaming Snake Case ook wat schreeuwerig overkomen en technischer overkomen.

### Eindvragen

Na het maken van de casus heb ik een aantal vragen aan de testpersonen gesteld. De vragen zijn gebaseerd op de casus, maar ook over hun gebruikelijke werkwijze. Over deze enkele punten van deze werkwijze struikelde Touchtribe. Ook worden er vragen gesteld zodat er gekeken kan worden of de wensen van de designers ook vervuld zijn. De vragen waren als volgt:

- Hoe bevallen de nieuwe richtlijnen je?
- Heb jij het gevoel dat je niet meer vrij kan zijn met het designen? En waarom?
- Heb je nog op- of aanmerkingen?
- Ben jij secuur in het gebruiken van je SCRUM? Dus als je een gedeelte af hebt, plaatst je dat dan ook op het SCRUM bord zodat development daar al mee kan beginnen?
- Werk jij met het designen mobile first? En waarom doe jij het op deze manier?
- Zijn er nog dingen waarvan jij denkt dat dat de samenwerking tussen design en development kan verbeteren?

De nieuwe richtlijnen bevielen over het algemeen bij alle testpersonen goed. De ene had al met dit soort richtlijnen bij andere bedrijven gewerkt en voor de andere was dit vrij nieuw.

Het is voor de testpersonen een kleine moeite om te doen. Zo vond alleen één testpersoon het onhandig omdat ze nu geen mapjes voor de bijvoorbeeld de kleuren default en select enzovoort kon maken. Testpersonen vonden het wel erg fijn dat de richtlijnen door middel van een casus werden doorgenomen en daarbij meteen vragen konden stellen. Doordat deze vragen gelijk beantwoord werden begrepen ze ook waarom ze een handeling moesten doen en onthielden dit ook beter.

Geen enkele designer voelde zich niet meer vrij met designen door de nieuwe richtlijnen. Zo kunnen ze nog steeds zelf tinten beslissen en hebben ze nog steeds volledige vrijheid met teksttypes. Het enige wat wordt toegevoegd zijn voor hun gevoel logische namen. Als dit het makkelijkste is voor development voelen ze zich belemmert. Alleen denkt één testpersoon dat ze minder snel kan werken doordat ze geen bijvoorbeeld geen mapje met 'PRIMARY/DEFAULT' meer kan aanmaken. Wel vond een andere testpersoon het fijn als deze richtlijnen pas na de wireframes ingaan, maar dit

---

<sup>4</sup> Geen gebruik maken van hoofdletters.



kan worden besproken per sprints. Bijvoorbeeld dat vanaf sprint 3 alle naamgeving moet kloppen zodat development daarmee kan beginnen.

De op- of aanmerkingen vielen mee. De Screaming Snake Case komt onvriendelijker over dan de lowercase die zij nu gebruiken, maar is geen belemmering. Voor één testpersoon gaf het een dubbel gevoel dat zij nu een keer colors moest neerzetten in het pallet en nogmaals als categorie bij de benaming. Ze vond dit minder optimaal, maar kan er wel mee werken. Verder vonden de testpersonen het erg fijn dat je door middel van mijn prototype uitleg kreeg en begeleid werd door de nieuwe richtlijnen. Het prototype en de voorbeelden maakte het visueel. Ook hebben een paar testpersonen door de test uitleg gekregen over Figma Tokens, waar zij voorheen nog nooit mee hadden gewerkt. Zij begrijpen nu hoe deze plug-in werkt en het nut ervan.

Nu kwamen de vragen over de werkwijze van de testpersonen. Op de vraag of de testpersonen secuur met SCRUM werkte kwam al gauw naar voren dat dit heel verschillend per project is. Het wordt niet bij elk project gebruikt, doordat de designers niet alleen met Touchtribe samenwerken. Als de vraag eruit development is, geven de designers aan met SCRUM te werken. Toch wordt het bord weleens vergeten en uiteindelijk niet helemaal secuur meegewerkt.

De volgende vraag ging over het mobile-first werken vanuit de designers. Dit ligt ook weer aan het project. Designers vragen aan de klant op welk platform de meeste gebruikers zitten. Momenteel is dit vaker op mobiel, maar toch wordt er niet altijd mobile first gewerkt. Dit ligt voornamelijk aan de opdrachtgever. Een enkele vindt mobile first werken makkelijker, omdat zodra je vanuit desktop werkt en dat moet omzetten naar mobile first het design meestal moeilijker is. Uiteindelijk komt het naar voren dat het bij junior designers wel wordt meegegeven dat ze mobile first moeten werken, omdat mobiel meer gebruikt wordt dan desktop en andere schermen. Senior designers hoeven minder uitleg meer te krijgen over designen, dus krijgen ze dit minder mee.

De laatste vraag ging over of de testpersonen nog punten wist waarmee zij denken dat de samenwerking tussen design en development beter gingen verlopen. Het kwam erop neer dat er duidelijke afspraken bij het begin van het project moeten worden afgesproken. Design en development moet uiteindelijk met elkaar samenwerken. Er moeten updates gegeven worden over tinten of naamgeving als Touchtribe in de toekomst toch besluit andere richtlijnen aan te nemen. Daarnaast moet er gewerkt gaan worden met sprints en moet er per sprint updates en afspraken gemaakt worden met development. Er moeten in deze sprints ook tijd vrijgemaakt worden om meetings in te plannen over het design. In deze meetings kan development de designs bekijken en hiermee sparren met het designteam, zodat de designs verbeterd kunnen worden. Deze sprints moeten wel in het project ingecalculeerd worden, omdat beide teams het erg druk hebben en je toch naar het beste resultaat wilt werken.

### 6.3 Adviesrapport

Uit de analyse is er een conclusie gekomen dat het gebruik van Figma Tokens het beste aan te raden valt. Dit advies is gedocumenteerd in een adviesrapport dat in *Bijlage 9 Adviesrapport* te vinden is. Het advies dat Touchtribe geleverd krijgt is als volgt in het kort:

Alternatief 1 is de best mogelijke oplossing voor het probleem. Dit alternatief houdt het toepassen van de plug-in Figma Tokens in. Figma Tokens is door enkele designers al gebruikt bij andere projecten die niet in samenwerking waren met Touchtribe. Sommige designers zijn er dus al mee bekend. Door de plug-in Figma Tokens te gebruiken kunnen zoveel mogelijk wensen en behoeften vanuit het bedrijf worden gewaarborgd. Dit lost een groot deel van het probleem op, doordat er te maken is met verschillende teams; designers en developers. Het gebruiken van deze plug-in zou 2

weken nodig hebben om aan de designers uit te leggen en er mee kennis te leren maken. Dit komt door de drukke schema's van de designers en het leren werken met de plug-in.

### Benodigheden

De benodigheden die nodig zijn om alternatief 1 uit te voeren is de ontwerpomgeving Figma, communicatie tussen beide teams en de plug-in Figma Tokens in Figma.

### Voordelen

- Plug-in wordt vertegenwoordigd vanuit Figma; waterdicht systeem
- Kan naar GitHub gestuurd worden
- Eenvoudiger; kan je eigen benamingen meegeven

### Nadelen

1. Als developers stoppen, houdt de plug-in op

Met deze plug-in is het script de Token-Transformer ontwikkelt. Met de Token-Transformer is getest met developers van Touchtribe, zie Hoofdstuk 5.2 Nul- en éénmeting. Bij de nulmeting kwam het gemiddelde cijfer 6 naar voren. Bij de éénmeting en dus het gebruik van de Token-Transformer kwam er een gemiddeld cijfer 8,2 uit. De developer experience is dus daadwerkelijk met ruim 2 punten op een schaal van 1 tot 10 omhooggegaan.

## Hoofdstuk 7 Conclusie

### 7.1 Hoofdvraag

De hoofdvraag luidt als volgt: ***“Wat zijn de wensen en behoeften van de doelgroep de ontwerpers en developers van Touchtribe m.b.t. het ontwikkelen van een nieuw designsysteem?”***

De wensen en behoeften van de doelgroep de ontwerpers en developers zijn in Hoofdstuk 4 Experimenteel onderzoek naar voren gekomen. De developers wensen gedefinieerde basics te zien, gekoppelde variabelen, componenten die voorzien zijn van naamgeving en een automatische geëxporteerde tokenflow naar de development omgeving. De designers wensen de vrijheid van ontwerpen bij het designsysteem te behouden. Ook zij willen een automatische tokenflow en wensen gerichte afspraken te hebben met de developers.

### 7.2 Deelvragen

De eerste deelvraag luidt als volgt: ***“Waar loopt de doelgroep tegenaan tijdens het werk?”***

In Hoofdstuk 4.2 Huidige manier van werken komt naar voor dat developers momenteel ertegenaan lopen dat het opgeleverde design niet consistent is. Designers maken geen gebruik van variabelen of geven geen naamgeving. Hierdoor moeten developers zelf uitzoeken wat bijvoorbeeld de typografie is. Ook wordt er dan soms verkeerde naamgeving meegegeven. Daarnaast missen regelmatig states van een button of formulervelden. Er staat niet genoteerd of er is niet ontworpen wat er moet gebeuren met een button wanneer deze een bepaalde state heeft; bijvoorbeeld wat er gebeurt als er op de button wordt geklikt. Designers en developers hebben beide andere werkwijzen en hierdoor komt het voor dat developers iets niet begrijpen, maar de designer al op een ander project zit. Voordat een developer antwoord heeft gekregen, is er al wat tijd overheen gegaan. De designers lopen er tegen aan dat er geen designsysteem aanwezig is, die duidelijk aangeeft hoe ze bepaalde componenten en variabelen moeten opleveren.

De tweede deelvraag luidt als volgt: ***“Wat is de huidige manier van werken bij de doelgroep?”***

In Hoofdstuk 4.2.1 Developers worden de stappen uitgebreid besproken die de developers doorlopen bij werken aan een project. Developers zetten een project op met een library. Daarna maken ze een koppeling met CMS. Na deze koppeling kan er aan het werk worden gegaan met de richtlijnen, oftewel het niet bestaande designsysteem. De eerste basis aan variabelen worden geïmplementeerd, zoals de kleuren typografie, icon en shadows. Hier is nooit alles van tevoren gedefinieerd wat irritaties geeft aan de developers. Zij moeten terug naar de designers of zelf de dingen aanvullen. Dit kost tijd. Als de hele styleguide aangevuld is kunnen zij beginnen met de ‘content blocks’.

Bij de designers komt naar voren dat zij niet echt weten waarom zij mobile first moeten werken. Hierdoor wordt dit ook vaak niet gedaan. Daarnaast doen sommige dit niet uit laksheid. Mobile first werk je niet alleen maar omdat mobiel momenteel meer gebruikt wordt, maar ook omdat dit voor development erg belangrijk is. De designers hebben geen specifieke werkwijze, waardoor de ontwerpen vaak niet volledig worden aangeleverd.

De derde deelvraag luidt als volgt: ***“Welke delen van de wensen en behoeften van de doelgroep worden al toegepast?”***

Uit Hoofdstuk 4.4 Al aanwezige wensen en behoeften komt naar voren dat developers en designers aangeven dat ze met de tool Figma werken. Dit is voor hun beide erg prettig, omdat hier al langere tijd mee werken. De developers geven aan dat er geprobeerd wordt de developers vroegtijdig bij het ontwerp te betrekken, maar dit door communicatie niet altijd soepel loopt. De designers geven aan dat er naar de inhoud van elk project wordt gekeken, zoals het zo goed mogelijk definiëren van kleuren en typografieën.

De vierde deelvraag luidt als volgt: ***“Welke informatie wenst de doelgroep te zien bij het designsysteem?”***

Uit Hoofdstuk 4.5 Wensen en behoeften komt naar voren dat developers gedefinieerde basics willen. Op deze manier wordt er voorkomen dat developers verkeerde variabelen doorvoeren in de code of zelf achter een onvolledig ontwerp aan moeten gaan. Ook willen zij gekoppelde variabelen zien die voorzien zijn van de correcte namen en juiste gegevens. Alle componenten moeten zijn voorzien van naamgeving. Verder is een erg belangrijke wens van de developers dat er een geautomatiseerde tokenflow is die exporteert wordt naar de development omgeving.

De designers wensen vrijheid te hebben. Zij willen bij het designsysteem niet belemmerd worden met ontwerpen. Ook zij zouden een geautomatiseerde tokenflow erg fijn vinden, omdat dit volgens de designers voor minder wrijving zou zorgen. Tot slot willen zij gerichte afspraken vanuit de developers, omdat de developers het er vaak onderling niet over eens zijn hoe bepaalde variabelen gedefinieerd moeten worden.

## Hoofdstuk 8 A-competenties

### 8.1 Onderzoek

In Hoofdstuk 1 Inleiding wordt de probleemanalyse opgesteld met hoofd- en deelvragen die SMART zijn gedefinieerd in Hoofdstuk 2 Hoofd- en deelvragen. Verder zijn hier ook de deelvragen te vinden. De hoofdvraag luidt als volgt: ***“Wat zijn de wensen en behoeften van de doelgroep de ontwerpers en developers van Touchtribe m.b.t. het ontwikkelen van een nieuw designsysteem?”*** Deze hoofdvraag en verdere deelvragen worden in het verslag beantwoord door het onderzoek in Hoofdstuk 3 Geserreerd literatuuronderzoek en Hoofdstuk 4 Praktisch onderzoek. Naast de hoofd- en deelvragen wordt er ook een probleemformulering in Hoofdstuk 1.1.3 Probleemformulering opgesteld en in Hoofdstuk 1.1.4 Doelstelling een doelstelling.

Er is gebruik gemaakt van een stakeholderanalyse in Hoofdstuk 5 Methoden en technieken. Op deze manier wordt er een beter beeld van de stakeholders weergegeven. Naast deze stakeholderanalyse is er een nul- en éénmeting uitgevoerd, om de verbetering van de wensen en behoeften van de doelgroep in kaart te brengen. Dit is terug te lezen in Hoofdstuk 5.2 Nul- en éénmeting.

Er zijn richtlijnen opgesteld die voort zijn gekomen uit de developers en zijn getest met de designers. Hier is meer over te lezen in Hoofdstuk 6.1 Designsysteem. Naast deze richtlijnen heb ik ook op technisch gebied een vertaling gemaakt naar een designsysteem door een script de Token-Transformer te maken. Deze voldoet aan de wensen en behoeften van de doelgroep. Over deze Token-Transformer is meer te lezen in Hoofdstuk 5.4 Figma-Tokens; De Token-Transformer.

### 8.2 Leren leren

In het begin moest ik nog erg wennen aan de professionele werksfeer op de werkvloer. Uiteindelijk viel dit allemaal wel mee, omdat iedereen jonger is en heel begaand is met elkaar. Hierna moest er begonnen worden aan het omzetten van de Figma Tokens. Ik wist niet zo goed waar ik moest beginnen. Ik had al contact met Vince van design gehad, hoe zij dit in andere projecten hebben aangepakt. Hierdoor werd ik doorgestuurd naar Jeffrey. Hij heeft mij een opzetje gegeven waar ik verder op ben gaan bouwen. Ik wilde Maarten benaderen voor een error waar ik niet uitkwam, maar hij was die dag niet op kantoor door meetings met klanten. Hierdoor heb ik het aan een developer op kantoor gevraagd. Fahrat heeft me goed begeleid en ik kon hem bij verdere vragen meteen aan zijn jas trekken. Verder heeft Maarten uitgelegd hoe een CMS moet worden ingericht. Hier is meer over te lezen in Hoofdstuk 8.3.1 Vertalen ontwerp. Mijn reflectie is verder te lezen in het Hoofdstuk 10 Reflectie.

Ik heb hiermee een STARR-reflectie gemaakt, om de competentie leren leren aan te kunnen tonen. Een STARR-reflectie is een methode om te reflecteren op je eigen handelen (Benders, Scribbr, 2021). De STARR-reflectie is als volgt:

### Situatie

Ik loop stage bij Touchtribe waarin zij te werk willen gaan met een automatische tokenflow bij het ontwerpen in de tool Figma.

### Taak

Ik moet een script schrijven om de tokens uit Figma Tokens om te zetten naar bruikbare variabelen.

### Actie

Ik heb eerst de documentatie van de plug-in Figma Tokens doorgelezen. Vervolgens heb ik contact gezocht met Jeffrey, een collega van Vince, die voor een ander project een transformer heeft gebouwd. Hij had me een opzetje gestuurd, die ik verder zelf heb aangepast naar de wensen en behoeften van de doelgroep.

### Resultaat

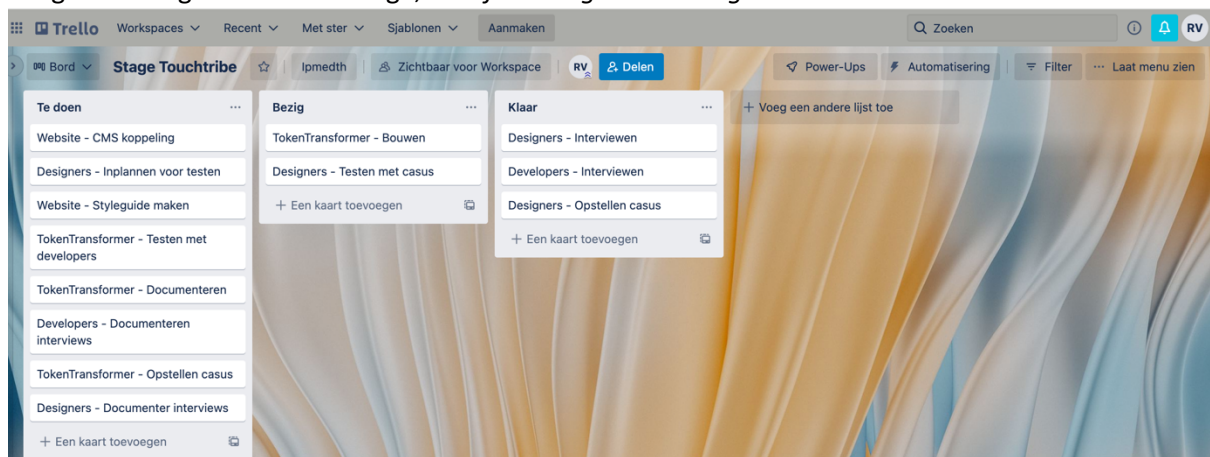
Een supergaaf bruikbaar script, die een gemiddeld cijfer van een 8,2 heeft gekregen bij het testen met de developers.

### Reflectie

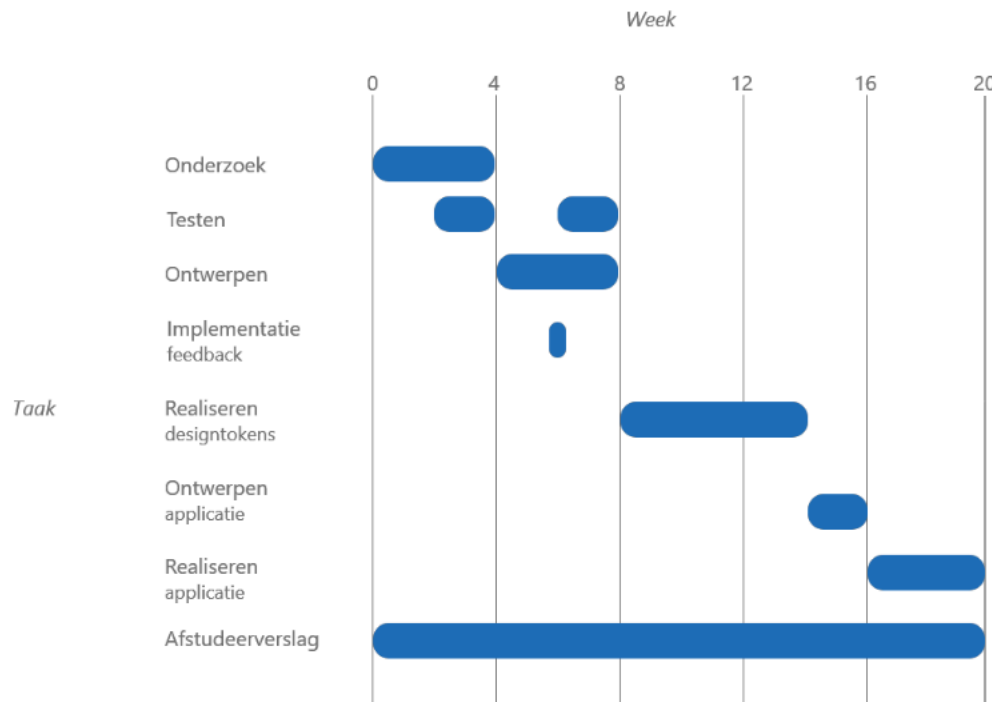
Ik vond het in het begin nogal ingewikkeld waar ik moest beginnen. Dit kwam doordat ik niet goed voor me kon zien wat er gemaakt moest worden. Uiteindelijk is het goed gekomen door de begeleiding van Jeffrey. De volgende keer zal ik meer moeten communiceren met de doelgroep, om goed te kunnen visualiseren wat zij willen hebben. Op deze manier kan ik sneller aan de gang.

## 8.3 Professioneel werken

Om de twee weken is er met Maarten een periodieke meeting gehouden, om door te spreken welke dingen afgemaakt zijn, aan welke dingen begonnen worden en tegen welke dingen ik was aangelopen. Daarnaast is het maken van de website volgens het nieuwe designsysteem in Jira genoteerd, zie *Afbeelding 13 Trello*. De planning volgens de Gantt grafiek uit het afstudeerplan is aangehouden gedurende de stage, zie *Afbeelding 14 Planning*.



Afbeelding 13 Trello



Afbeelding 14 Planning

Doordat een andere student van de studie Informatica van de Hogeschool Leiden ook bij Touchtribe afstudeert, heeft Maarten aangegeven dat het goed is om ook elkaars verslag feedback te geven. Vooral in het begin, toen het afstudeerplan ingeleverd moest worden was dit erg handig. Verder hebben we ook brainstormsessies gehouden over bijvoorbeeld een goede hoofd- en deelvragen.

#### 8.4 Innovatie

Om de twee weken heb ik een periodieke meeting met Maarten over mijn proces gedurende de stage en een moment om vragen te stellen waar ik gelijk antwoord op kreeg. Dit werd gehouden op een woensdag via Google Meeting. Daarnaast vond ik het erg lastig om iets op te stellen als de opdrachtgever ook niet helemaal weet wat hij wilde. Hierdoor vond ik het begin van de stage lastig, doordat ik niet helemaal voor me kon halen wat voor een soort designsysteem er van mij verwacht werd. Onderzoek ernaar doen was dan ook pittiger, omdat je niet gericht gaat zoeken. Uiteindelijk heb ik dit bij Maarten, Linda en Vince neergelegd en later ook met Jaap besproken. Ik ben uiteindelijk met Linda gaan zitten en die had voorgelegd dat zij bijvoorbeeld graag automatisch tokens wilde ontvangen die omgezet werden naar bepaalde javascript variabelen waar Touchtribe mee aan het werk kon. Ook vertelde ze de knelpunten waartegen aangelopen werd tijdens het werk met de designers. Hier heb ik een goede richting door gekregen, waardoor ik redelijk wist wat er van mij verwacht werd.

Touchtribe doet nog wel het een en ander met het team. Dit vond ik het begin nogal spannend, omdat ik natuurlijk jong en nieuw was. Al gauw werd ik bij het teamuitje erg betrokken door mijn collega's en heb ik erg leuke gesprekken gehad. Hierdoor ben ik meer uit mijn schulp getrokken.

Ik ben met een creatieve oplossing gekomen bij het namaken van een deel van Cerqlar website met CMS, omdat Touchtribe zelf mappers gebruikt en ik de data op mijn eigen manier eruit heb gehaald.

Meer hierover te lezen in Hoofdstuk 9.3 Gebruikersinteractie – Realiseren. Ook ben ik bij het bouwen van de Token-Transformer er creatief geweest om de wens van Touchtribe te realiseren. Over de Token-Transformer is meer te lezen in Hoofdstuk 5.4 Figma-Tokens; De Token-Transformer.

Het advies leunt op de analyse, alleen wordt dit getest in iets andere setting. Het wordt getest door middel van de nul- en éénmeting uit Hoofdstuk 5.2 Nul- en éénmeting.

## 9.1 Gebruikersinteractie – Analyseren

### Bevindingen

In Hoofdstuk 3 Geserreerd literatuuronderzoek zijn de belangrijke kernpunten van een designsysteem naar voren gekomen. Hieruit kan een beeld gevormd worden wat een designsysteem inhoudt.

In Hoofdstuk 4 Praktisch onderzoek is de huidige manier van werken van de developers en ontwerpers onderzocht. Er is gekeken waar de developers en ontwerpers tegen aanlopen bij het gebruik van het huidige designsysteem. Daarnaast werd er gekeken naar wensen en behoeften van de doelgroep die al in het huidige designsysteem wordt toegepast. Tot slot wordt er gekeken naar welke informatie de doelgroep wenst te zien bij het nieuwe designsysteem.

Verder is uit het experimenteel onderzoek is naar voren gekomen dat Touchtribe en het designteam over al meerdere informatie beschikte, dan dat ze zelf wiste. Na het onderzoek tussen Storybook en Figma, kwam naar voren dat Figma twee verschillende tools bevatten die Figma elementen kon omzetten naar JSON-tokens en deze kon versturen naar GitHub. Toen ik dit voorlegde bij Vince van het designteam, kwam er naar voren dat zij dit bij andere bedrijven en samenwerkingen al gebruikte. Touchtribe wist hier niks van. Zo komt sterk naar voren dat de communicatie tussen deze twee takken niet al te best is. Na me verdiept te hebben in Figma Tokens, kwam ik bij het volgende. Ik ging wat spelen met een zelfgemaakt prototype in Figma, om te kijken of de JSON-tokens goed waren. Dit had ik bij Linda voorgelegd, maar zij wees mij dat het development-team het op een andere manier wilde ontvangen.

### Analyse

In Hoofdstuk 4.3.4 Customer Journey is er gebruik gemaakt van een customer journey om de manier van werken van de doelgroep in kaart te brengen. Hier zijn ook de problemen waar Touchtribe tegen aanloopt naar voren gekomen. In Hoofdstuk 4.6 SWOT-Analyse is er een SWOT-analyse opgesteld waarbij er geanalyseerd is welke punten belangrijk zijn om terug te laten komen in de manier van werken.

In Hoofdstuk 5.1 Stakeholderanalyse wordt er gekeken naar 4 stakeholders die betrokken zijn bij deze stageopdracht; Touchtribe, klanten, developers en ontwerpers. In Hoofdstuk 5.2 Nul- en éénmeting is er een nul- en éénmeting uitgevoerd met het script de Token-Transformer en getest met de developers. Hoofdstuk 5.3 Opzetten designsysteem zijn de basispunten die een goed designsysteem vormen onder elkaar gezet en samengevoegd met de nieuwe richtlijnen. Deze basispunten zijn uit Hoofdstuk 3.4 Inhoud designsysteem gehaald. In Hoofdstuk 5.5 Opzet usability test designers is de opzet van de usability test opgesomd met daarbij de bijbehorende casus uitgelegd.

### Conclusie

In Hoofdstuk 6.1 Designsysteem is het nieuwe designsysteem van Touchtribe te lezen. Hier worden de afspraken binnen Touchtribe uitgeschreven, die doorgespeeld worden naar de ontwerpers. Daarnaast worden ook de richtlijnen onder elkaar gezet en het opstellen van een styleguide. Het

designsysteem kan verbonden worden met GitHub door middel van Figma Tokens. Hierna moeten de tokens door een script genaamd 'design.js' gehaald worden om de tokens om te zetten naar bruikbare variabelen voor Touchtribe. Dit nieuwe designsysteem is ontworpen door de informatie uit Hoofdstuk 3 Geserreerd literatuuronderzoek en Hoofdstuk 4 Praktisch onderzoek.

#### 9.1.1 Gebruikersonderzoek

De probleemanalyse wordt in Hoofdstuk 1.1 Probleemanalyse besproken waar een probleemstelling is opgesteld met hoofd- en deelvragen die SMART zijn geformuleerd. In Hoofdstuk 3 is er gebruik gemaakt van een Geserreerd literatuuronderzoek en in Hoofdstuk 4 van een Praktisch onderzoek.

#### 9.1.2 Methoden en technieken

In Hoofdstuk 5 Methoden en technieken is een subkop 5.1 Stakeholderanalyse, 5.2 Nul- en éénmeting, 5.3 Opzetten designsysteem, 5.4 Figma Tokens; de Token-Transformer en 5.5 Opzet usability test designers te vinden.

#### 9.1.3 Prototyping & design

In Hoofdstuk 5.3 Opzetten designsysteem is er een prototype gemaakt, zie *Bijlage 5 Prototype*. Dit is een prototype met de nieuwe richtlijnen uit Hoofdstuk 5.3 Opzetten designsysteem. Daarnaast is er gebruik gemaakt van een customer journey in Hoofdstuk 4.3.4 Customer Journey.

#### 9.1.4 Usability

Er is een usability test met 5 testpersonen van het designteam uitgevoerd in Hoofdstuk 6.1 Usability test designers.

### 9.2 Gebruikersinteractie – Adviseren

#### 9.2.1 Onderzoek

Het onderzoek wat tot het advies heeft geleid is gehaald uit Hoofdstuk 3 Geserreerd literatuuronderzoek en Hoofdstuk 4 Praktisch onderzoek.

#### 9.2.2 Rapportage

Het adviesrapport is in Hoofdstuk 6.3 Adviesrapport te vinden en volledig te lezen in *Bijlage 9 Adviesrapport*.

#### 9.2.3 Prototyping

Het prototype wat hierbij hoort is het script de Token-Transformer uit Hoofdstuk 5.4 Figma Tokens; de Token-Transformer.

### 9.3 Gebruikersinteractie – Realiseren

#### 9.3.1 Vertaling van ontwerp

##### **CerQlar**

Doordat Touchtribe alleen grote projecten had liggen, mocht ik voor mijn stageopdracht een website nabouwen en een styleguide maken volgens de nieuwe richtlijnen. De aangewezen website is CerQlar. CerQlar is een platform waar de handelsworkflow op een effectieve en efficiënte manier beheert kan worden. CerQlar maakt gebruik van een Figma bestand waar de styleguide in opgesteld is. Het vervelende is dat dit project al is afgerond en de styleguide gedurende het project is verbeterd voor de developers. Ik heb in het nieuwe prototype de nieuwe richtlijnen meegenomen.



Verder maakt CerQlar in deze website gebruik van CMS, ‘content-blocks’, font-families die niet eenvoudig te vinden zijn en tokens.

## Designsysteem

Het nieuwe designsysteem wordt toegepast in *Bijlage 8 Designsysteem CerQlar*. Er moest gehouden worden aan de volgende richtlijnen:

- Benaming tokens:  
De tokens moeten een benaming in de Screaming Snake Case krijgen. Dit houdt in dat alles in hoofdletter moet en bij woordscheiding een underscore-teken gebruikt wordt. Dit is te zien in *Bijlage 4 Designsysteem CerQlar*.
  - Categorie in benaming:  
Er moet gebruik gemaakt worden van categorie in de benaming van de variabelen. Een voorbeeld is: colors/PRIMARY\_COLOR. Er is gebruik gemaakt van ‘/colors’ en ‘/typography’.
  - Kleuren gerepresenteerd op basiskleur:  
De kleuren moeten gerepresenteerd worden op basiskleur volgens ontwerpprincipes. CerQlar maakt alleen geen gebruik van een reeks aan tinten, waardoor deze richtlijn niet toegepast hoefde te worden.
  - Figma Tokens:  
Er wordt gebruik gemaakt van de plug-in Figma Tokens om de tokens uiteindelijk in JSON-tokens te kunnen versturen naar de development. Uit dit designsysteem zijn bruikbare tokens gekomen, die door het script gehaald kunnen worden en hierdoor gebruikt kunnen worden in de code.
- 
- Mobile first werken:  
Bij het ontwerpen moeten er mobile first gewerkt worden. Er is geen nieuw ontwerp gemaakt, dus deze richtlijn was hier niet van toepassing.
  - Gebruiken maken van sprints:  
Er moet duidelijk aangegeven worden wanneer een design of een stuk daarvan af is, zodat het development team hier al mee aan de slag kan. Deze richtlijn is niet van toepassing geweest, omdat het design al af was en er met het nabouwen niet met designers samengewerkt hoefde te worden.

## Website nabouwen

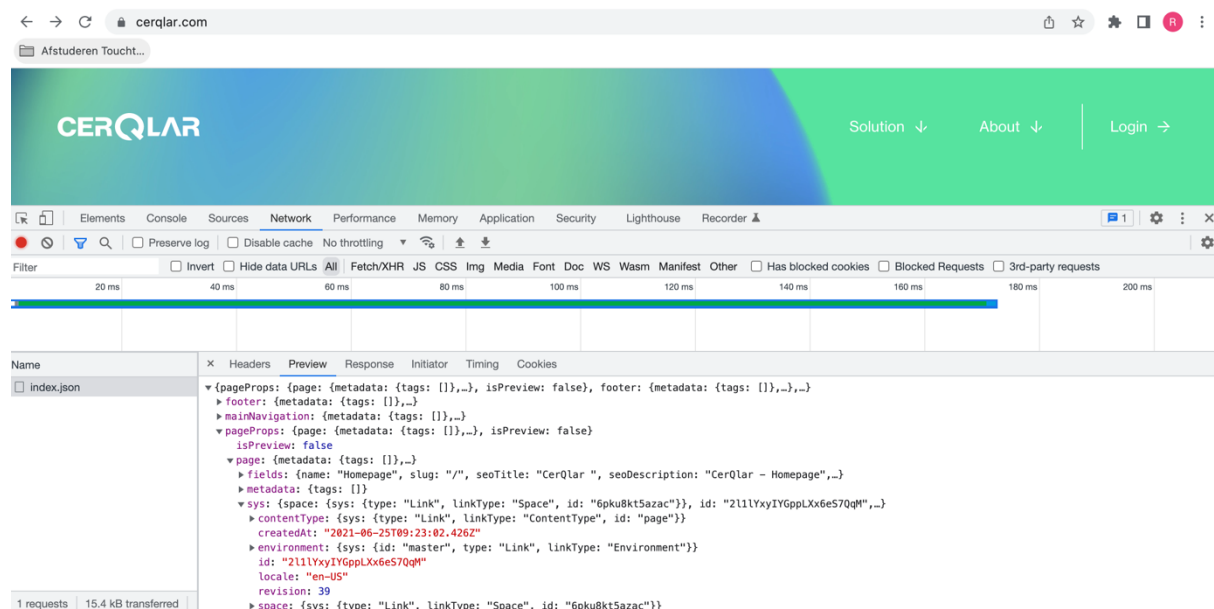
### Styleguide

In *Bijlage 8 Designsysteem CerQlar* is de nieuwe styleguide van CerQlar te vinden. Hier is ook de GitHub van de website te vinden.

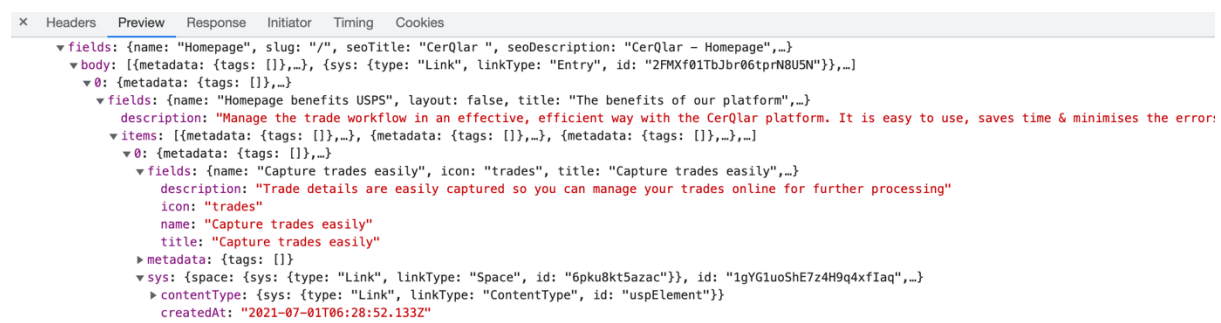
### CMS

Touchtribe maakt gebruik van CMS. CMS is een content management system waarin je zo eenvoudig mogelijk zelf de inhoud van de website kan beheren, zonder dat je programmeer kennis nodig hebt (Speek, 2012). Doordat CerQlar niet veel contact meer heeft over de CMS met Touchtribe, konden ze niet bij het al opgezette CMS van CerQlar. Om toch bij de werkwijzen van Touchtribe te blijven moest er zelf een ingericht worden. In een geplande meeting met Maarten heeft hij uitleg gegeven over het inrichten van een CMS. Touchtribe maakt gebruik van Contentful, een headless CMS. Hier

kan je een nieuw project aanmaken. De volgende stap is een content model aanmaken, maar hiervoor moest er gekeken worden in de inspector in de browser van de website. Onder het kopje Network is de index.json te vinden, zie *Afbeelding 15 Network index.json*. Om alle pagina's in de CMS te kunnen zetten moet je naar pageProps > sys > id, ofwel het content type, om te kijken naar het soort content model dat er gemaakt moet worden. In dit geval was het een Page. Om te kijken welke content er in het content model plaats moet vinden, moet er worden gekeken in de Fields. In deze Fields stond een Body en pageHeading. Er wordt even ingezoomd op de Body. Om de volledige content te vullen, moet de Fields worden uitgeklaapt en moet er gekeken worden naar de items. Voordat de items kunnen worden ingevuld, moet er nogmaals in de sys gekeken worden naar het type id die erbij hoort. In dit geval uspElement. Daarna moeten alle items ingevuld worden in de content. Uiteindelijk hoef je alleen de bijbehorende gegevens erbij in te vullen, bijvoorbeeld de naam "Capture trades easily", zie *Afbeelding 16 Inspector body*. Dit moet voor alle items en alle pagina types gedaan worden. In *Afbeelding 17 Content* is te zien dat er verschillende content types zijn aangemaakt.



*Afbeelding 15 Network index.json*



*Afbeelding 16 Inspector body*

The screenshot shows the Touchtribe Contentful Apps interface. The top navigation bar includes 'Touchtribe Contentful Apps', 'Space home', 'Content model', 'Content', 'Media', 'Apps', 'Settings', and a 'TRIAL' badge. The main section is titled 'Content' and features a search bar with 'Content type' set to 'Any' and a filter icon. Below the search bar, there are tabs for 'Scheduled Content', 'Releases', and 'Shared views'. A sidebar on the left shows 'All' and 'STATUS (4)' with options for 'Published' and 'Changed'. The main table lists content items with columns for Name, Content Type, Updated, Author, and Status. All items shown are 'PUBLISHED'.

Name	Content Type	Updated	Author	Status
CTA Terms & Conditions Page	callToActionElement	Yesterday, 1:34 PM	Me	PUBLISHED
Terms & Conditions	Page	Yesterday, 1:33 PM	Me	PUBLISHED
CTA Privacy Page	callToActionElement	Yesterday, 1:32 PM	Me	PUBLISHED
Privacy Policy	Page	Yesterday, 1:29 PM	Me	PUBLISHED
Footer Navigation List right	linkListElement	Yesterday, 1:04 PM	Me	PUBLISHED
Contact us CTA	callToActionElement	Yesterday, 1:03 PM	Me	PUBLISHED
News CTA	callToActionElement	Yesterday, 1:02 PM	Me	PUBLISHED
Footer Navigation List left	linkListElement	Yesterday, 1:01 PM	Me	PUBLISHED
Careers	callToActionElement	Yesterday, 1:01 PM	Me	PUBLISHED

Afbeelding 17 Content

Hierna moest de CMS worden toegepast in de code. Touchtribe deed dit door middel van mappers met Typescript. Ik gebruikte geen Typescript, dus moest ik zelf een handigheid vinden om deze data uit de CMS te kunnen halen, zie *Afbeelding 18 CMS data*.

The screenshot shows a code editor with the following JavaScript code:

```

cerqlar > src > contentBlocks > JS Homepage.js > [0] Header
24   'fields.slug': "homepage",
25   include: 3,
26
27   });
28   .then((response) => {
29     this.setState({ items: response.items })
30     console.log(response)
31   });
32 }
33
34 render() {
35   return (
36     <Container>
37       <Header>
38         <Title>{this.state.items.map((page) => {
39           return <div key={page.sys.id}>{page.fields.body[0].fields.title}</div>;
40         })}</Title>
41         <Description>{this.state.items.map((page) => {
42           return <div key={page.sys.id}>{page.fields.body[0].fields.description}</div>;
43         })}</Description>
44         <p>{this.state.items.map((page) => {
45           return <li key={page.sys.id}>{page.fields.body[0].fields.items[0].fields.title}</li>;
46         })}</p>
47         <p>{this.state.items.map((page) => {
48           return <li key={page.sys.id}>{page.fields.body[0].fields.items[0].fields.description}</li>;
49         })}</p>
50       </Header>
51     </Container>
52   );
53 }
54 }
55

```

Afbeelding 18 CMS data

## Tokens

De tokens die geïmporteerd worden zijn ook wel de thema's die Touchtribe opzet bij een project. It thema bevat de informatie die uit het designsysteem komt; de kleuren, de typografie, de box-shadow etc. Zie *Afbeelding 19 Thema's* voor de thema's van CerQlar. Deze tokens paste ik toe aan de styled components.

```
JS cards.js U JS colors.js U X
cerqlar > src > theme > basics > JS colors.js > ...
1 export const COLOR = {
2   AQUA_BLUE: "#3f8efc",
3   GRASS_GREEN: "#56e39f",
4   SANDY_BEIGE: "#f5f5f1",
5   NIGHT_BLACK: "#000000",
6   SNOW_WHITE: "#ffffff",
7   SUCCES_GREEN: "#22a922",
8   ERROR_RED: "#d52b1e",
9   WARNING_YELLOW: "#ffb612",
10  SUCCES_GREEN_LIGHT: "#daf2da",
11  ERROR_RED_LIGHT: "#f9dadd",
12  WARNING_YELLOW_LIGHT: "#fff4dc",
13  LIGHT_GREY: "#d5d5d6",
14  DARK_GREY: "#8d8d8d",
15 }
16
17
```

Afbeelding 19 Thema's

## Font

CerQlar maakt gebruik van font-families Neue Haas Grotesk en TT Interfaces. Deze fonts zijn niet eenvoudig te vinden, maar kunnen zodra ze gevonden zijn eenvoudig worden toegevoegd in React. Deze font moet vanuit je eigen desktop geïmporteerd worden in je React project. Vervolgens moet geïmporteerd worden in de index.js;

```
import './fonts/NeueHaas/NeueHaasDisplayRoman.ttf';
```

Hierna moet in de index.css een @font-face worden aangemaakt;

```
@font-face {
  font-family: "NeueHaas";
  src: local("NeueHaas"),
    url("./fonts/NeueHaas/NeueHaasDisplayRoman.ttf") format("truetype");
  font-weight: regular;
}
```

### 9.3.2 Usability testen

Ik heb realiseren anders aangetoond, door een nul- en éénmeting te doen met het script de Token-Transformer. Dit heb ik gedaan doordat mijn opdracht niet op de normale wijze aan te tonen is met usability testen met de ISO25010. Om deze competentie toch zo goed mogelijk aan te tonen is er toch met dit script getest en is de website CerQlar nagebouwd met de tokens en het script.

### 9.3.3 Stakeholders

Uit Hoofdstuk 5.1 Stakeholderanalyse komen de stakeholders van Touchtribe naar voren. Dit zijn de developers, designers, Touchtribe en de klanten. In dit geval komen alleen developers, Touchtribe en klanten aanbod.

## Hoofdstuk 10 Nawoord

Designsystemen en token zijn al tijden een onderwerp van gesprek binnen Touchtribe, maar het is ons tot nu toe niet gelukt om hier een goede werkwijze in te vinden met de designers.

De beperkte ervaring hiermee binnen het team van designers en developers zorgden er tot u toe dat hier geen stappen in gemaakt konden worden. Dankzij het onderzoek en de inzet van Robin hebben we nu een werkwijze geformuleerd die voor zowel designers als developers geschikt is. De voorbeeld implementatie voor CerQlar bewijst dat de technische realisatie hiervan ook al werkt.

Robin heeft met haar onderzoek en uitwerking gezorgd dat Touchtribe een grotere stap dichterbij bij onze doelen zijn gekomen. We zijn nu klaar om deze nieuwe werkwijze te implementeren.

Maarten Jongerius

## Hoofdstuk 11 Reflectie

Voor mijn reflectie is er gebruik gemaakt van het reflectiemodel Korthagen. Het model van Korthagen is een methode die je kunt gebruiken in je reflectieverslag om te reflecteren op je handelen (Benders, Reflecteren met het Model van Korthagen, 2019).

### **Situatie ervaren**

Op maandag 7 februari ben ik begonnen aan mijn afstudeerstage bij Touchtribe. Hier kreeg ik een opdracht om een designsysteem te ontwerpen om de samenwerking tussen ontwerpers en developers beter te laten verlopen. Zij nemen aan dat een designsysteem mogelijk zou helpen om de pijnpunten uit de weg te gaan. Developers zijn duur, lastig te vinden en nog moeilijker om te behouden.

### **Terugblikken**

In het begin vond ik het erg lastig om te kunnen visualiseren wat ik moest bouwen. Doordat Touchtribe zelf ook nog niet helemaal rond hadden wat zij wilde, maakte dit het voor mij extra lastig. Ik heb toen meerdere malen gesprekken gehad met mijn stagedocent Jaap Kanbier, om het hier over te hebben. Ik voelde me hier namelijk nogal onzeker over en begon stress te ervaren. Na deze gesprekken ben ik goed gaan zitten met mijn stagebegeleider Maarten en werden deze stress gevoelens weggenomen. Ik wist wat ik moest gaan bouwen en waar ik moest beginnen. Daarnaast moet ik ook goed op een rijtje hebben staan wat ik allemaal moest gaan doen, doordat ik nogal chaotisch ben. Dit zou mij ook meer rust geven. Ik benaderde Vincent van design en die stuurde mij door naar Jeffrey. Jeffrey legde uit hoe hij voor een ander bedrijf een transformer had gebouwd. Met deze informatie en de documentatie van internet ben ik aan de slag gegaan.

### **Belangrijke aspecten begrijpen**

Door hulp te zoeken bij mijn stagedocent en stagebegeleider werd ik mij er bewust van dat je bij hulp gewoon aan de bel moet trekken. De mensen om mij heen weten niet dat ik ergens tegen aanloop of stress ervaar als ik hier niks over uitspreek. Door het uit te spreken ben ik goed geholpen en is er uiteindelijk een goed resultaat uitgekomen.

### **Alternatief ontwikkelen**

Eventuele handelsalternatieven waren:

- Eerst erg veel zelfonderzoek doen  
Door zelf heel het internet af te pluizen kom ik misschien bij de antwoorden.
- Meteen hulpvragen

Om tijd te besparen kan ik ook meteen hulpvragen. Hiermee kan je een klein zetje op de goede weg krijgen.

### **Uitproberen**

Toen ik niet uit een bug kwam en zelf al veel onderzoek had gedaan op internet, wilde ik hulp bij Maarten gaan vragen. Maarten was niet aanwezig, dus moest ik hulp gaan vragen bij andere collega's. Dit vond ik eerst wat spannend, maar heb uiteindelijk hulp gevraagd bij Fahrat die mij geholpen had met de bug.

### **Nieuwe situatie ervaren**

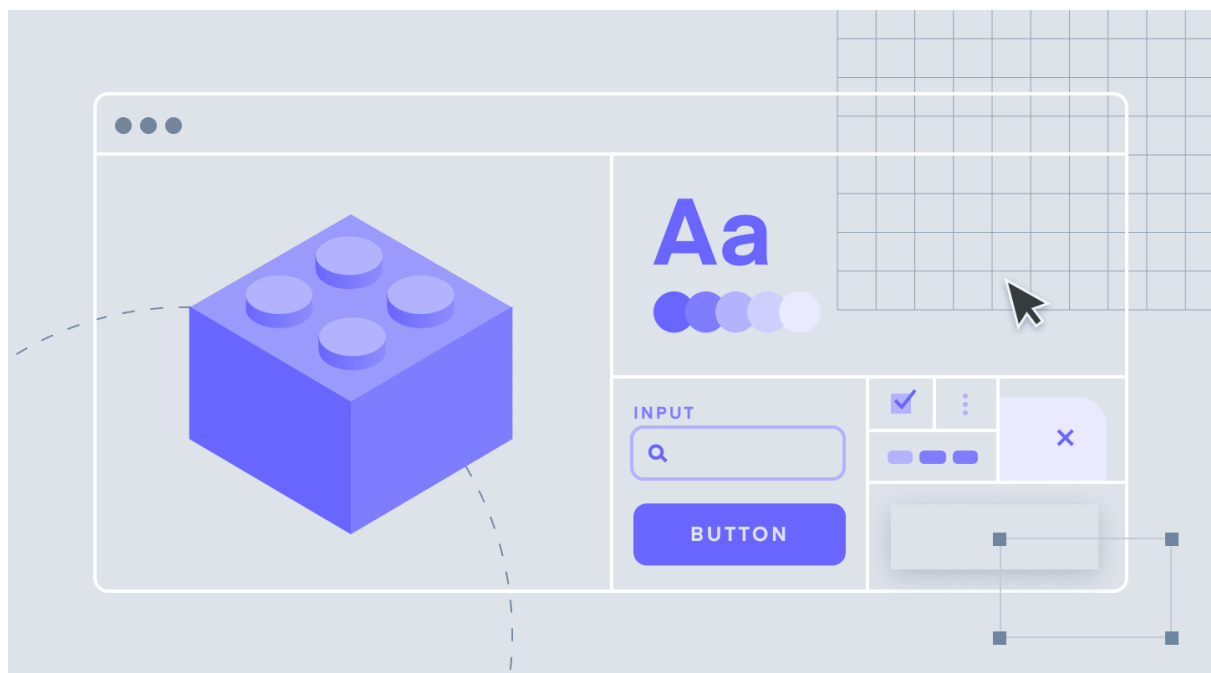
Fahrat gaf aan dat hij het helemaal niet erg vond om mij te helpen en had al vaker bij Touchtribe stagiaires geholpen. Dit gaf mij een fijn gevoel.

## **Bijlagen**

Bijlage 1 Theoretisch kader en literatuuronderzoek

# **Theoretisch kader en literatuuronderzoek**

## **Designsysteem**



## Hoofdstuk 1 Een designsysteem

### 1.1 Atomic Design

Bij een optimale gebruikerservaring draait het voornamelijk allemaal om consistentie, hier speelt een designsysteem een grote rol. Een designsysteem is een set aan standaarden en richtlijnen over hoe je dingen maakt en ontwerpt, denk hierbij aan een brand- of styleguide. In de guide staat waar je rekening mee moet houden als je met een merk of bedrijf aan de slag gaat (Soda Studio, 2021). Enkele voorbeelden hiervan zijn lettertypes, kleuren, afmetingen enzovoort. Om dieper in te gaan op een designsysteem, wordt het onderverdeeld in verschillende onderdelen: atomen, moleculen en organismes (Woznica, 2018). Deze Atomic Components vormen samen een Atomic Design, zie *Afbeelding 1 Atomic Components*.



*Afbeelding 1 Atomic Components*

<b>Doorzoek de site</b>	<b>LABEL</b>	Atomen zijn de kleinste delen van een project, zoals de buttons, kleuren, titels, labels, lettertypes, paragrafen, enzovoort. Deze atomen kunnen worden toegepast op elke context, globaal of binnen andere componenten, zie <i>Afbeelding 2 Voorbeeld atomen</i> . Het zijn erg belangrijke, maar afzonderlijke onderdelen. Zonder de moleculen hebben ze niet veel betekenis (Oostenrijk, 2022).
<input type="text" value="Input"/>	<b>INPUT</b>	
<b>Vind &gt;</b>	<b>BUTTON</b>	

*Afbeelding 2 Voorbeelden atomen*



Moleculen daarentegen zijn een samenstelling van meerdere componenten of atomen. Moleculen worden gezamenlijk meer waard. In *Afbeelding 3 Voorbeeld moleculen*, is te zien dat de losse atomen nu betekenis geven om een gebruiker bepaalde dingen te laten opzoeken. Het wordt een zoekbalk. Elke molecuul kan een eigen eigenschap hebben en functionaliteiten creëren door atomen te gebruiken, die op zichzelf geen enkele functie of actie hebben.

## Doorzoek de site



*Afbeelding 3 Voorbeeld moleculen*

Het derde onderdeel zijn de organismen. Organismen zijn een combinatie van moleculen die samenwerken of zelfs met atomen die meer uitgebreide grensvlakken vormen, ook wel de bouwstenen. Een voorbeeld van een organismen is een header van een website. Deze header bestaat uit een logo, de navigatie en de zoekfunctie, zie *Afbeelding 4 Voorbeeld organismen*.



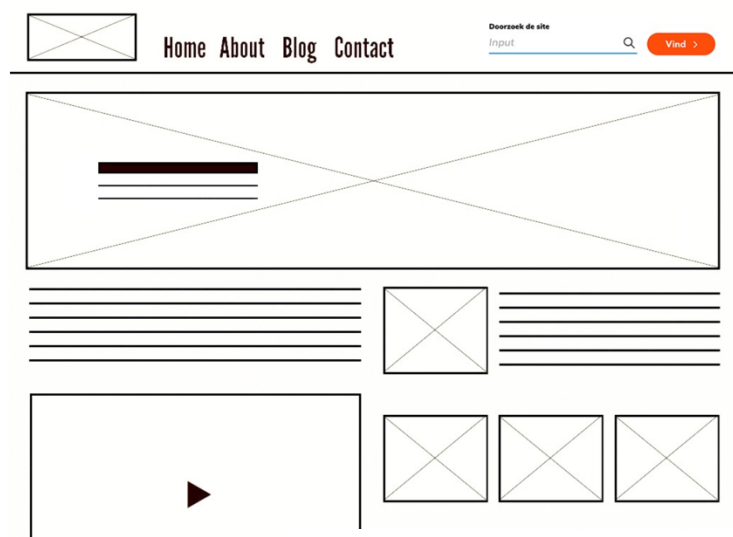
*Afbeelding 4 Voorbeeld organismen*

Om het volledige Atomic Design door te nemen, worden ook de templates en de pages doorgenomen. Templates zijn een blauwdruk van een website of applicatie. Het template bestaat uit gegroepeerde organismen die samen een pagina vormen, zie *Afbeelding 5 Voorbeeld templates*. Tot slot heb je nog de pages, dit is het template met ingevulde content die van toepassing is voor de organisatie of de klant.

Een Atomic Design helpt om dubbele taken te voorkomen, doordat het veel duidelijker is welke componenten hergebruikt kunnen worden en er geen dubbele code wordt geschreven. Door de opbouw van kleine naar grotere onderdelen is het voor programmeurs ook een stuk eenvoudiger om projecten over te dragen.

Naast componenten, houdt een designsysteem ook documenten, artikelen, voorbeelden, codefragmenten, screenshots, ontwerprichtlijnen, filosofieën en andere digitale middelen in (Alli, 2020).

Het wordt meestal online gehost als een website of intern gedocumenteerd. Het is een grote kennisbank die deels bestaat uit een UI-kit, deeldocumentatie met instructies, taal en coderingsrichtlijnen.



*Afbeelding 5 Voorbeeld templates*

## 1.2 Voorbeelden designsysteem

### Hoofdstuk 2 Gebruik van designsysteem

Er zijn vijf redenen waarom er een designsysteem gebruikt moet worden. De eerste reden is dat ontwerp- en ontwikkelingswerk snel en op grote schaal gemaakt kan worden. Bij een ontwerpsysteem staat het vermogen om snel ontwerpen te repliceren door gebruik te maken van vooraf gemaakte UI-componenten centraal. Dit scheelt tijd, doordat ontwerpers steeds dezelfde elementen kunnen gebruiken (Fessenden, 2021). Een manier om deze snelheid te garanderen is geïntegreerde producten en workflows te gebruiken die samenwerking ondersteunen. Dit zal het team helpen om in een sneller tempo te werken. Als alle feedback op één plek verzameld kan worden, kunnen ontwerpers snel versies herzien en publiceren (Wadsworth, 2019).

De tweede reden is dan ook dat ontwerpers zich minder hoeven concentreren op ontwerpbronnen te zoals het visuele uiterlijk en kunnen hierdoor de focus leggen op complexere problemen zoals prioriteitstelling van informatie, workflow optimalisatie en reisbeheer.

De derde reden houdt in dat het een uniforme taal binnen en tussen cross functionele teams creëert. Als ontwerpverantwoordelijkheden verschuiven of wanneer teams veranderen, wordt er tijd verspild aan ontwerp- of ontwikkelingstijd rondom miscommunicatie. Dit is een struikelpunt van Touchtribe dat in Hoofdstuk 1.1.4 wordt Achtergrond beschreven. Een voorbeeld hiervan is dat er nu geen discussie gevoerd hoeft te worden over het uiterlijk van een component zoals een button, omdat dit in het designsysteem teruggevonden kan worden.

De vierde reden is dat het voor virtuele consistentie zorgt tussen producten, kanalen en andere afdelingen. Als teams van verschillende afdelingen onafhankelijk van elkaar opereren, kan het missen van een designsysteem leiden tot discussie.

De vijfde reden is dat het kan dienen als een educatief hulpmiddel en naslagwerk voor ontwerpers op juniorniveau en inhoud bijdragers. Nadrukkelijke geschreven gebruikersrichtlijnen en stijlgidsen helpen individuele bijdragers aan boord te krijgen die nieuw zijn in UI-ontwerp of het maken van inhoud. Ook dienen ze als herinnering voor de rest van de bijdragers.

## Hoofdstuk 3 Tools

### 3.1 Verschillende tools

Er zijn vier type tools die je kan gebruiken voor je designsysteem; design tools, documentation tools, design documentation tools en custom platforms.

#### Design tools

Met design tools zoals Figma, Adobe XD, Sketch en Framer kan je online schermen ontwerpen. Over het algemeen zijn deze tools niet bedoeld voor documentatie (Swierts, 2020). Ze geven volgende opties zoals:

- Het delen van het scherm
- Het exporteren van afbeeldingen
- Online schermen kunnen opslaan
- Het hebben van gedeelde bibliotheken

#### Documentations tools

Documentations tools hosten toegankelijke websites die iedereen kan gebruiken om informatie op te halen. Deze tools richten zich op de ontwikkelingskant en bieden de gebruiker manieren om code te hosten en te kunnen tonen. Tot nu toe bestaat native integratie met ontwerptools nog niet, waardoor ontwerpers afbeeldingen van een ontwerp moeten uploaden. De afbeeldingen moeten vervangen worden bij elke wijziging die ze maken. Voorbeelden van documentations tools zijn Gitbook, VuePress, Fractal en Confluence (Swierts, 2020).

#### Design documentation tools

De tools Frontify, Zeroheight, Nuclino, Lucid en Bynder zijn tools die gericht zijn op design. De layouts die deze tools aanbieden zijn meer ondersteunend voor het tonen van merk en ontwerp. Enkele van deze tools worden automatisch gesynchroniseerd met de gebruiker zijn ontwerptool, waardoor het gemakkelijk is om het ontwerpsysteem up-to-date te houden met de laatste wijzigingen (Swierts, 2020).

#### Custom platforms

Er zijn mogelijkheden om je eigen platform te creëren. Een eigen platform houdt meer vrijheid in met wat mogelijk is. Het kost ook aanzienlijk meer inspanning, vereist technische kennis en ondersteunt geen synchronisatie met de ontwerptool. Enkele voorbeelden zijn Google's Material Design, Shopify's Polaris en IBM's Carbon Design System (Swierts, 2020).

Er is onderzoek gedaan door Hike One naar de verschillende design tools en de prioriteiten naar attributen binnen deze tools. In *Afbeelding 7 Tool Attributes Hike One onderzoek* is de opbouw van Hike One te zien. Zij hebben de attributen geordend op basis van belangrijkheid. Integratie met ontwerptools en prijs zijn altijd grote beslissers, vandaar dat ze bovenaan staan. Goede

ontwerpsystemen integreren ontwikkeling, dus code-opties en versiebeheer sluiten daaronder aan. Daarnaast hebben vrijheid en structuur ook een grote impact op de mogelijkheden. Sommige tools bieden veel opties voor paginalay-out en paginaboomstructuur, terwijl andere een vast sjabloon gebruiken.

De minder belangrijke attributen zijn de mogelijkheden om je eigen domeinnaam te kunnen gebruiken of de mogelijkheid hebben om het uiterlijk van de tool te personaliseren. De prioriteiten van de attributen is voor elk bedrijf anders, dus in het adviesrapport voor Touchtribe kan er een andere tabel komen.

Tool Attributes	Frontify	Zeroheight	Invision DSM	Nuclino	Confluence	Bynder	Notion	Gitbook	Custom
Integration with Sketch/Figma/Adobe	++ Sketch	+++ Sketch, Figma, Adobe XD, Zeplin	+	+	- Figma with plugin	+	+	-	?
Price* (purchase & setting up)	+	+	-	++	+	?	++	+	--
Options for development (e.g. code snippets)	+	++	+	+	+	--	+	+	?
Version Control	-	++	++	++	++	-	++	++	?
Freedom in structure	+	+	-	+	++	--	++	+	++
Custom domain	++	++	--	--	+	++	--	++	++
Tool carries out brand (in style)	++	+	+	-	+	-	-	+	++
Built-in support for multiple styles	+	-	-	-	-	-	-	-	++
Single Sign-On possibilities	+	+	+	++	+	+	+	++	?

++ Very good    + Good    - Bad    -- Very bad    ? Unknown

Afbeelding 7 Tool Attributes Hike One onderzoek

## 3.2 Storybook

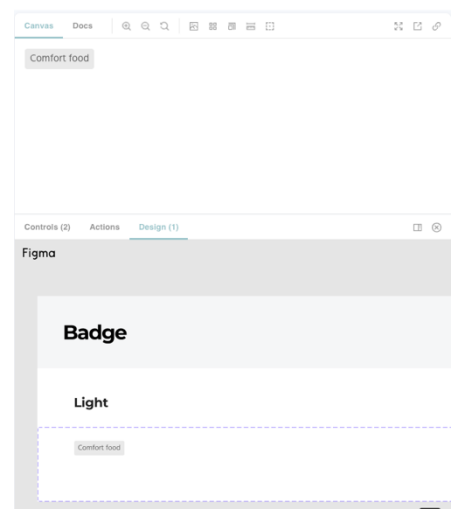
Storybook is een open source tool voor het bouwen van UI componenten en pagina's. Het stroomlijnt de ontwikkeling, het testen en de documentatie van de gebruikersinterface (Storybook, 2022). Storybook kan gebruikt worden voor React, Vue, Angular, Web components, Ember, HTML, Svetle en Preact. Storybook kan gebruikt worden om de een makkelijk overzicht te krijgen van de gemaakte componenten in Figma.

Het zien van een Figma design in Storybook

Om Storybook met Figma te verbinden moet je een aantal stappen doorlopen. Na deze stappen kan je een Figma design in je Storybook omgeving zien (Figma, 2022). Dit Figma bestand update automatisch als er een verandering in de Figma omgeving heeft plaatsgevonden, zie Afbeelding 8 Koppeling Storybook en Figma.

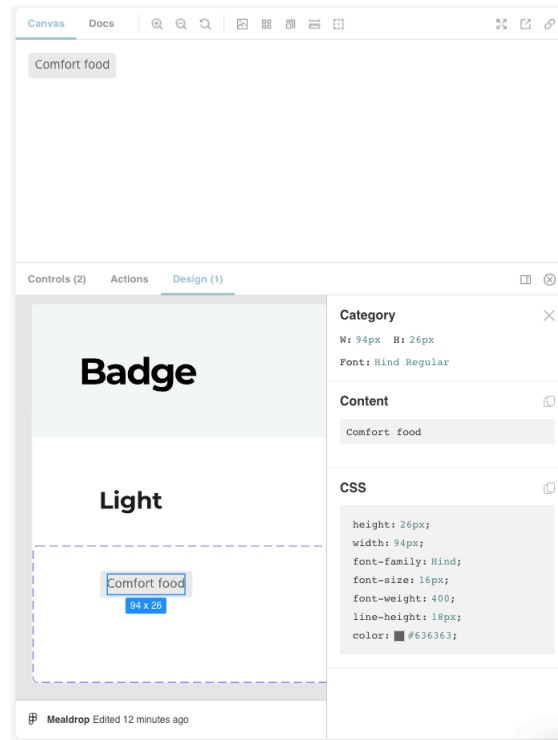
### 3.2.1 Inspecteren in Storybook

Na het toevoegen van het design in Storybook is het design te zien, maar het is niet interactief, zie Afbeelding 9 Koppeling Storybook en Figma. Figma moet nog steeds geopend worden om de tokens te kunnen zien. Figma addon heeft een feature dat ervoor zorgt dat een gebruiker Storybook niet hoeft te verlaten om de design tokens van een Figma object te kunnen zien. Deze feature heeft



Afbeelding 8 Koppeling Storybook en Figma.

**“figspec”**. Deze feature gebruikt een Figma API en vereist een Figma-toegangstoken. Voor deze toegangstoken heb je een Figma account nodig (New Line, sd).



*Afbeelding 9 Koppeling Storybook en Figma.*

### 3.2.2 Voor- en nadelen Storybook

Er zijn enkele voor- en nadelen aan het gebruiken van Storybook gekoppeld met Figma om een nieuw designsysteem op te stellen. Storybook krijgt in de toekomst nog wel plug-ins, maar niet die JSON-tokens versturen. Dit is een belangrijke eis vanuit Touchtribe.

#### Voordelen

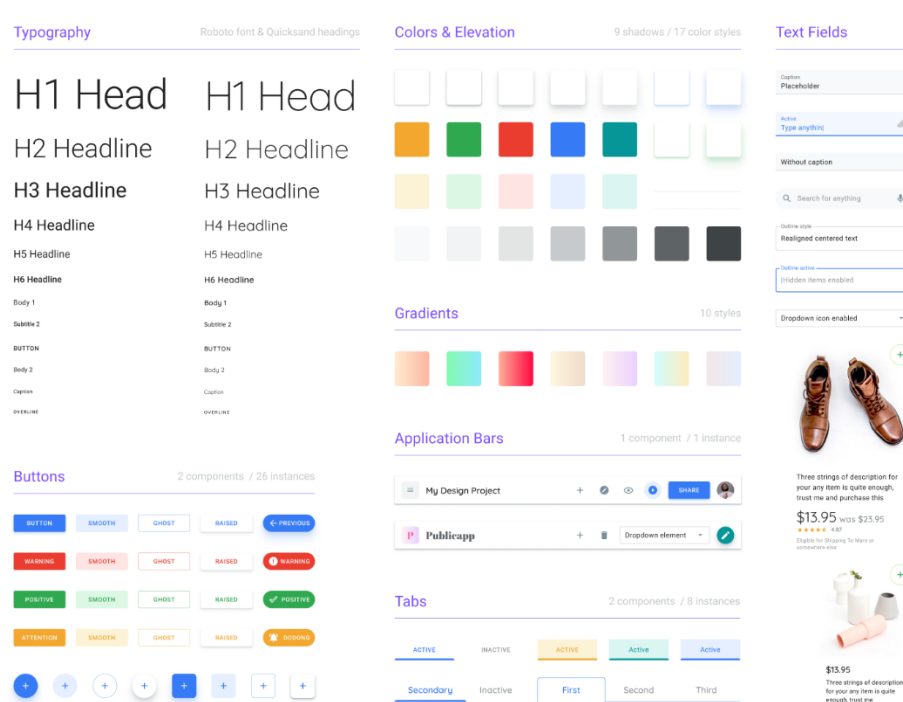
- Enkel component zien; je kan een enkel component te zien krijgen, in plaats van alle componenten van die categorie; zoals Buttons/button-01.
- CSS; als de CSS wordt vergeten neer te zetten in het Figma bestand door de ontwerpers, wordt het in Storybook automatisch neergezet
- Update automatisch; bij aanpassingen vanuit Figma wordt het automatisch geüpdatet in Storybook

#### Nadelen

- Traag; Als Storybook helemaal gevuld is met component duurt het inladen erg lang
- Geen JSON-tokens kunnen versturen; dit kan alleen in Figma zelf
- Voorkeur developers; developers vinden de informatie vanuit Figma halen makkelijker

### 3.3 Figma

Figma is een cloudgebaseerde tool voor het ontwerpen en prototypen van de UI waarmee meerdere teams kunnen samenwerken bij het maken, testen en implementeren van interface- of productontwerpen. Functies omvatten versiegeschiedenis, projectmanagement, opmerkingen, gebruikersgebaseerde machtigingen, het maken van plug-ins en eenmalige aanmelding (Figma, 2022). In *Afbeelding 10 Designsysteem Figma* is een voorbeeld te zien van een designsysteem in Figma.



*Afbeelding 10 Designsysteem Figma*

### 3.3.1 JSON-tokens; Figma Tokens

Er zijn verschillende plug-ins die aangeboden worden vanuit Figma waaruit JSON-tokens verstuurd kunnen worden naar GitHub. Dit is een belangrijke eis vanuit Touchtribe.

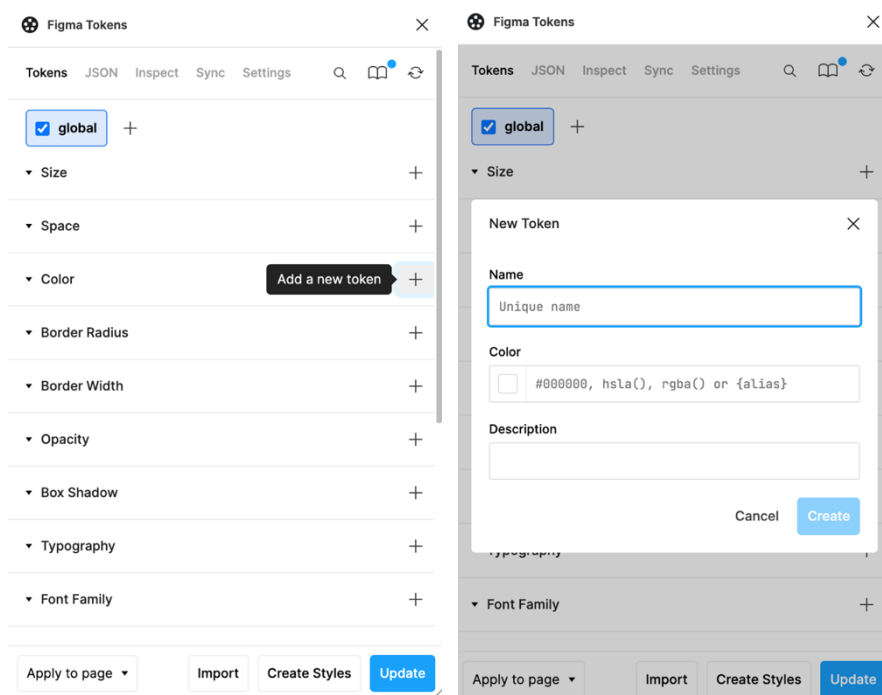
#### Beschikbare tokens

In Figma Tokens kunnen er herbruikbare en semantische token gemaakt worden waarnaar vervolgens naar verwezen kan worden in het ontwerpproces. De beschikbare token groepen zijn als volgt:

Group	Application
Sizing	Width, height
Spacing	Auto-Layout (Gap, padding)
Color	Styles and Layers; Fill, border (right click)
Border radius	All corners, individual
Border width	Border with
Box Shadow	Styles and Layers (one or many, drop/inner)
Opacity	Opacity
Font family	Text layers (in combination with weight)
Font weight	Text layers (in combination with family)
Font size	Text layers
Line height	Text layers
Letter spacing	Text layers
Paragraph spacing	Text layers
TextCase	Text layers
TextDecoration	Text layers
Typography compositions	Text layers, styles

## Creëren tokens

Het creëren van een token via de gebruikersinterface is eenvoudig. Een van de manieren is als volgt; je opent de plug-in en klikt op het plus icon. Door middel van deze knop kan je een token toevoegen. Hierna wordt er een ander venster geopend waarin je de gegevens kan plaatsen. Deze tokens kunnen worden aangepast door met je rechtermuisknop op de gewenste token te klikken. Er verschijnt een optie met *Edit Token*. De tokens kunnen ook gegroepeerd worden, zie *Afbeelding 11 Groeperen tokens*.



*Afbeelding 11 Groeperen tokens*

## Aliassen

In Figma Tokens kan er in waarde naar andere tokens verwezen worden. Dit betekent dat een merkkleur verwezen kan worden als `colors.red.500`, of een achtergrondtoken hebben die zijn waarde ontleent aan een `colors.black` token, terwijl voorgrondtoken zijn waarde ontleent aan een `colors.white` token. Om een alias in de tokens te gebruiken, worden ze in de volgende notatie geschreven: `{spacing.sm}`. Een alias kan ook in de oudere `$` notatie geschreven worden, zoals: `$spacing.sm`. Verwijzingen zijn hoofdlettergevoelig. Dit betekent dat verwijzingen geschreven moeten worden zoals de tokens worden genoemd. Op deze manier weet de plug-in dat hij opzoek moet gaan naar een token van `spacing.sm`, als het gevonden wordt gebruikt hij die waarde. Telkens als `space.sm` wordt gewijzigd, verandert ook het token dat er naar verwijst.

## Stijlen

De plug-in convert automatisch kleur- en typografiestijlen naar tokens. Het beste deel van deze aanpak is dat de plug-in probeert uw basiseenheden te bepalen en hiervoor tokens te maken. Dat betekent dat 4 stijlen die allemaal verwijzen naar *Inter* als een lettertype, met 2 lettergewichten,



Regular en Bold, een set basistokens worden van font-inter, font-weight-bold, verschillende lettergroottes, regelhoogte, letterafstand en alinea en een set Typografie tokens.

### **Synchroniseren met externe bronnen**

Figma Tokens kan gesynchroniseerd worden met de volgende bronnen:

- JSONBin
- GitHub; push en pull naar repository. Gecombineerd met GitHub Actions en style-richtingen creëert dit je eigen pijplijn.
- URL (read-only)

### 3.3.2 JSON-tokens; Design tokens

De plug-in Design Tokens is door Luka Oppermann ontwikkeld. Met Design Tokens kan je je Figma ontwerp omzetten in JSON-tokens. Deze JSON-tokens kunnen gegenereerd worden als een zipfile en naar de GitHub worden gestuurd. De plug-in zet de stijlen die worden gedefinieerd in Figma om in ontwerptokens. Dit omvat tekst-, kleur, raster- en effectstijlen.

#### Aliassen

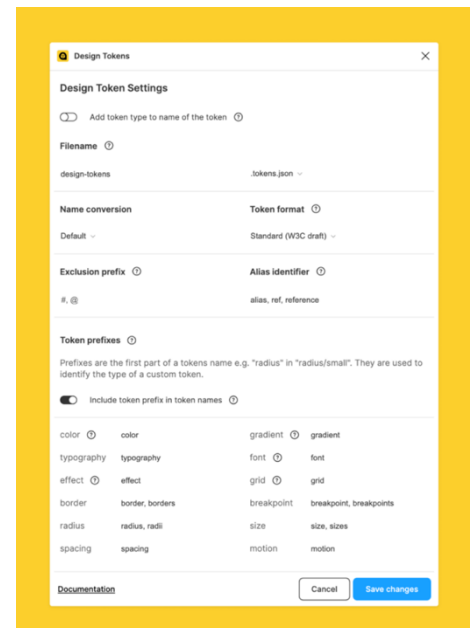
Met het standaard token format kan er een alias/referentie voor kleurtokens gedefinieerd worden via het beschrijvingsveld, zie *Afbeelding 12 Design Tokens Settings*.

Een voorbeeld; bij de kleurstijl danger-color dat verwijst naar de kleurstijl colors/red kan een ref:regel worden toegevoegd aan het beschrijvingsveld van de danger-color:

*Use this color for destructive actions only.*  
*ref: core-colors.red*

Dit wordt gecreëerd in JSON als:

```
"danger-color": {  
  "description": "Use this color for destructive actions only.",  
  "type": "color",  
  "value": "{core-colors.red}"  
};
```



*Afbeelding 12 Design Tokens Settings*

#### Stijlen

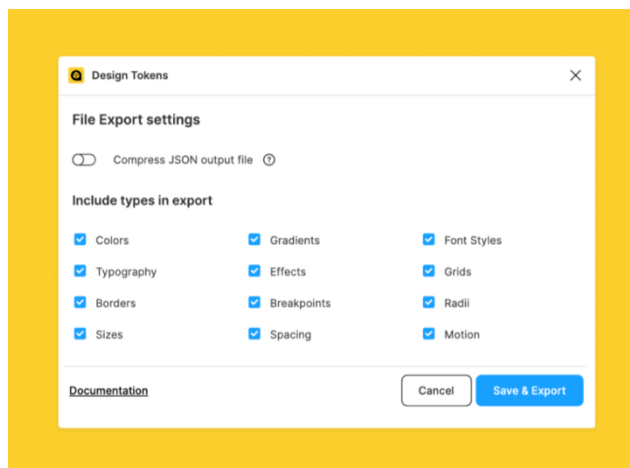
De plug-in ondersteunt aangepaste tokens voor randen, radii, maten, afstand en beweging. Elk aangepast ontwerptopen moet zich binnen een frame op het hoogste niveau bevinden met een naam die begint met `_tokens`. Dit betekent dat de structuur als volgt is: pagina > `_tokens/sizes` > `sizes/8`. De token zelf moet een naam hebben die begint met `sizes`, `spacing`, `borders`, `radii` of `motion` en moet de Main Component of een variant zijn. Dit is zodat de plug-in kan identificeren wat geen token is.

#### Zipfile

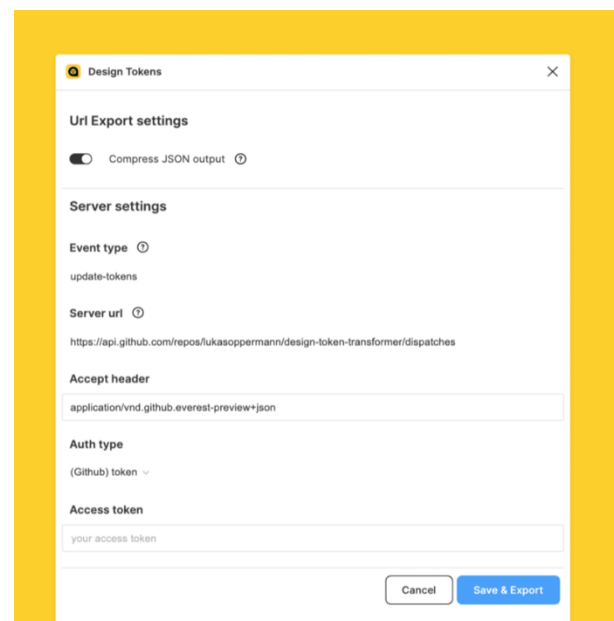
Er is een optie om de JSON-tokens in een zipfile te ontvangen. Er kan worden gekozen om een compressie van dit bestand te maken. Hiermee wordt het bestand kleiner, maar worden de witruimte en regeleinden verwijderd, zie *Afbeelding 13 File Export Settings (zip)*. De zipfile kan een naam naar keuzen krijgen.

## Synchroniseren GitHub

Voor het versturen naar GitHub moeten enkele gegevens worden ingevuld zoals de event type, server URL, accept header, auth type en acces token, zie *Afbeelding 14 URL Export Settings (GitHub)*. De belangrijkste gegevens zijn de server URL en acces token. De server URL moet het volgende bevatten: <https://api.GitHub.com/repos/:username/:repo/dispatches>. De acces token is je personal acces token van GitHub. Naast het opzetten van de repository, moet je enkele bestanden vanaf een andere repository genaamd Design Tokens Transformer halen. Het gaat om de workflows, package-lock.json en package.json. Zonder deze bestanden doet de converter het niet.



*Afbeelding 13 File Export Settings*



*Afbeelding 14 URL Export Settings (GitHub)*

### 3.2.3 Vergelijking Figma Tokens vs Design Tokens

Doordat het development team niet op de hoogte was van de verdere vooringingen van het ontwerpteam, had ik mij inmiddels verdiept in een plug-in genaamd Design Tokens. Om te kijken welke plug-in het beste gaat werken voor het team, heb ik de voor- en nadelen onder elkaar gezet.

#### **Design Tokens**

##### *Voordelen*

- Design Tokens kan de tokens in een .zip opleveren die gedownload kan worden op een device
- Kan naar GitHub gestuurd worden

##### *Nadelen*

- Is door een developer gebouwd; is niet helemaal waterdicht
- Verzend af en toe de tokens niet; zijn niet goed gelabeld zijn
- Als developer stopt, houdt de plug-in op
- Aparte bestanden voor gebruik; voor gebruik moet je bijvoorbeeld eerst de package-json van de Design-Tokens-Transformer overnemen in je eigen package-json bestand.

#### **Figma Tokens**

##### *Voordelen*

- Plug-in wordt vertegenwoordigd vanuit Figma; amper bugs
- Kan naar GitHub gestuurd worden
- Eenvoudiger; kan je eigen namen geven

##### *Nadelen*

- Als developers stoppen, houdt de plug-in op

### 3.2.4 Voor- en nadelen Figma

Er zijn enkele voor- en nadelen aan het gebruiken van Figma om een nieuw designsysteem op te stellen.

#### **Voordelen**

- Momenteel wordt er al meegewerkt door designers en developers
- JSON-tokens; er kunnen JSON-tokens verstuurd worden naar GitHub
- Handige plug-ins; eindeloze handige plug-ins die een designsysteem ondersteunen

#### **Nadelen**

- Verkeerde gelabelde namen; ontwerpers labelen niet altijd de componenten juist, waardoor developers heel erg moeten zoeken naar componenten
- Zien alle componenten; je ziet alle componenten naast elkaar, in plaats van dat je ook enkele kan zien.

## 3.4 Inhoud goed designsysteem

### 3.4.1 Richtlijnen

Een designsysteem bestaat uit verschillende delen om zoveel mogelijk structuur te geven:

#### **Content, informatie en doelen**

Bij het opstellen van een designsysteem is het erg belangrijk om over de doelen van de eindgebruikers na te denken. Welke informatie is voor wie belangrijk? Het is belangrijk om gesprekken te voeren met de eindgebruikers om de wensen en behoeften hierin te verwerken. Zodra het doel bepaald is en het helder is wat er in een bedrijf al aanwezig is, kan er begonnen worden met het ontwikkelen van een nieuw designsysteem (Hacq, 2018).

#### **Elementen en componenten**

De beste designsystemen bevatten zowel UI-elementen (een catalogus van herbruikbare componenten), als standaard merkgidsen, waaronder het kleurenpalet, typografie en iconografie. Verder bestaat het uit gegevensvisualisatie, illustraties, paginastructuur en framework, toegankelijkheidsrichtlijnen en ontwerpprincipes. Ontwerpprincipes zijn leidende zinnen die de teams helpen om het doel van het product te bereiken dankzij het ontwerp. Ze helpen de teams om zinvolle ontwerpbeslissingen te nemen. (Hacq, 2018).

#### **Ontwerpers als designsysteem gebruikers**

Het is voor ontwerpers erg belangrijk dat er gedocumenteerd staat hoe en wanneer elk onderdeel gebruikt moet worden. Dit helpt om ontwerpers vrij om hun focus te leggen aan het besteden aan het bedenken van oplossingen van unieke uitdagingen, in plaats zich te concentreren op componenten die al in gebruik zijn opnieuw uit te vinden, (Cunningham, 2022). De elementen moeten de volgende punten bevatten om ontwerpers te helpen begrijpen hoe ze het moeten gebruiken:

- *Beschrijving*: Wat is dit element? Wat is het doel?
- *Gebruiksopmerkingen*: Wanneer en waar moet dit element gebruikt worden?
- *Verwante elementen*: Als dit element op een ander element lijkt, overweeg dan om naar andere verwante elementen te verwijzen. In veel gevallen kan het handig zijn om te schetsen wanneer het ene of het andere element gebruikt moet worden gebruikt, afhankelijk van de gebruikerssituatie en het doel.
- *UX of visuele opmerkingen* (optioneel): Aanvullende opmerkingen over de interactie of het visuele ontwerp van het element.
- *Opmerkingen voor ontwikkelaars* (optioneel, tenzij ontwikkelaars ook beoogde gebruikers zijn): Aanvullende opmerkingen over hoe een component is gebouwd.

#### **Developers als designsysteem gebruikers**

Het hangt van de doelen van het designsysteem af of organisaties het designsysteem gebruiken om ook de code voor elk UI-element te catalogiseren. Hierdoor kunnen ontwikkelaars het designsysteem gebruiken om standaardcomponenten te gebruiken in plaats van ze elke keer anders te bouwen. Als het designsysteem ook gebruikt wordt door developers, moet ervoor gezorgd worden dat de meegeleverde code en de dev-opmerkingen even uitgebreid zijn als de ontwerpelementen. Op deze manier wordt het werk van de developers zo gestroomlijnd mogelijk te maken.

### 3.4.2 Typografie

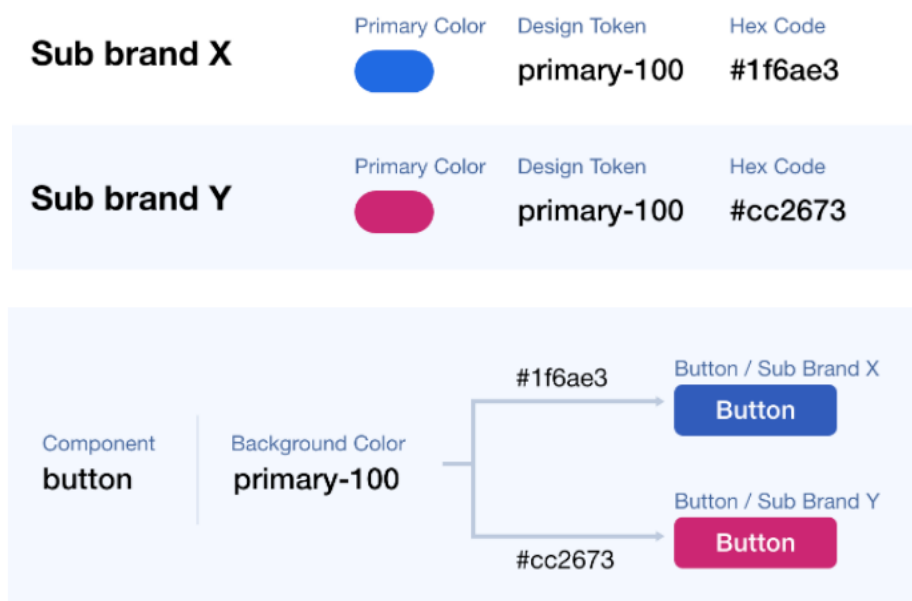
Typografie is een van de belangrijkste aspecten van een designsysteem om in een vroeg stadium vast te stellen. Door middel van typografie brengt de website of applicatie de informatie over aan de gebruikers. Het is vaak een van de fundamentele componenten waar je grip op wilt krijgen, omdat het harmonieus samen moet werken met andere elementen zoals pictogrammen. Zelfs de lijnhoogten kunnen zelfs veel andere structurele elementen beïnvloeden, zoals rasters (Design Systems, sd).

### 3.4.3 Kleuren

Naast het opzetten van de typografie is het opzetten van een kleurenpalet ook erg belangrijk om daarna door te kunnen met het verder ontwerpen van componenten, patronen en sjablonen (Choudhary, 2021).

#### Labelen kleuren

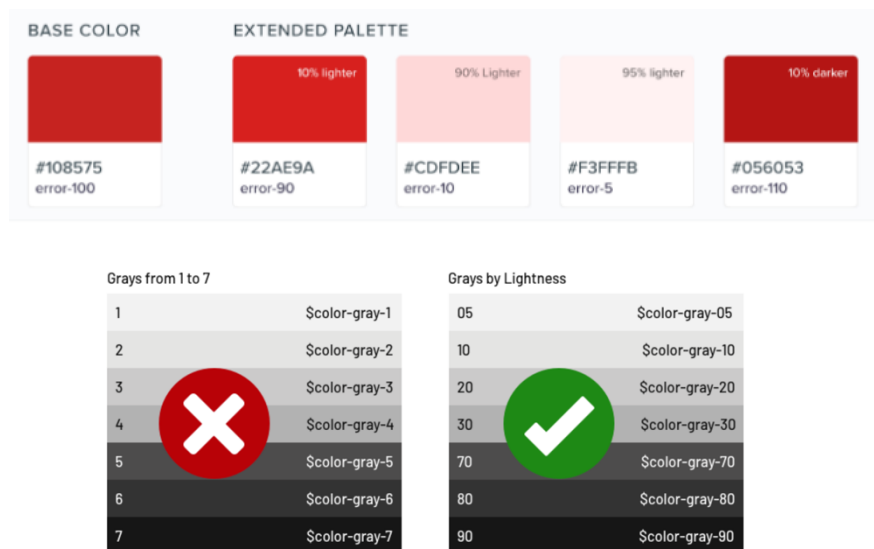
Het is erg van belang dat de namen van de kleuren goed gelabeld worden, omdat er met meerdere partijen samengewerkt moet worden. Namen zoals 'primary color' is veel makkelijker om door te communiceren dan hex-waarde zoals '#1F6AEE3', zie *Afbeelding 15 Voorbeeld Primary Color*.



*Afbeelding 15 Voorbeeld Primary Color*

Elke kleurenlabel representeert een basiskleur, terwijl de 'XX' nummers verschillende tinten representeren; zoals 'color label-XX'. Het wordt aangeraden om kleuren een naam te geven die gebaseerd zijn op hun functies en niet de tint, zie *Afbeelding 16 Voorbeeld kleurenlabel*. De schaal weerspiegelt een bekend bereik van donker naar licht, maakt het mogelijk om nieuwe opties ertussen te injecteren. De waarde van de 'XX' hangt af van de volgende drie aspecten:

- Basiskleur: 100
- Tinten donkerder dan de basiskleur: > 100
- Tinten lichter dan de basiskleur: < 100



*Afbeelding 16 Voorbeeld kleurenlabel*

Daarnaast is het handig om in het secundaire palet kleuren een feedbackkleur te geven. Zo staan bij de meeste systemen rood voor fouten, groen voor succes, geel voor waarschuwing en blauw voor informatieve berichten. Feedbackkleur is van cruciaal belang, omdat deze bovenaan de pagina staat en in wisselwerking staat met andere belangrijke componenten en/of wordt aangetroffen als gevolg van een ongewenste omstandigheid (Curtis, 2016).

## Nieuwe kleuren platten valideren

Een andere belangrijke overweging is die van de toegankelijkheid. Controleer of er voldoende contrast is tussen de achtergrond- en tekstelementen (W3, 2016). Er moet gestreefd worden om te voldoen aan de WCAG-richtlijnen voor kleurcontrast, zie *Afbeelding 17 WCAG-richtlijnen*.

Element Type	Contrast Ratios
Standard text Text is less than 18 point if not bold and less than 14 point if bold	4.5 : 1
Large Text Text is more than 18 point if not bold and more than 14 point if bold	3 : 1
UI elements	3 : 1

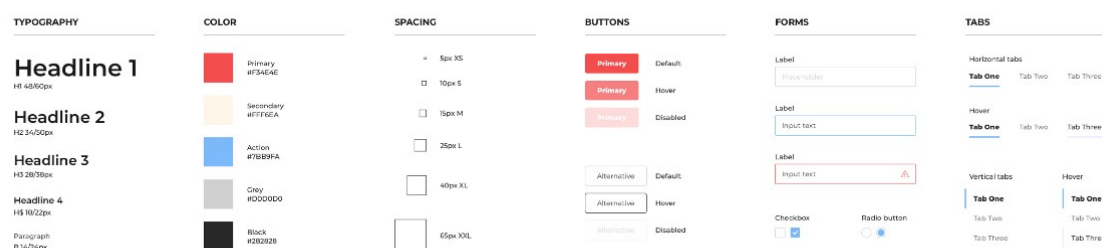
### *Afbeelding 17 WCAG-richtlijnen*

## Feedback toepassen voor uitbrengen

Het is van belang dat het kleurenpalet, net zoals het designsysteem, eerst naar de belanghebbende gaat voor feedback voordat deze uitgebracht wordt. Hiervoor zijn de sprints erg handig. Er kan per sprint feedback gegeven worden vanuit het development-team voordat er van het designsysteem JSON-tokens gemaakt worden.

## 3.5 Voorbeelden designsytemen

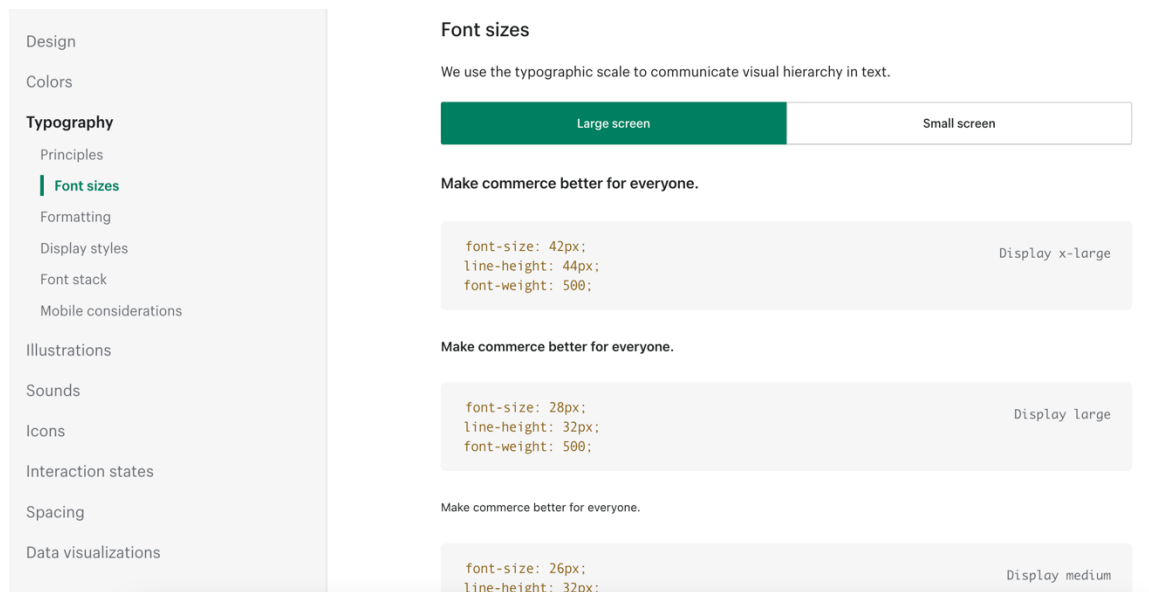
In *Afbeelding 4 Voorbeeld designsysteem* is een designsysteem te zien. Hier worden de typografie meegegeven, de kleuren met HEX-waarde. De spacing, als een ontwerp die bevat en andere mogelijke variabelen die een designsysteem kan bevatten zoals buttons, forms en tabs.



### *Afbeelding 4 Voorbeeld designsysteem*

In *Afbeelding 5 Voorbeeld Shopify* is het designsysteem van Shopify te zien. Dit is een website waar alle styleguide van Shopify te vinden is.





*Afbeelding 4 Voorbeeld Shopify*

## Bibliografie

- Soda Studio. (2021, mei 25). *Design systems voorbij design*. Opgehaald van Sodastudio: <https://www.sodastudio.nl/2021/05/25/design-systems-voorbij-design/>
- Woznica, D. (2018, augustus 1). *Atomic Design and ReactJS*. Opgehaald van Danilowoz: <https://danilowoz.com/blog/atomic-design-with-react>
- Oostenrijk, V. (2022). *Atomic design: wat is het en hoe kan het bijdragen aan een goed webdesign?* Opgehaald van Whello: <https://whello.nl/marketing-tips/web-design/atomic-design-wat-het-en-hoe-kan-het-bijdragen-aan-een-goed-webdesign/>
- Alli, E. (2020, juni 3). *10 Best Design Systems and How to Learn (and Steal) From Them*. Opgehaald van Designerup: <https://designerup.co/blog/10-best-design-systems-and-how-to-learn-and-steal-from-them/>
- Fessenden, T. (2021, april 11). *Design Systems 101*. Opgehaald van Nn Group: <https://www.nngroup.com/articles/design-systems-101/>
- Wadsworth, T. (2019, oktober 7). *Building a Design System: Speed, Scale, Collaboration and Innovation*. Opgehaald van XD Adobe: <https://xd.adobe.com/ideas/principles/design-systems/building-design-system-speed-scale-collaboration-innovation/>
- Kuipers, D. (2022). *Boilerplate*. Opgehaald van Online Marketing Agency: <https://onlinemarketingagency.nl/marketingtermen/boilerplate/>
- Figma. (2022). *Figma Tokens*. Opgehaald van Figma: <https://docs.tokens.studio>
- Oracle. (2022). *Content Blocks*. Opgehaald van Docs Oracle: <https://docs.oracle.com/en/cloud/saas/marketing/eloqua-user/Help/ContentBlocks/ContentBlocksOverview.htm#:~:text=Content%20blocks%20are%20groups%20of,%2C%20shared%20content%2C%20and%20more.>
- Responsive Design. (sd). *Defining Breakpoints*. Opgehaald van Responsive Design : <https://responsivedesign.is/strategy/page-layout/defining-breakpoints/>
- Swierts, S. (2020, december 10). *The best tools for documenting design systems*. Opgehaald van Hike: <https://hike.one/update/the-best-tools-for-documenting-design-systems>
- Storybook. (2022). *Build component driven UIs faster*. Opgehaald van Storybook: <https://storybook.js.org/>
- Figma. (2022). *Storybook and Figma*. Opgehaald van Figma: <https://help.figma.com/hc/en-us/articles/360045003494-Storybook-and-Figma>
- New Line. (sd). *Connecting Figma with Storybook*. Opgehaald van Newline: <https://www.newline.co/courses/storybook-for-react-apps/connecting-figma-with-storybook>
- Figma. (2022). *Figma*. Opgehaald van Figma: [www.figma.com](https://www.figma.com)
- Hacq, A. (2018, mei 22). *Everything you need to know about Design Systems*. Opgehaald van UX Design: <https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969>
- Cunningham, K. (2022). *What is a UX Design System?* Opgehaald van Fuzzymath: <https://fuzzymath.com/blog/what-is-ux-design-system/>
- Design Systems. (sd). *Typography Guides*. Opgehaald van Design Systems: <https://www.designsystems.com/typography-guides/>
- Choudhary, A. (2021, januari 30). *Defining colors in your design system*. Opgehaald van UX Design: <https://uxdesign.cc/defining-colors-in-your-design-system-828148e6210a>
- Curtis, N. (2016, april 13). *Color in Design Systems*. Opgehaald van Medium: <https://medium.com/eightshapes-llc/color-in-design-systems-a1c80f65fa3>
- W3. (2016). *Understanding WCAG 2.0*. Opgehaald van W3: <https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html>

- Fission. (2021). *Screaming Snake Case* . Opgehaald van Fission codes:  
<https://fission.codes/blog/screaming-snake-case/>
- Amzn GitHub. (sd). *Style Dictionary*. Opgehaald van Amzn GitHub:  
<https://amzn.github.io/style-dictionary/#/README>
- Npm. (2022). *Token-transformer v0.0.19*. Opgehaald van Npm:  
<https://npm.io/package/token-transformer>
- Speek, M. K. (2012, september 27). *Wat is een Content Management Systeem (CMS)?*  
 Opgehaald van Max: <https://www.max.nl/blog/content-management-systeem/>
- Touchtribe. (2022). *Touchtribe: Beyond e-commerce*. Opgehaald van Touchtribe:  
<https://www.touchtribe.nl/>
- Ward, C. (2020, Oktober 6). *An Introduction To Developer Experience (DevEx, DX)*.  
 Opgehaald van Humanitec: <https://humanitec.com/blog/developer-experience>
- Knoot, J. (2020). *FOETSJE-MODEL*. Opgehaald van Strategisch marketingplan:  
<https://www.strategischmarketingplan.com/marketingmodellen/foetsje-model/>
- Benders, L. (2021, oktober 26). *Scribbr*. Opgehaald van Reflecteren met de STARR-methode  
 | Stappenplan + voorbeelden: <https://www.scribbr.nl/stage/starr-methode/#:~:text=De%20meestgebruikte%20methode%20om%20te,hel%20resultaat%20en%20de%20reflectie.>
- Benders, L. (2019, november 18). *Reflecteren met het Model van Korthagen*. Opgehaald van  
 Scribbr: <https://www.scribbr.nl/stage/model-van-korthagen/>

## Bijlage 2 Interviews

**Vincent Koopmans**

**Interview 1:** <https://youtu.be/DbnOQOMXd-A>

**Interview 2:** [https://youtu.be/yvYN\\_IYSnq8](https://youtu.be/yvYN_IYSnq8)

## Bijlage 3 Figma bestand

**Link naar Figma bestand:** <https://www.figma.com/file/ZPxCjLFucvplyif45pMJ1Z/Styleguide-metingen?node-id=0%3A1>

## Bijlage 4 Nul- en éénmeting

**Nulmeting**

**Robin van Vliet**

**Video:** [https://youtu.be/zZcYKDmTR\\_A](https://youtu.be/zZcYKDmTR_A)

**Tim Huijg**

**Video:** <https://youtu.be/hVbbTXhFr5E>

**Linda van Dijk**

**Video:** <https://youtu.be/3Tz0AqG7wNU>

**Jasper Zuyderwijk**

**Video:** <https://youtu.be/OOx5kjopad0>

**Éénmeting**

**Robin van Vliet**

**Video:** [https://www.youtube.com/watch?v=zZcYKDmTR\\_A](https://www.youtube.com/watch?v=zZcYKDmTR_A)

**Tim Huijg**

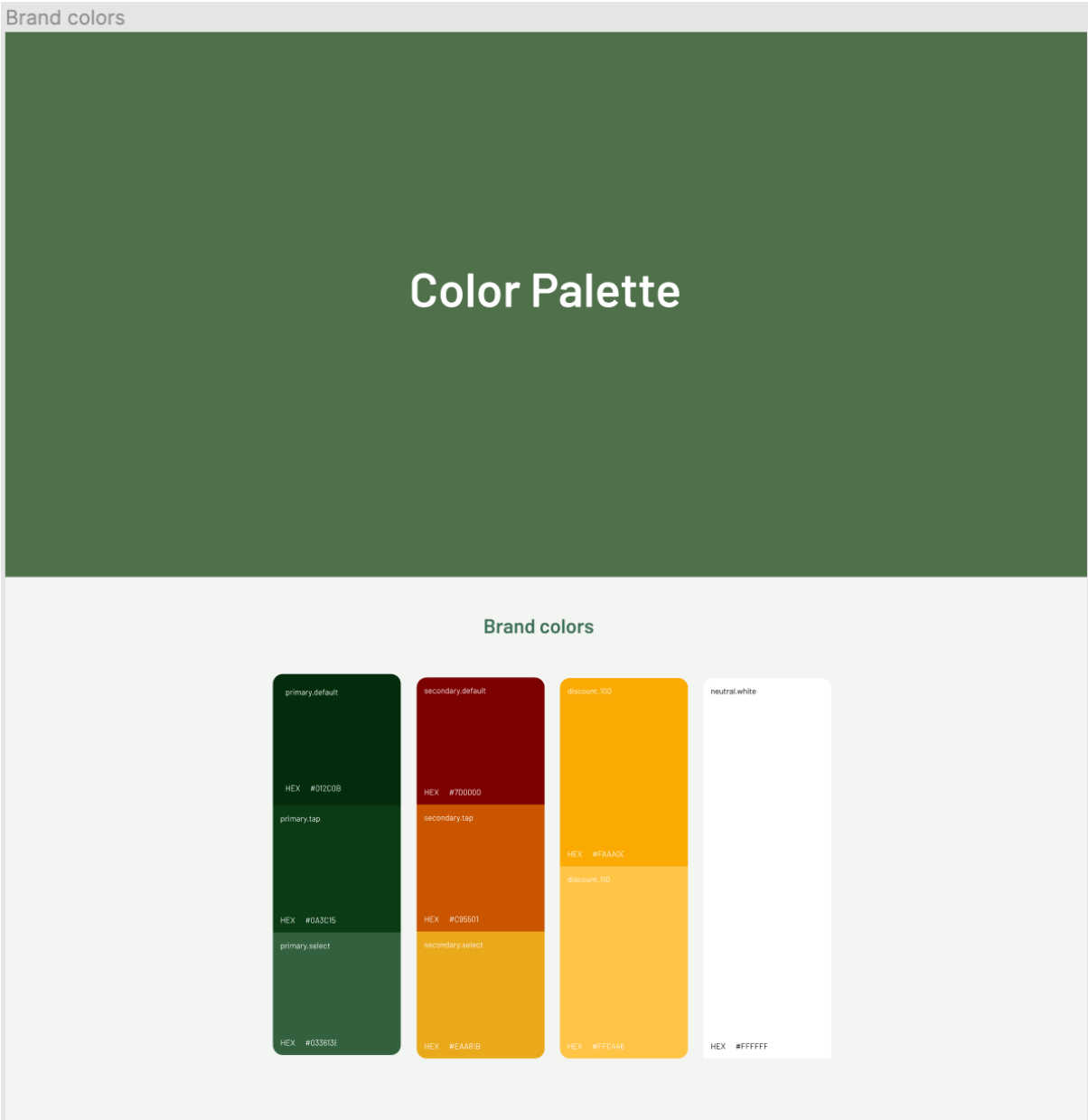
**Video:** <https://youtu.be/UVLfTrenAsU>

**Linda van Dijk**

**Video:** [https://youtu.be/H8Q34e\\_maPU](https://youtu.be/H8Q34e_maPU)

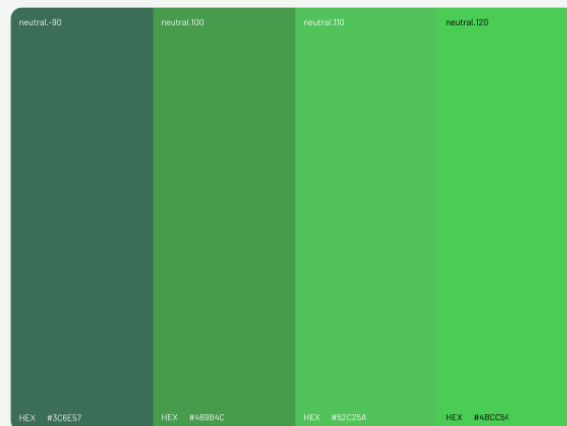
**Jasper Zuyderwijk**

**Video:** <https://youtu.be/yyS9JlrSpos>



## Neutral colors

### Neutral colors



## Feedback colors

### Feedback colors



# Typography

Font

**Barlow**

Font weight

**Bold** Regular Light **Black**

Sizing

Heading 01

**Aa**

32px

Heading 02

**Aa**

28px

Heading 03

**Aa**

24px

Heading 04

**Aa**

16px

Heading 05

**Aa**

14px

# Box-shadow



**card**

box-shadow: 0px 4px 20px  
rgba(0, 0, 0, 0.1);



**tabbar**

box-shadow: 0px 4px 20px  
rgba(0, 0, 0, 0.18); 0px -1px 8px  
20px rgba(0, 0, 0, 0.05);








**button**

box-shadow: 0px 0px 8px  
rgba(0, 0, 0, 0.1);



# Border-radius

	x-small	0
	small	4
	medium	6
	large	12
	x-large	20

DesignPrototypeInspect

Code

CSS >

background: #E5E5E5;

Text Styles

typography

Ag HEADING\_01 · 32/0

Ag HEADING\_02 · 28/Auto

Ag HEADING\_03 · 24/Auto

Ag HEADING\_04 · 16/Auto

Ag HEADING\_05 · 14/Auto

Color Styles

colors

SECONDARY\_TAP

SECONDARY\_DEFAULT

SECONDARY\_SELECT

PRIMARY\_TAP

NEUTRAL\_WHITE

DISCOUNT\_100

DISCOUNT\_110

DesignPrototypeInspect

Color Styles

colors

SECONDARY\_TAP

SECONDARY\_DEFAULT

SECONDARY\_SELECT

PRIMARY\_TAP

NEUTRAL\_WHITE

DISCOUNT\_100

DISCOUNT\_110

PRIMARY\_PINK

NEUTRAL\_-90

NEUTRAL\_100

NEUTRAL\_110

NEUTRAL\_120

FEEDBACK\_SUCCESS\_LIGHT

FEEDBACK\_SUCCESS

FEEDBACK\_SUCCESS\_DARK

FEEDBACK\_WARNING\_LIGHT

FEEDBACK\_WARNING

FEEDBACK\_WARNING\_DARK

DesignPrototypeInspect

NEUTRAL\_100

NEUTRAL\_110

NEUTRAL\_120

FEEDBACK\_SUCCESS\_LIGHT

FEEDBACK\_SUCCESS

FEEDBACK\_SUCCESS\_DARK

FEEDBACK\_WARNING\_LIGHT

FEEDBACK\_WARNING

FEEDBACK\_WARNING\_DARK

FEEDBACK\_ERROR\_LIGHT

FEEDBACK\_ERROR

FEEDBACK\_ERROR\_DARK

PRIMARY\_DEFAULT

PRIMARY\_SELECT

Effect Styles

boxshadow

CARD

TABBAR

Layers	Assets*	Page 1 ▾
#	Border-radius	
#	Box-shadow	
#	button	
	Button	
T	box-shadow: 0px 0px 8px rg...	
T	button	
#	tabbar	
T	box-shadow: 0px 4px 20px r...	
T	tabbar	
	Tabbar	
#	card	
	Rectangle 86	
T	box-shadow: 0px 4px 20px r...	
T	card	
T	Box-shadow	
	Rectangle 39384	
#	Typography	
	Heading 01	
#	Heading 01	
T	Aa	
T	32px	

Layers	Assets*	Page 1 ▾
#	Neutral colors	
#	Feedback colors	
	Color	
	Color/feedback.error-dark	
	Codes	
T	feedback.error-dark	
	Color/feedback.error	
	Codes	
T	feedback.error	
	Color/feedback.warning-light	
	Color	
	Color	
T	Feedback colors	
	Rectangle 18	
	Rectangle 18	
#	Brand colors	
	Color	
	Color/secondary.select	
	Color/secondary.tap	
	Color/secondary.default	

```

{
  "colors": {
    "SECONDARY_TAP": {
      "value": "#c95501",
      "type": "color"
    },
    "SECONDARY_DEFAULT": {
      "value": "#7d0000",
      "type": "color"
    },
    "SECONDARY_SELECT": {
      "value": "#eaa81b",
      "type": "color"
    },
    "PRIMARY_TAP": {
      "value": "#0a3c15",
      "type": "color"
    },
    "NEUTRAL_WHITE": {
      "value": "#ffffff",
      "type": "color"
    },
    "DISCOUNT_100": {
      "value": "#faaa00",
      "type": "color"
    },
    "type": "boxShadow"
  },
  "borderradius": {
    "X-SMALL": {
      "value": "0px",
      "type": "borderRadius"
    },
    "SMALL": {
      "value": "4px",
      "type": "borderRadius"
    },
    "MEDIUM": {
      "value": "8px",
      "type": "borderRadius"
    },
    "LARGE": {
      "value": "12px",
      "type": "borderRadius"
    },
    "X-LARGE": {
      "value": "20px",
      "type": "borderRadius"
    }
  },
  "typography": {
    "HEADING_01": {
      "value": {
        "fontFamily":
"$fontFamilies.barlow",
        "fontWeight":
"$fontWeights.barlow-0",
        "lineHeight":
"$lineHeights.0",
        "fontSize":
"$fontSize.4",
        "letterSpacing":
"$letterSpacing.0",
        "paragraphSpacing":
"$paragraphSpacing.0",
        "textCase":
"$textCase.none",
        "textDecoration":
"$textDecoration.none"
      },
      "type": "typography"
    },
    "HEADING_02": {
      "value": {

```

## Bijlage 6 Casus designers

Hoi!

Ik heb onderzoek gedaan naar het opstellen van een designsysteem en daarbij een juiste styleguide opstellen. Je moet enkele kleine opdrachten doen door middel van een stappenplan om te kijken of er bepaalde dingen nog niet duidelijk zijn. Het gaat hier niet om het opbouwen van frames en componenten, maar om het labelen van de juiste elementen en het uiteindelijk kunnen versturen naar de development. Ik vraag je alleen om taken uit te voeren, het verder opbouwen van een styleguide is aan jou.

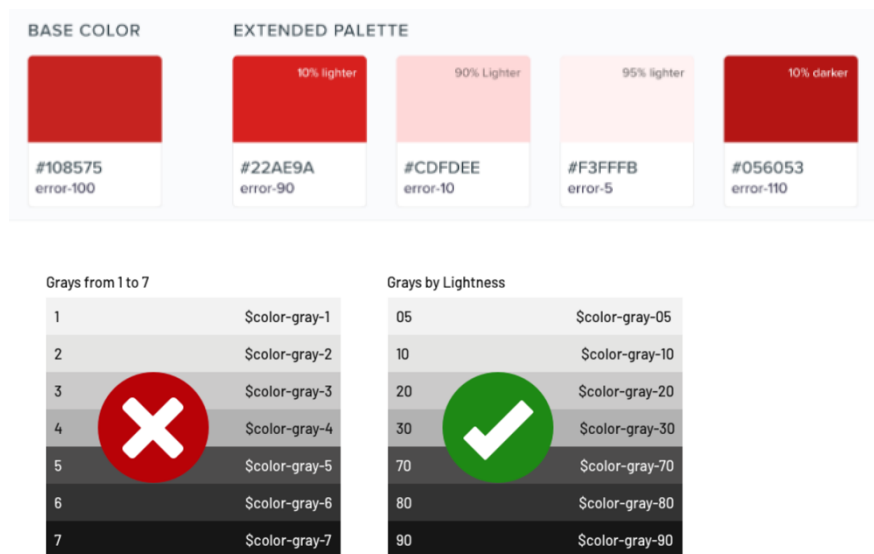
Na test worden wat vragen gesteld, die jezelf moet beantwoorden op basis wat je van de nieuwe richtlijnen vindt. Deze test wordt opgenomen, maar uiteindelijk alleen gebruikt voor schooldoeleinden.

Op de volgende bladzijde worden nog een 4 dingen uitgelegd. Deze zijn wellicht nieuw voor je, maar in de toekomst mee moeten worden genomen in de styleguides.

## Tinten

We gaan beginnen. Om te beginnen ga ik je kort wat uitleggen over tinten. Elke kleurenlabel representeert een basiskleur, terwijl de 'XX' nummers verschillende tinten representeren; zoals 'color label-XX'. Het wordt aangeraden om kleuren een naam te geven die gebaseerd zijn op hun functies en niet de tint, zie *Afbeelding 1 Voorbeeld kleurenlabel*. De schaal weerspiegelt een bekend bereik van donker naar licht, maakt het mogelijk om nieuwe opties ertussen te injecteren. De waarde van de 'XX' hangt af van de volgende drie aspecten:

- Basiskleur: 100
- Tinten donkerder dan de basiskleur: > 100
- Tinten lichter dan de basiskleur: < 100



Afbeelding 1 Voorbeeld kleurentabel

## Screaming Snake Case

De Screaming Snake is een schrijfstijl waarin elke spatie wordt vervangen door een onderstrepingsteken en alles wordt in hoofdletter geschreven. Een voorbeeld hierbij is: PRIMARY\_DEFAULT.

## Figma bestand

Er staat een Figma bestand geopend. Dit is een prototype dat ik heb gemaakt met de basiselementen die het development op deze manier wilt ontvangen. Kijk even rustig rond, want zo gaan we in een leeg project te werk. **Let wel op, alles wat je maakt moet juist gelabeld worden, dus ook alle layers! (Colors, Typography, Border-radius, Box-shadow).** Hierna laat ik je zien hoe de plug-in Figma Tokens werkt.

## Figma Tokens

Ik ga je laten zien hoe je een element toevoegt aan Figma Tokens

- Het toevoegen van een kleur
  - colors/PRIMARY\_DEFAULT
- Het toevoegen van een typografie
- Het toevoegen van een box-shadow
- Het toevoegen van een border-radius

## Aan de slag!

De styleguides moeten allemaal in een frame gemaakt worden. Zorg dat je de juiste onderdelen als component maakt.

### Colors

#### Stap 1

De eerste stap is een styleguide maken met 3 willekeurige kleuren. Het moet het volgende bevatten:

- **Geef ze de juiste categorie (colors/)**
- **Noem ze primary default, primary tap en primary select. Hiermee geef je ze meteen de juiste variabelen.**
- **Voeg deze styles hierna toe: (voorbeeld) colors/PRIMARY\_DEFAULT**

#### Stap 2

De tweede stap is een styleguide maken met 3 tinten (je mag ze zelf kiezen) en geef ze meteen de juiste variabelen:

- **Geef ze de juiste categorie (colors/)**
- **Noem ze neutral.90, neutral.100 en neutral.110. Hiermee geef je ze meteen de juiste variabelen.**
- **Voeg deze styles hierna toe: (voorbeeld) colors/NEUTRAL\_100**

### Typography

#### Stap 3

De derde stap is een styleguide maken met 2 fonts en geef ze de juiste variabelen:

- **Geef ze de juiste categorie (typography/)**
- **Noem ze heading.01 en heading.02. Hiermee geef je ze meteen de juiste variabelen.**
- **Voeg deze styles hierna toe: (voorbeeld) typography/HEADING\_01**

### Box-shadow

#### Stap 4

De vijfde stap is een styleguide maken met 1 box-shadow en geef deze de juiste variabelen.

- **Geef het de juiste categorie (boxshadow/)**
- **Noem het BUTTON. Hiermee geef je meteen de juiste variabele mee.**
- **Voeg deze styles hierna toe: (voorbeeld) boxshadow/BUTTON**

### Border-radius

#### Stap 5

De vierde stap is een styleguide maken met 1 border-radius en geef deze de juiste variabelen:

- **Geef het de juiste categorie (borderradius/)**
- **Noem het X-SMALL. Hiermee geef je meteen de juiste variabele mee.**
- **Voeg deze styles hierna toe: (voorbeeld) borderradius/X-SMALL**

### Figma-tokens

#### Stap 6

**Bij stap 6 ga je de nieuwe aangemaakte variabelen in de plug-in importeren. Dit doe je door:**

- Rechtermuisknop/Plug-in/Figma Tokens
- Styles/ Import Styles/ Import
- Doordat deze plug-in ook elementen meeneemt die development niet hoeft te hebben, moet je de volgende elementen uitklikken:
  - fontFamilies, lineHeights, fontWeights, fontSize, letterSpacing, paragraph, textCase en textDecoration.
- Hierna zouden ze naar GitHub gepusht kunnen worden, maar dat hoeft voor nu niet!

### **Casus vragen**

#### **Roy (Senior Designer)**

- Hoe bevallen de nieuwe richtlijnen je?

Ik heb al zoveel designsystems opgezet bij vorige bedrijven, dat ik bijvoorbeeld shadows typ in plaats van boxshadows. Het is een kleine moeite en het is prima om me daaraan aan te passen.

- Heb jij het gevoel dat je niet meer vrij kan zijn met het designen? En waarom?

Nee. De manieren hoe kleuren zijn opgezet, ik kan nog steeds tinten beslissen. Ik heb nog steeds best wel veel vrijheid met text styles. Als dit de makkelijkste manier voor development is is het prima, voel me niet belemmerd.

- Heb je nog op- of aanmerkingen?

Het enige wat me stoort zijn de hoofdletters, dus de screaming snake case. Het komt voor mij zo onvriendelijk over, maar het maakt voor me niet uit. Het ziet er zo schreeuwerig uit, verder prima.

- Ben jij secuur in het gebruiken van je SCRUM? Dus als je een gedeelte af hebt, plaatst je dat dan ook op het SCRUM bord zodat development daar al mee kan beginnen?

Ja wel redelijk, als ik in scrum werk wel. Ik werk aan verschillende projecten en daar wordt het niet altijd gebruikt. Als development dat wilt gebruiken dan wil ik dat secuur gebruiken.

- Werk jij met het designen mobile first? En waarom doe jij het op deze manier?

Per project verschillend, idealiter vragen we aan de klant waar de meeste gebruikers zitten. Momenteel is het vaker mobile, maar het niet altijd mobile first. Ligt aan de opdrachtgever.

- Zijn er nog dingen waarvan jij denkt dat dat de samenwerking tussen design en development kan verbeteren?

Touwtrek wedstrijd, wie heeft er gelijk. Begin van het project al vaststellen wat ieder wilt en dat je met zijn alle samenwerkt. Updates geven over tinten of naamgeving, geen discussie worden maar goed uitkomen.

#### **Rida (Junior designer)**

- Hoe bevallen de nieuwe richtlijnen je?

Niet heel veel verandering, alleen nu alles met hoofdletter gaan werken.

- Heb jij het gevoel dat je niet meer vrij kan zijn met het designen? En waarom?

Nee, het gaat alleen om de logische benamingen. Je kan nog steeds alles designen.

- Heb je nog op- of aanmerkingen?

Nee, vrij duidelijk.



- Ben jij secuur in het gebruiken van je SCRUM? Dus als je een gedeelte af hebt, plaatst je dat dan ook op het SCRUM bord zodat development daar al mee kan beginnen?  
Nee, ligt er hoe de deadline zit.

- Werk jij met het designen mobile first? En waarom doe jij het op deze manier?  
Ik werk mobile first. Het heeft bij ons prio omdat meeste telefoon hebben en desktop minder gebruikt wordt.

- Zijn er nog dingen waarvan jij denkt dat dat de samenwerking tussen design en development kan verbeteren?  
Meer met elkaar afstemmen. Per design sprints updaten en dan afspraken met development. Werken met sprints.

### **Dominique (Senior designer)**

- Hoe bevallen de nieuwe richtlijnen je?  
Prima, alleen de mapjes is even wennen. Dat het aparte mapjes voor design moet gaan worden.

- Heb jij het gevoel dat je niet meer vrij kan zijn met het designen? En waarom?  
Met het designen voel ik me nog steeds vrij. Qua snelheid kan het wel als we die extra mapjes van PRIMARY/DEFAULT niet hebben, maar dat zal een kwestie van wennen zijn.

- Heb je nog op- of aanmerkingen?  
Naast de mapjes niks. Het voelt dubbel om onder colors te zetten. Het is geen ramp, dat is best oke om mee te werken. Iets minder optimaal in Figma, maar daarom kunnen we het nog wel doen.

- Ben jij secuur in het gebruiken van je SCRUM? Dus als je een gedeelte af hebt, plaatst je dat dan ook op het SCRUM bord zodat development daar al mee kan beginnen?  
Ligt heel erg aan het project, ligt eraan of we met scrum werken. Als die vraag er is vanuit development, maar in de realiteit wordt het scrumbord vergeten of niet heel secuur mee gewerkt.

- Werk jij met het designen mobile first? En waarom doe jij het op deze manier?  
Ligt ook weer aan het project, als ik een website maak, meestal wel. Ik moet altijd kunnen schalen naar mobiel, maar het ligt er aan wat het project wilt.

- Zijn er nog dingen waarvan jij denkt dat dat de samenwerking tussen design en development kan verbeteren?  
Development en design werken langs elkaar heen. Beter ingerichte afspraken, zodat ze beter op elkaar aansluiten met bijvoorbeeld 2 weken er tussen dan 2 maanden. Het gebruik hiervan gaat al onwijs helpen. Als dit werkt ook doorgeven aan Resoluut, die wilt het ook doorvoeren.

### **Tim (Junior designer)**

- Hoe bevallen de nieuwe richtlijnen je?

Ik denk dat het handig is dat het wel uitgelegd wordt aan de designers, dus samen door te lopen door dit stappenplan met deze test lukt het makkelijk.

- Heb jij het gevoel dat je niet meer vrij kan zijn met het designen? En waarom?

Ik denk dat ik dit pas doen als ik mijn wireframes heb staan. Dit zou ik niet vanaf het begin willen doen. Maar dit kan afgesproken worden in sprints, dan wordt er rekening mee gehouden worden in de sprints. Als dit een afspraak kan worden en ruimte vrijgemaakt voor kan worden prima.

- Heb je nog op- of aanmerkingen?

Fijn dat je een voorbeeld ofwel een prototype hebt. Door voorbeeld is het visueel en fijn. Ik wist niks van de Figma Tokens, dat snap ik nu door je uitleg en stappenplan. Als iemand het je kan uitleggen is het top.

- Ben jij secuur in het gebruiken van je SCRUM? Dus als je een gedeelte af hebt, plaatst je dat dan ook op het SCRUM bord zodat development daar al mee kan beginnen?

Ligt heel erg aan het project. Soms wordt het meer als checklist gezien.

- Werk jij met het designen mobile first? En waarom doe jij het op deze manier?

Bijna altijd mobile first, als je niet eerst mobile first werkt is het moeilijk om het naar makkelijke schermen om te zetten.

- Zijn er nog dingen waarvan jij denkt dat dat de samenwerking tussen design en development kan verbeteren?

Je maakt een design, dat development wilt je altijd checken. Ruimte vrijgemaakt worden om te sparren beter kan gemaakt worden. Development wilt dat nu voorkomen. Goede communicatie of duidelijke afspraken om samen in te checken om designs te bespreken, en niet als politie maar om voor het beste resultaat te komen. In gecalculeerd worden in een project, om ook uit het oog van de klant te krijgen.

### **Robin (Medium/junior designer)**

- Hoe bevallen de nieuwe richtlijnen je?

Het is even wennen, veel regeltjes. We zijn momenteel erg aan het freestylen, vandaar dat het ook misgaat. Het is meer technisch. Het mag nog steeds kanariegeel heten enzo. Er wordt nu ook een universeel template gemaakt worden vanuit design. Designers zijn minder precies als developers.

- Heb jij het gevoel dat je niet meer vrij kan zijn met het designen? En waarom?

Ik denk dat ik dit pas doen als ik mijn wireframes heb staan. Dit zou ik niet vanaf het begin willen doen. Maar dit kan afgesproken worden in sprints, dan wordt er rekening mee gehouden worden in de sprints. Als dit een afspraak kan worden en ruimte vrijgemaakt voor kan worden prima.

- Heb je nog op- of aanmerkingen?

Geen op- of aanmerkingen.

- Ben jij secuur in het gebruiken van je SCRUM? Dus als je een gedeelte af hebt, plaatst je dat dan ook op het SCRUM bord zodat development daar al mee kan beginnen?

De meeste projecten die ik doe zit development niet bij de scrum. Ik heb het weleens gedaan. Met sprints gaat dit wel werken.

- Werk jij met het designen mobile first? En waarom doe jij het op deze manier?  
Hangt af van het project. We toetsen waar mensen het platform het meest opgebruikt. Als het onbeslist is of half half is doen we wel mobile first want dat is moeilijker.
- Zijn er nog dingen waarvan jij denkt dat dat de samenwerking tussen design en development kan verbeteren?  
Als het project af is laten de designers de developers in de steek. Dus als dit meer begeleidend gebeurt, zal het wel beter gaan. Meer support vanuit UX kant.

## Bijlage 7 Resultaten usability test designers

### Video opnames

Rida: <https://youtu.be/JGyhh-hWRhE>

Dominique: <https://youtu.be/Zl-2ca6NElQ>

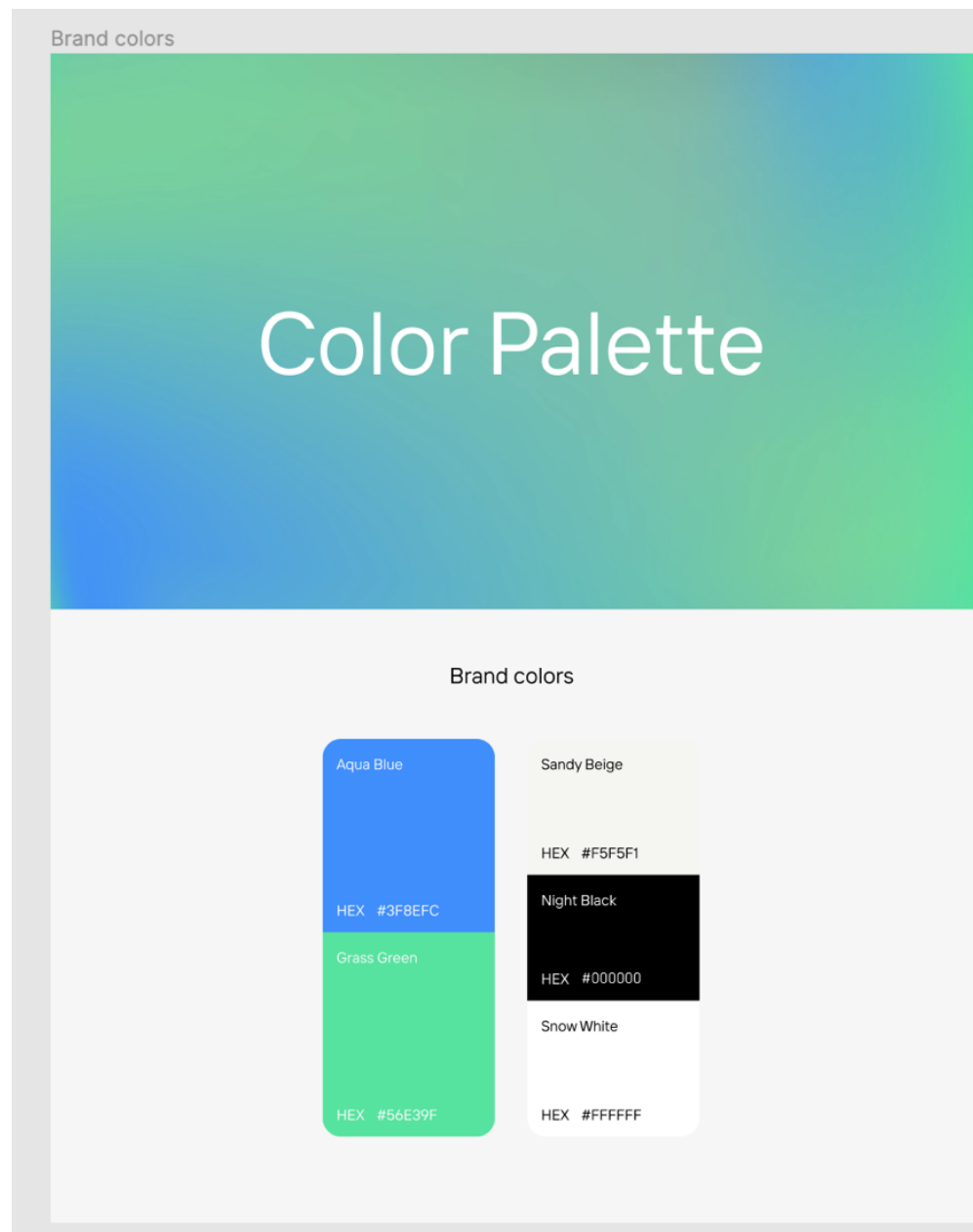
Roy: [https://youtu.be/0xg\\_U0EZVfY](https://youtu.be/0xg_U0EZVfY)

Tim: <https://youtu.be/kBFsArt4Yeo>

Robin: <https://youtu.be/otlQJveOuUg>

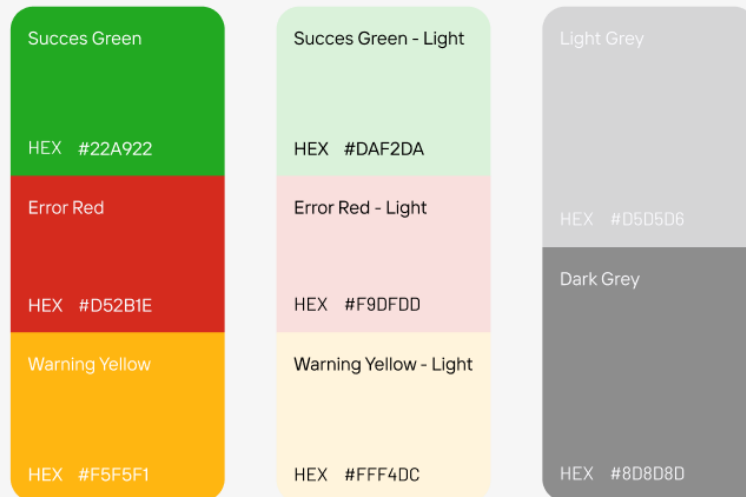
## Bijlage 8 Designsysteem CerQlar

GitHub link: <https://github.com/RobinvanVliet/cerqlar.git>



## Functional colors

### Functional colors



# Typography

## Font

TT Interfaces

Neue Haas Grotesk

## Font weight

Regular

**Demi Bold**

Light

Roman

**Medium**

**Bold**

## XL/L

H, Size, Lineheight

Heading 01

Aa

56px

110%

Heading 02

Aa

40px

120%

Heading 03

Aa

32px

130%

Heading 04

Aa

24px

130%

Heading 05

Aa

18px

130%

Heading 06

Aa

16px

160%

## M/S

H, Size, Lineheight

Heading 01

Aa

40px

120%

Heading 02

Aa

24px

130%

Heading 03

Aa

18px

130%

Heading 04

Aa

16px

130%

## Overaching

H, Size, Lineheight

Body

Aa

16px

160%

Subtext

Aa

12px

160%



Heading 04

Aa

24px

130%

Heading 05

Aa

18px

130%

Heading 06

Aa

16px

160%

## M/S

H, Size, Lineheight

Heading 01

Aa

40px

120%

Heading 02

Aa

24px

130%

Heading 03

Aa

18px

130%

Heading 04

Aa

16px

130%

## Overaching

H, Size, Lineheight

Body

Aa

16px

160%

Subtext

Aa

12px

160%

Logos

# Logos

Logos

**CERQLAR**

CERQLAR



## Icons

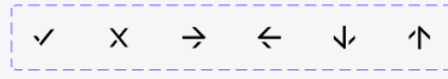
### Icons

Functional

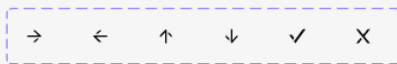
32px



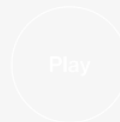
24px




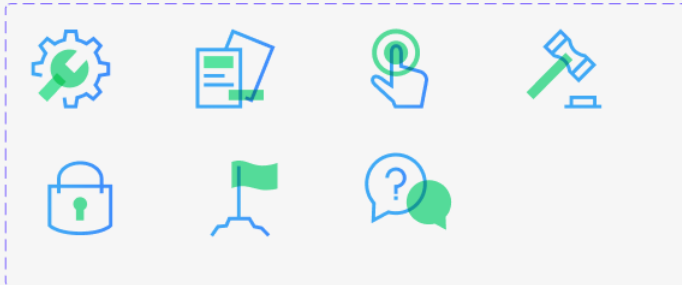
16px
















Video



 Icons Rich



▼ typography
Ag XL_L_HEADING_01 · 56/110
Ag XL_L_HEADING_02 · 40/120
Ag XL_L_HEADING_03 · 32/130
Ag XL_L_HEADING_04 · 24/130
Ag XL_L_HEADING_06 · 16/160
Ag XL_L_HEADING_05 · 18/130
Ag M_S_HEADING_01 · 40/120
Ag M_S_HEADING_02 · 24/130
Ag M_S_HEADING_03 · 18/130
Ag M_S_HEADING_04 · 16/130
Ag OVERACHING_BODY · 16/Auto
Ag OVERACHING_SUBTE... · 12/160
Ag OVERACHING_BUTT... · 18/160

Design	Prototype	Inspect
Ag OVERACHING_EXPL...		· 160/130
Color Styles		
▼ colors		
	AQUA_BLUE	
	GRASS_GREEN	
	SANDY_BEIGE	
	NIGHT_BLACK	
	SNOW_WHITE	
	SUCCESS_GREEN	
	ERROR_RED	
	WARNING_YELLOW	
	SUCCESS_GREEN_LIGHT	
	ERROR_RED_LIGHT	
	WARNING_YELLOW_LIGHT	
	LIGHT_GREY	
	DARK_GREY	

# *TOUCHTRIBE*

# *ADVIESRAPPORT*

Het toepassen van het automatiseren van designtokens

*30-05-2022*

## Managementsamenvatting

Dit advies is geschreven voor het oplossen van het probleem waar Touchtribe tegenaan loopt; Touchtribe loopt er tegenaan dat ontwerpers en developers vaak onenigheid hebben over het opgeleverde werk. Het opgeleverde design is niet altijd even goed te maken voor een developer. Een ander ergerpunt van de developers is dat zij handmatig designtokens uit het ontwerp moeten halen, meer hierover wordt besproken in Hoofdstuk 4 Praktisch onderzoek.

De hoofdvraag van het onderzoek luidt als volgt: *“Wat zijn de wensen en behoeften van de doelgroep de ontwerpers en developers van Touchtribe m.b.t. het ontwikkelen van een designsysteem?”* Dit adviesrapport is geschreven in opdracht van Touchtribe die het een designsysteem opgezet willen krijgen om consistentie in het werkveld te creëren en developer experience wil verbeteren. Hiermee wordt de werksfeer positief beïnvloed. Er is onderzoek gedaan in het literatuuronderzoek en experimenteel onderzoek om advies aan Touchtribe te kunnen geven.

Uit het onderzoek is naar voren gekomen dat er gebruik gemaakt moet worden van Figma in plaats van Storybook, doordat Figma bruikbare plug-ins heeft om het probleem van Touchtribe op te lossen. Daarnaast zijn er twee alternatieven naar voren gekomen. Alternatief 1 is het gebruik maken van de plug-in Figma Tokens voor het automatiseren van de tokenflow. Alternatief 2 is het gebruik maken van de plug-in Design Tokens voor het automatiseren van de tokenflow.

Het advies is om te kiezen voor alternatief 1 Figma Tokens. Figma Tokens is een tool in Figma die van een styleguide JSON-variabelen maakt. Deze plug-in is enkele designers al gebruikt bij andere projecten die niet in samenwerking waren met Touchtribe. Sommige designers zijn er dus al mee bekend. Door de plug-in Figma Tokens te gebruiken kunnen zoveel mogelijk wensen en behoeften vanuit het bedrijf worden gewaarborgd. Dit lost een groot deel van het probleem op, doordat er te maken is met verschillende teams; designers en developers. Het gebruiken van deze plug-in zou 2 weken nodig hebben om aan de designers uit te leggen en er mee kennis te leren maken. Dit komt door de drukke schema's van de designers en het leren werken met de plug-in. Er is voor dit alternatief geen budget nodig, doordat het door een stagiair wordt ontwikkeld.

## Inhoudsopgave

<b><u>MANAGEMENTSAMENVATTING.....</u></b>	<b><u>101</u></b>
<b><u>INLEIDING .....</u></b>	<b><u>103</u></b>
<b><u>ONDERZOEK.....</u></b>	<b><u>104</u></b>
<b><u>ALTERNATIEVEN.....</u></b>	<b><u>106</u></b>
<b><u>ALTERNATIEF 1 .....</u></b>	<b><u>106</u></b>
<b><u>ALTERNATIEF 2 .....</u></b>	<b><u>106</u></b>
<b><u>CONCLUSIE.....</u></b>	<b><u>107</u></b>
<b><u>BIBLIOGRAPHY.....</u></b>	<b><u>ERROR! BOOKMARK NOT DEFINED.</u></b>

## Inleiding

Touchtribe is een web-development bedrijf dat e-commerce oplossingen bouwt die als doel hebben merken te laten groeien. Touchtribe loopt er tegenaan dat ontwerpers en developers vaak onenigheid hebben over het opgeleverde werk. Het opgeleverde design is niet altijd even goed te maken voor een developer. Een voorbeeld hierbij is dat developers een ontwerp krijgen waar niet alle variabelen een naam hebben of afstanden niet kloppen.

Een ander ergerpunt van de developers is dat zij handmatig designtokens uit het ontwerp moeten halen, meer hierover wordt besproken in Hoofdstuk 4 Praktisch onderzoek. Een designtoken zijn bijvoorbeeld de kleuren en typografie. Deze onenigheden kosten tijd en geld. Er wordt verder voor de alternatieven een rekening gehouden te worden met budget, doordat dit probleem opgelost moet worden door een stagiair.



## Onderzoek

Uit hoofdstuk 3 Tools is naar voren gekomen dat er een designsysteem ontwikkelt kan worden in Storybook, Figma of een bestaande tool. Storybook is een open source tool voor het bouwen van UI componenten en pagina's. Het stroomlijnt de ontwikkeling, het testen en de documentatie van de gebruikersinterface (Storybook, 2022). Storybook kan gebruikt worden voor React, Vue, Angular, Web components, Ember, HTML, Svetle en Preact. Storybook kan gebruikt worden om de een makkelijk overzicht te krijgen van de gemaakte componenten in Figma. Figma is een cloudgebaseerde tool voor het ontwerpen en prototypen van de UI waarmee meerdere teams kunnen samenwerken bij het maken, testen en implementeren van interface- of productontwerpen. Functies omvatten versiegeschiedenis, projectmanagement, opmerkingen, gebruikersgebaseerde machtigingen, het maken van plug-ins en eenmalige aanmelding (Figma, 2022).

Storybook kan eenvoudig gekoppeld worden aan Figma, maar de componenten die je hier kan zien zijn ook in Figma zelf te zien. In Storybook kan er geen gebruik gemaakt worden van plug-ins. Wel kan je een enkel component zien zoals een button, in plaats van alle buttons zoals in Figma. De CSS wordt automatisch in Storybook gezet. Daarnaast update Storybook automatisch het Figma bestand bij, bij aanpassingen. Enkele struikelpunten zijn dat Storybook erg traag wordt als er veel componenten voor een heel designsysteem ingeladen worden. Storybook kan geen gebruik maken van plug-ins, dus de eisen vanuit Touchtribe kunnen niet allemaal nageleefd worden. Momenteel werken de desginers al een lange tijd in Figma. In Figma heb je heel veel nuttige plug-ins waar Touchtribe gebruik van kan gaan maken om het designsysteem zo optimaal mogelijk te laten werken. Een van de wensen van de organisatie is het gebruik kunnen maken van JSON-tokens vanuit het designsysteem die naar Github verstuurd kunnen worden. Daarnaast willen ze duidelijke richtlijnen en benamingen vanuit het Figma bestand. Als de voor- en nadelen aan elkaar zijn afgewogen komt naar voren dat een designsysteem in Figma de beste manier is. Dit komt doordat de wensen van Touchtribe hier het meest in worden vervuld.

Dan heb je in Figma twee opties om designtokens te versturen; Design Tokens en Figma Tokens. In Hoofdstuk 3 Tools is hier verder over te lezen. De plug-in Design Tokens kan de tokens in een .zip bestand opleveren die gedownload kan worden op elk device. Ook kunnen deze tokens naar GitHub gepusht worden. Enkele nadelen hierbij zijn dat het door een enkele developer is gebouwd en er daardoor nog bugs inzitten. Af en toe worden de tokens niet goed verzonden als deze onjuist gelabeld zijn. Als de developer ermee stopt, houdt de plug-in op. Daarnaast heb je aparte bestanden voor het gebruik van de plug-in. Je moet zelf de package.json van de Design-Token-Transformer van GitHub overnemen in je eigen package.json bestand.

De plug-in Figma Tokens wordt vertegenwoordigd vanuit Figma en heeft hierdoor amper bugs. Ook deze tokens kunnen naar GitHub gestuurd worden. Bij deze plug-in kan je een eigenaam meegeven aan de variabelen. Dit is fijner voor de designers werken. Zij hoeven daar geen rekening mee te houden. Het nadeel bij Figma Tokens is dat als de developers stoppen, de plug-in ook stopt.

De nieuwe richtlijnen van het designsysteem zijn in Hoofdstuk X 5.1 Designsysteem te lezen. Doordat Touchtribe voor verschillende projecten werkt, moet het designsysteem per project opnieuw worden opgesteld. De richtlijnen en plug-ins helpen de designers dit snel en gemakkelijk te kunnen uitvoeren.

De alternatieven worden op het eind door het FOETSJE-model gehaald. Het FOETSJE-model toetst de potentiële marketingstrategie op de haalbaarheid van 7 doelstellingen (Knoot, 2020). De 7 doelstellingen zijn als volgt:

- **F:** Financieel – Genoeg middelen beschikbaar om het plan uit te voeren?
- **O:** Organisatorisch – Is het plan organisatorisch uitvoerbaar?
- **E:** Economisch – Past het plan binnen de doelstellingen van de organisatie?
- **T:** Technologisch – Is het plan technisch uitvoerbaar?
- **S:** Sociaal – Past het plan binnen de sociale doelstellingen van de organisatie?
- **J:** Juridisch – Is het plan juridisch uitvoerbaar?
- **E:** Ethisch – Is het plan ecologisch verantwoord?

## Alternatieven

### Alternatief 1

De eerst mogelijke oplossing voor het probleem is het toepassen van de plug-in Figma Tokens. Figma Tokens is door enkele designers al gebruikt bij andere projecten die niet in samenwerking waren met Touchtribe. Sommige designers zijn er dus al mee bekend. Door de plug-in Figma Tokens te gebruiken kunnen zoveel mogelijk wensen en behoeften vanuit het bedrijf worden gewaarborgd. Dit lost een groot deel van het probleem op, doordat er te maken is met verschillende teams; designers en developers. Het gebruiken van deze plug-in zou 2 weken nodig hebben om aan de designers uit te leggen en er mee kennis te leren maken. Dit komt door de drukke schema's van de designers en het leren werken met de plug-in.

#### Benodigheden

De benodigheden die nodig zijn om alternatief 1 uit te voeren is de ontwerpomgeving Figma, communicatie tussen beide teams en de plug-in Figma Tokens in Figma.

#### Voordelen

- Plug-in wordt vertegenwoordigd vanuit Figma; waterdicht systeem
- Kan naar GitHub gestuurd worden
- Eenvoudiger; kan je eigen benamingen meegeven

#### Nadelen

2. Als developers stoppen, houdt de plug-in op

### Alternatief 2

Het tweede alternatief is het automatiseren van tokens door middel van de plug-in Design Tokens. Design Tokens is een plug-in net als Figma Tokens die designtokens door kan sturen naar GitHub. Met deze plug-in hebben designers nog niet gewerkt. De uiteindelijke wensen en behoeften kunnen met deze plug-in ook gewaarborgd worden, maar is de manier van gebruik minder prettig. Deze plug-in heeft ongeveer 4 weken nodig om uitgelegd te worden aan de designers, doordat deze complexer in elkaar zit in tegenstelling tot alternatief 1.

#### Benodigheden

De benodigheden die nodig zijn om alternatief 2 uit te voeren is de ontwerpomgeving Figma, communicatie tussen beide teams en de plug-in Design Tokens in Figma.

#### Voordelen

1. Design Tokens kan de tokens in een .zip opleveren die gedownload kan worden op een device
2. Kan naar GitHub gestuurd worden

#### Nadelen

- Is door een enkele developer gebouwd; zitten nog bugs in
- Verzend af en toe de tokens niet; als ze niet juist gelabeld zijn
- Als developer stopt, houdt de plug-in op
- Aparte bestanden voor gebruik; voor gebruik moet je bijvoorbeeld eerst de package-json van de Design-Tokens-Transformer overnemen in je eigen package-json bestand.

## Conclusie

Op basis van de onderzoeksresultaten en een zorgvuldige afweging van de voor- en nadelen van de beschreven alternatieven is het advies om alternatief 1 te kiezen. Alternatief 1 kwam uit het FOETSJE-model het beste naar voren:

- **F: Financieel** – Er zijn genoeg middelen beschikbaar om het plan uit te voeren doordat het door een stagiair ontwikkeld wordt.
- **O: Organisatorisch** – Het plan is organisatorisch uitvoerbaar, doordat de designers en developers alleen tijd hoeven vrij te maken voor het testen met de Figma Tokens.
- **E: Economisch** – Het plan past binnen de doelstellingen van de organisatie, doordat het in de wensen en behoeften van de organisatie valt.
- **T: Technologisch** – Het plan is technisch uitvoerbaar, doordat er voor het alternatief Figma en de plug-in nodig is.
- **S: Sociaal** – Het plan past binnen de doelstellingen van de organisatie, doordat het in de wensen en behoeften van de organisatie valt.
- **J: Juridisch** – Het plan is juridisch uitvoerbaar, doordat er bij het toepassen van dit plan geen wetten en regels worden overschreven.
- **E: Ethisch** – Het plan is ecologisch verantwoord, doordat het geen schade aan het milieu, dieren of mensen.

Wanneer dit advies wordt toegepast, kan het probleem tussen de designers en developers worden opgelost door de volgende redenen uit Hoofdstuk 3.5 Wensen en behoeften:

- **Gedefinieerde basics;** de basics kunnen simpelweg worden gedefinieerd door het nog steeds gebruik maken van Figma. In Figma Tokens is er de mogelijkheid de variabelen een eigenaam te geven. In Design Tokens moet er aan de bepaalde namen worden voldaan, anders kunnen de tokens niet getransformeerd worden.
- **Variabelen gekoppeld;** De variabelen kunnen door het gebruik van Figma consistent worden gekoppeld en toegepast.
- **Geautomatiseerde tokenflow;** Het designsysteem moet automatisch geëxporteerd en gekoppeld worden met de develop omgeving, ofwel het moet verstuurd kunnen worden naar GitHub. De JSON-tokens hoeven hierdoor niet handmatig uit Figma te worden gehaald. Daarnaast worden de tokens door het script “Token-Transformer” die ontwikkelt is voor Touchtribe meteen gebruikt worden tijdens het coderen.
- **Vrijheid;** de designers hebben aangegeven hun vrijheid te willen behouden bij het werken met een designsysteem. Door Figma Tokens kunnen zij bijvoorbeeld nog steeds hun eigenamen zoals “PIMPEL\_PAARS” geven aan een parse kleur.

Om het advies, oftewel alternatief 1 te implementeren, is het ontwikkelde script de “Token-Transformer” nodig. Hier is meer over te lezen in Hoofdstuk 4.4 Figma Tokens; de Token-Transformer.

## Bibliography

- Soda Studio. (2021, mei 25). *Design systems voorbij design*. From Sodastudio: <https://www.sodastudio.nl/2021/05/25/design-systems-voorbij-design/>
- Woznica, D. (2018, augustus 1). *Atomic Design and ReactJS*. From Danilowoz: <https://danilowoz.com/blog/atomic-design-with-react>
- Oostenrijk, V. (2022). *Atomic design: wat is het en hoe kan het bijdragen aan een goed webdesign?* From Whello: <https://whello.nl/marketing-tips/web-design/atomic-design-wat-het-en-hoe-kan-het-bijdragen-aan-een-goed-webdesign/>
- Alli, E. (2020, juni 3). *10 Best Design Systems and How to Learn (and Steal) From Them*. From Designerup: <https://designerup.co/blog/10-best-design-systems-and-how-to-learn-and-steal-from-them/>
- Fessenden, T. (2021, april 11). *Design Systems 101*. From Nn Group: <https://www.nngroup.com/articles/design-systems-101/>
- Wadsworth, T. (2019, oktober 7). *Building a Design System: Speed, Scale, Collaboration and Innovation*. From XD Adobe: <https://xd.adobe.com/ideas/principles/design-systems/building-design-system-speed-scale-collaboration-innovation/>
- Kuipers, D. (2022). *Boilerplate*. From Online Marketing Agency: <https://onlinemarketingagency.nl/marketingtermen/boilerplate/>
- Figma. (2022). *Figma Tokens*. From Figma: <https://docs.tokens.studio>
- Orace. (2022). *Content Blocks*. From Docs Oracle: <https://docs.oracle.com/en/cloud/saas/marketing/eloqua-user/Help/ContentBlocks/ContentBlocksOverview.htm#:~:text=Content%20blocks%20are%20groups%20of,%2C%20shared%20content%2C%20and%20more.>
- Responsive Design. (n.d.). *Defining Breakpoints*. From Responsive Design : <https://responsivedesign.is/strategy/page-layout/defining-breakpoints/>
- Swierts, S. (2020, december 10). *The best tools for documenting design systems*. From Hike: <https://hike.one/update/the-best-tools-for-documenting-design-systems>
- Storybook. (2022). *Build component driven UIs faster*. From Storybook: <https://storybook.js.org/>
- Figma. (2022). *Storybook and Figma*. From Figma: <https://help.figma.com/hc/en-us/articles/360045003494-Storybook-and-Figma>
- New Line. (n.d.). *Connecting Figma with Storybook*. From Newline: <https://www.newline.co/courses/storybook-for-react-apps/connecting-figma-with-storybook>
- Figma. (2022). *Figma*. From Figma: [www.figma.com](https://www.figma.com)
- Hacq, A. (2018, mei 22). *Everything you need to know about Design Systems*. From UX Design: <https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969>
- Cunningham, K. (2022). *What is a UX Design System?* From Fuzzymath: <https://fuzzymath.com/blog/what-is-ux-design-system/>
- Design Systems. (n.d.). *Typography Guides*. From Design Systems: <https://www.designsystems.com/typography-guides/>
- Choudhary, A. (2021, januari 30). *Defining colors in your design system*. From UX Design: <https://uxdesign.cc/defining-colors-in-your-design-system-828148e6210a>
- Curtis, N. (2016, april 13). *Color in Design Systems*. From Medium: <https://medium.com/eightshapes-llc/color-in-design-systems-a1c80f65fa3>
- W3. (2016). *Understanding WCAG 2.0*. From W3: <https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html>

- Fission. (2021). *Screaming Snake Case* . From Fission codes:  
<https://fission.codes/blog/screaming-snake-case/>
- Amzn GitHub. (n.d.). *Style Dictionary*. From Amzn GitHub: <https://amzn.github.io/style-dictionary/#/README>
- Npm. (2022). *Token-transformer v0.0.19*. From Npm: <https://npm.io/package/token-transformer>
- Speek, M. K. (2012, september 27). *Wat is een Content Management Systeem (CMS)?* From Max: <https://www.max.nl/blog/content-management-systeem/>
- Touchtribe. (2022). *Touchtribe: Beyond e-commerce*. From Touchtribe:  
<https://www.touchtribe.nl/>
- Ward, C. (2020, Oktober 6). *An Introduction To Developer Experience (DevEx, DX)*. From Humanitec: <https://humanitec.com/blog/developer-experience>
- Knoot, J. (2020). *FOETSJE-MODEL*. From Strategisch marketingplan:  
<https://www.strategischmarketingplan.com/marketingmodellen/foetsje-model/>
- Benders, L. (2021, oktober 26). *Scribbr*. From Reflecteren met de STARR-methode |  
 Stappenplan + voorbeelden: <https://www.scribbr.nl/stage/starr-methode/#:~:text=De%20meestgebruikte%20methode%20om%20te,hel%20resultaat%20en%20de%20reflectie.>
- Benders, L. (2019, november 18). *Reflecteren met het Model van Korthagen*. From Scribbr:  
<https://www.scribbr.nl/stage/model-van-korthagen/>