



Faculteit Bedrijf en Organisatie

Een gebruiker wegwijs maken doorheen een applicatie van aanzienlijke omvang

Jakob Lierman

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Karine Samyn
Co-promotor:
Bert Maurau

Instelling: Cardify

Academiejaar: 2019-2020

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Een gebruiker wegwijs maken doorheen een applicatie van aanzienlijke omvang

Jakob Lierman

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Karine Samyn
Co-promotor:
Bert Maurau

Instelling: Cardify

Academiejaar: 2019-2020

Tweede examenperiode

Woord vooraf

Deze bachelorproef werd geschreven in het kader van het behalen van het diploma “Bachelor in de Toegepaste Informatica”, afstudeerrichting Mobile Apps.

Mijn interesses gaan verder dan enkel de technische kant van mijn opleiding. Net daarom heb ik voor dit onderwerp gekozen. Het verkennen van verschillende UX en UI-elementen sprak me enorm aan. Het was een leerrijke ervaring om applicaties en IT voor te leggen aan iedereen in mijn omgeving, jong en oud, om dan van dichtbij te observeren hoe men met de technologie omspringt.

Deze scriptie zou niet tot stand zijn gekomen zonder enkele personen. Hierbij wil ik van dit voorwoord graag gebruik maken om deze personen te bedanken.

Zonder Bert Maurau, mijn co-promotor, had ik deze scriptie niet kunnen aanvullen met gegevens en voorbeelden vanuit Cardify. Hij en Seppe Vereecken stonden steeds klaar om mijn vragen te beantwoorden en deze scriptie aan te vullen met nuttige info en voorbeelden.

Deze scriptie werd geschreven tijdens de Covid-19-crisis. Bram Van de Velde heeft tools aangeboden om mijn proef zo vlot mogelijk te laten verlopen vanop afstand. Zo kon ik de proof-of-concept applicatie delen met iPhone-gebruikers zonder hiervoor met de participant in contact te komen.

Quentin Braet gaf deze scriptie een extra dimensie door zijn kennis van bij In The Pocket te delen. Door deze inzichten op te nemen in de tekst is deze scriptie een volwaardig schrijven waar naartoe kan verwezen worden bij de implementatie van of onderzoek naar learnability in software.

Karine Samyn, mijn promotor, volgde mijn voortgang nauw op en gaf op regelmatige basis feedback. Door deze opbouwende kritiek werd deze scriptie zonder veel problemen tot een goed einde gebracht.

Amber Priem hielp me doorheen de statistische analyses, waardoor ik mijn gemeten waardes kon omzetten in een conclusie. Ook was ze steeds de eerste die klaar stond om een tekstblok na te lezen of te controleren op grammaticale fouten.

Ten slotte had ik graag nog mijn ouders bedankt voor de financiële steun om deze opleiding tot een goed einde te brengen. Uiteraard had ik hen en iedereen die mijn scriptie even doornam graag nogmaals bedankt om deze tekst na te lezen, ook al ligt deze uit hun interessegebied.

Ik wens u veel leesplezier toe.

Jakob Lierman

Gent, 28 mei, 2020

Samenvatting

Elke gebruiker moet leren werken met de applicaties die hij of zij op zijn of haar toestel installeert. Men kan deze gebruiker helpen door middel van implementatie van learnability-elementen doorheen de software. De gebruiker kan op weg gezet worden door middel van een onboarding wanneer hij of zij de applicatie voor de eerste maal openst.

De grote meerderheid aan bedrijven en ontwikkelaars hebben echter de tijd of middelen niet om deze technieken zo geoptimaliseerd mogelijk uit te werken. Zo zijn deze vaak druk bezet met het halen van strakke deadlines om nieuwe functionaliteiten op punt te hebben dat de gebruikservaring vaak wat achterwege gelaten wordt.

Deze scriptie geeft duidelijkheid over de verschillende soorten onboarding en help-elementen en hoe deze best te gebruiken om een positief effect te hebben bij de eindgebruiker. Er wordt ook besproken hoe men best de implementatie van deze technieken test aan de hand van usability testing. Dit schrijven werd aangevuld met inzichten van ontwikkelaars.

Om het effect van learnability-elementen te meten werd er een usability test uitgevoerd met enkele participanten. Uit dit onderzoek resulteert dat het gebruik van deze elementen wel degelijk voordelig is, als en slechts als deze op de juiste manier verwerkt zijn doorheen de software. Elke software verschilt en voor elke software moet opnieuw bekeken worden waar men best hulp kan voorzien. Dit gaat vlot aan de hand van usability tests doorheen fasen van de ontwikkeling.

De relatie tussen (het gebrek aan) in-app user training en de gebruiksduur en/of levensduur van de applicatie wordt aanbevolen als toekomstig onderzoek. Hierbij kan men deze proef als startpunt gebruiken met een gerichte applicatie en een steekproef waarbij elke participant interesse toont in de functionaliteiten van de applicatie.

Inhoudsopgave

1	Inleiding	19
1.1	Probleemstelling	20
1.2	Onderzoeksvraag	20
1.2.1	Hoofdonderzoeksvraag	20
1.2.2	Deelonderzoeksvragen	20
1.3	Onderzoeksdoelstelling	21
1.4	Opzet van deze bachelorproef	21
2	Stand van zaken	23
2.1	User experience in software	23
2.2	Belangrijke factoren bij user experience	24
2.2.1	User Experience design in de praktijk	27

2.3 Usability testing	30
2.3.1 Laboratory en field testing	31
2.3.2 Andere testmethoden	32
2.3.3 Testpersonen	33
2.3.4 De System Usability Scale	35
2.4 Learnability	35
2.5 Onboarding	36
2.5.1 Starten met de implementatie van onboarding	37
2.5.2 Verschillende types van onboarding	39
2.5.3 Verschillende technieken en implementaties van onboarding	40
2.5.4 Testen van onboarding	44
2.6 In-app training	45
2.7 De verschillen tussen UX en UI	46
2.7.1 Software voor UI design	47
3 Methodologie	49
3.1 Het experiment	49
3.1.1 De populatie en steekproef	50
3.1.2 Gemeten variabelen	51
3.2 De proof-of-concept applicatie	51
3.2.1 De learnability elementen	53
3.2.2 Ontwikkeling	53
3.3 De test afnemen	55

4	Over de metingen	57
4.1	De gebruikstijd bij bepaalde opdrachten	57
4.2	Het vragen om hulp	58
4.3	Gebruik van de voorziene functionaliteiten	58
4.4	De SUS-score	59
4.5	Voorkeur voor onboarding en help-elementen	59
4.6	Voorkeur voor help-sectie	60
4.7	De gebruiksduur en/of levensduur van de applicatie	60
5	Resultaten van de proef	65
5.1	Onderzoeksraag 1	65
5.2	Onderzoeksraag 2	67
5.3	Onderzoeksraag 3	67
5.4	Onderzoeksraag 4	68
6	Conclusie	71
A	Onderzoeksvoorstel	73
A.1	Introductie	73
A.2	Literatuurstudie	74
A.2.1	Wat is 'onboarding'?	74
A.2.2	Voorgaand onderzoek	74
A.2.3	UX testing	74
A.3	Methodologie	75
A.4	Verwachte resultaten	75

A.5	Verwachte conclusies	76
B	Interview In The Pocket	77
B.1	Interview informatie	77
B.2	Vraag 1	77
B.3	Vraag 2	78
B.4	Vraag 3	79
B.5	Vraag 4	79
B.6	Vraag 5	80
C	Interview Cardify	81
C.1	Interview informatie	81
C.2	Vraag 1	81
C.3	Vraag 2	82
C.4	Vraag 3	82
C.5	Vraag 4	82
C.6	Vraag 5	83
D	System Usability Scale template	85
E	Deelnameformulier	87
E.1	Inleiding	87
E.2	Mobiel toestel	88
E.3	Over jezelf	88
E.4	Afsluiting	89

F	Berekeningen R onderzoeksvraag 1	91
G	Berekeningen R onderzoeksvraag 2	95
H	Berekeningen R onderzoeksvraag 3	97
I	Berekeningen R onderzoeksvraag 4	99
	Bibliografie	101

Lijst van figuren

2.1	Het waarom, wat en hoe van User Experience Design	24
2.2	Belangrijke factoren bij User Experience	25
2.3	De User Experience honingraat	26
2.4	Voorbeeld Medium desktop	27
2.5	Voorbeeld Medium mobiel	28
2.6	Voorbeeld Airbnb	29
2.7	Voorbeeld Nest Smart Thermostat	30
2.8	Voorbeeld Hubspot	31
2.9	Voorbeeld session recording	34
2.10	Consistentie doorheen UI-elementen	36
2.11	Consistentie doorheen Cardify producten	37
2.12	Implementatiefout bij onboarding	38
2.13	Voorbeeld welkomstbericht	40
2.14	Voorbeeld tooltip	41
2.15	Voorbeeld rondleiding Slack	41
2.16	Voorbeeld voortgangsindicatoren	42
2.17	Voorbeeld checklist	42

2.18 Voorbeeld hotspots	43
2.19 Voorbeeld uitgestelde accountcreatie	44
2.20 Voorbeeld gepersonaliseerde onboarding	44
2.21 Voorbeeld help-sectie	46
2.22 Voorbeeld UI-designproces	47
3.1 Beschrijving van de steekproef	50
3.2 Basisfunctionaliteiten van de proof-of-concept applicatie	52
3.3 Uitgebreide opties van de proof-of-concept applicatie	52
3.4 Icoon van Piggy, de proof-of-concept applicatie	53
3.5 Een rondleiding doorheen de proof-of-concept applicatie	54
3.6 Uitgebreide help-sectie in de proof-of-concept applicatie	54
4.1 Had de participant hulp nodig bij het voltooien van de opdrachten? 61	
4.2 Verschillende manieren om een bedrag aan een spaardoel toe te voegen	62
4.3 Of de participant het “+”-symbool gebruikt bij het toevoegen van een bedrag	62
4.4 Antwoorden op vraag 4 en 10 van de SUS-vragenlijst	62
4.5 De voorkeur van de participant: een applicatie met of zonder onboarding en help-elementen	63
4.6 Ziet de participant zichzelf de help-sectie gebruiken	63
4.7 Of de participant de applicatie zou houden	63
5.1 Navigatie in de proof-of-concept applicatie	68
5.2 De naam van de gebruiker wijzigen in de proof-of-concept applicatie	69
5.3 Of de participant de naam wijzigde bij het verkennen van de applicatie	69

Lijst van tabellen

4.1	Beschrijving van tijden zonder het gebruik van onboarding en help-elementen	58
4.2	Beschrijving van tijden met het gebruik van onboarding en help-elementen 58	
4.3	Beschrijving van de SUS-score zonder of met learnability-elementen 59	
4.4	Beschrijving van SUS-vragen omtrent learnability zonder onboarding en help-elementen	59
4.5	Beschrijving van SUS-vragen omtrent learnability met onboarding en help-elementen	59
5.1	χ^2 resultaten indien de participant hulp nodig had	65
5.2	t-testen van alle opdrachten	66

Acroniemen

CLI Command Line Interface. 36

CTA Call-To-Action. 40

FAQ Frequently Asked Questions. 45

GUI Grafische User Interface. 36

ROI Return on Investment. 37

SUS System Usability Scale. 35, 51, 55

UI User Interface. 3, 13, 14, 23, 36, 40, 42, 44, 46, 47

UX User Experience. 3, 7, 13, 19, 23–27, 30, 33, 35, 36, 40, 44–47, 67, 68, 71

UXD User Experience Design. 13, 23, 24, 26

1. Inleiding

If the user can't use it, it doesn't work.

Susan Dray

In digitale tijden als deze is de vraag naar nieuwe software groot. Programmeurs en IT-bedrijven hebben hun handen vol. Zowat elke sector wil mee zijn met de digitale boot. Je komt hedendaags overal software tegen; de computer op het werk, de smartphone in je broekzak, het bedieningspaneel van een grote kraan, de kassa in de supermarkt, ... Je gebruikt hoogstwaarschijnlijk tal van software in het dagelijkse leven. Maar hoe leer je nu het best omgaan met deze computerprogramma's? Programma's bevatten vaak heel veel functies. Veel van deze functies blijven echter onbenut omdat de gebruiker niet voldoende begrijpt hoe deze functies in zijn werking treden. De programmeurs achter deze software hebben vaak hun handen al vol met het ineen knutselen van al deze functionaliteit waardoor zij zelf geen tijd hebben om te kijken hoe de eindgebruiker met deze functionaliteit omspringt.

How do I explain what I do at a party?
The short version is that I say I
humanize technology.

Fred Beecher, Director of UX, The Nerdery

Hier komt de UX-designer aan bod. Een UX-designer zorgt er voor dat software bruikbaar is voor de eindgebruiker. Taken die zijn job omschrijven omvatten, maar zijn niet beperkt

tot, het maken van prototypes, testen van (deel)producten bij gebruikers, observeren van gedrag van gebruikers op bepaalde functionaliteiten en stukken software, *user flows* creëren en ook onderzoek doen naar de doelgebruiker (White, 2020). De UX-designer zal dus ook een flow creëren dat ervoor zorgt dat de gebruiker alle functionaliteit van de applicatie goed begrijpt en snel onder de knie heeft.

Eén van de bekendste implementaties hiervan is de “onboarding”. Je komt vaak in contact met onboarding wanneer je de applicatie voor de eerste maal opstart. Zo’n onboarding kan zeer verschillend zijn van applicatie tot applicatie. Er bestaan uiteraard meer manieren om de gebruiker de weg te wijzen doorheen software. Een simpele *tooltip* of zelfs een help-pagina of leerplatform doet ook wonderen.

1.1 Probleemstelling

Bij veel software-ontwikkelaars en -designers stelt de vraag zich frequent of een bepaalde implementatie van onboarding of in-app user training wel het gewenste effect bekomt. Er bestaan verscheidene manieren om dit in een applicatie te verwerken, om echter te weten of de ene manier een beter resultaat boekt dan de andere is een onderzoek nodig.

We bekeken de voordelen van onboarding en in-app user training tot nu toe voornamelijk vanuit het oogpunt van de eindgebruiker. Echter kan dit ook effect hebben op andere aspecten binnen een bedrijf. Deze scriptie zal ook de effecten op het klantbehoud analyseren. Er zal dus onderzocht worden of de klant zich meer geneigd gaat voelen een bepaalde applicatie met een andere en/of betere implementatie van de verschillende technieken frequenter te gebruiken.

1.2 Onderzoeksraag

1.2.1 Hoofdonderzoeksraag

Zoals reeds aangehaald in sectie 1.1 zal dit onderzoek zich focussen op implementaties van onboarding en in-app user training en de gevolgen van deze implementaties. Daaruit vloeit volgende hoofdonderzoeksraag voort:

- Kan een (betere) onboarding en in-app user training ervoor zorgen dat de eindgebruiker een beter inzicht heeft op de totale functionaliteit van een grote applicatie?

1.2.2 Deelonderzoeksraag

Ter ondersteuning van de hoofdonderzoeksraag zijn er ook nog enkele deelonderzoeks-vragen opgesteld:

- Hoe een grote hoeveelheid aan functionaliteiten beheersbaar houden voor de eindgebruiker?
- Heeft (het gebrek aan) in-app user training effect op de gebruiksduur en/of levensduur van de applicatie?
- Hoe de eindgebruiker wegwijs maken in een grote applicatie?

Doorheen deze scriptie zal op deze deelonderzoeksvragen een antwoord geformuleerd worden.

1.3 Onderzoeksdoelstelling

Het hoofddoel van dit onderzoek is het aantonen van de werking en gevolgen van verschillende technieken om de eindgebruiker familiair te maken met de applicatie. Door middel van een proof-of-concept zal worden aangetoond welke technieken het meest gewenste resultaat hebben op de eindgebruiker.

Een tweede doel bestaat uit het onderzoeken of (het gebrek aan) deze technieken ook effect heeft op de gebruiksduur en/of levensduur van de applicatie. Zo zal gemeten worden of de gebruiker, na het eerste gebruik van de software, de applicatie wil houden of niet.

Een laatste doel is om er voor te zorgen dat ontwikkelaars die dit lezen een idee krijgen van de UX-technieken die men kan gebruiken bij de implementatie van hun applicatie.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeks domein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4 worden de gemeten waarden uit de proef (visueel) weergegeven.

In Hoofdstuk 5 formuleert men, met behulp van statistische analyses, de bevindingen die af te leiden zijn uit de metingen.

In Hoofdstuk 6, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Stand van zaken

Zoals uit het vorig hoofdstuk kan worden afgeleid, zal deze scriptie onderzoek doen naar de implementatie en effecten van bepaalde UX en UI-elementen. Maar alvorens van start te gaan hiermee zal er gekeken worden naar wat de algemene rol is van UX en UI in software-ontwikkeling. Wanneer alle aspecten van UX design duidelijk zijn bekijken we hoe men de usability van een applicatie of product kan testen. Daaruit gaan we verder naar de verschillende technieken die men kan implementeren om het eenvoudiger te maken voor een gebruiker om de applicatie te gebruiken, zoals onboarding en in-app training.

In dit hoofdstuk worden ook enkele interviews verwerkt (bijlage B en C). Zo wordt er een duidelijke link gelegd met het werkveld en de praktijk.

2.1 User experience in software

In het traditioneel proces van het ontwikkelen van software staat de functionaliteit centraal. Ontwikkelaars bekijken alle vereisten en starten met de belangrijkste. Functionaliteit krijgt hier doorgaans de voorkeur. Harutyunyan en Riehle (2019) stelden vast dat dit de laatste jaren echter aan het wijzigen is. De traditionele softwareontwikkeling is plaats aan het maken voor softwareontwikkeling met User Experience in het achterhoofd. Dit fenomeen noemt men User Experience Design (UXD). Omdat de term UXD in de literatuur nog sterk evolueert, heeft deze nog geen algemeen aanvaarde definitie. Men kan stellen dat User Experience Design een proces is waarbij men gebruiksgedrag zal manipuleren aan de hand van de bruikbaarheid en wenselijkheid in de interactie met een product.

Men doet al lang onderzoek naar User Experience in software. Zo toonden Carroll en

Carrithers (1984) het belang van training in complexe systemen al aan anno 1984. Deze training van gebruikers behandelen we later in dit hoofdstuk.



Figuur 2.1: Het waarom, wat en hoe van User Experience Design

Een UX-designer bekijkt het product niet enkel als het product op zich. Deze persoon analyseert hoe de eindgebruiker het product in gebruik neemt en past het product aan zodat de gebruikservaring optimaal is. De designer neemt het *waarom*, *wat* en *hoe* van productgebruik in acht (figuur 2.1) (Hassenzahl, 2013). De *wat* in productgebruik verwijst gewoonlijk naar wat een gebruiker kan doen door middel van het product. Dit is bijvoorbeeld “een foto maken” of “een spel kopen”. De *hoe* staat dan ook effectief voor hoe de gebruiker het product gebruikt. Dit is meer op een operationeel niveau zoals het navigeren door software met behulp van knoppen en andere attributen. De designer zal zich voornamelijk focussen op het “hoe” van het productgebruik. Dit omvat onder andere de gegeven functionaliteit op een aantrekkelijke manier zeer toegankelijk maken.

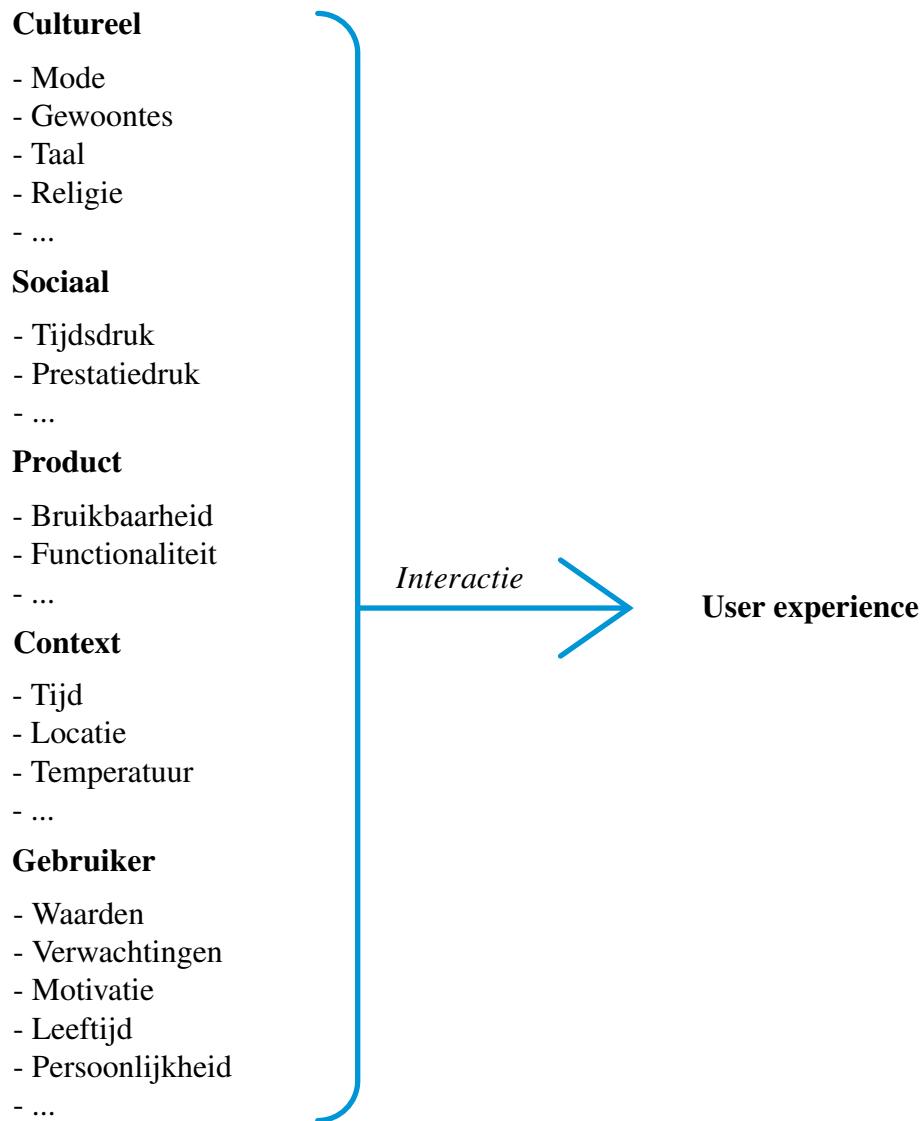
2.2 Belangrijke factoren bij user experience

Er zijn talloze factoren die ervoor zorgen dat men op een verschillende manier naar dezelfde applicatie moet kijken. Zo zijn de gebruikers allemaal verschillend, maar er moet ook rekening gehouden worden met verschillende omgevingen, culturen, enz. Een toestel om parameters op te meten bij schepen moet dus waterbestendig zijn. Een kindvriendelijke tablet is best schokbestendig. Een applicatie om notities te maken bij vergaderingen is best geluidloos.

De factoren kunnen gegroepeerd worden in vijf groepen (figuur 2.2). Culturele factoren omvatten bijvoorbeeld religie, taal en gewoontes. Zo moet men bij het ontwerpen van een website met asielzoekers als doelgroep bijvoorbeeld rekening houden met het gebrek aan kennis van de landstaal.

Een mobiele applicatie waarbij de gebruiker een vervoersbewijs moet voorleggen op een voertuig van het openbaar vervoer zal bijvoorbeeld rekening moeten houden met het feit dat de sociale factor tijdsdruk hier belangrijk is. Indien deze applicatie niet tijdig het vervoersbewijs laat zien, zal er een hele wachtrij ontstaan die dan vertragingen tot gevolg heeft.

Onder factoren met betrekking tot de context waarin het product gebruikt wordt kan men bijvoorbeeld tijd en locatie plaatsen. Bij het bestellen van een pakket krijg je vaak een tracking-link waarbij ook het tijdstip van levering staat. Een internationale leverancier moet dus zeker voorzien dat de tijd van de levering in de juiste tijdzone weergegeven wordt.



Figuur 2.2: Belangrijke factoren bij User Experience

De gebruiker zelf verschilt uiteraard ook. Een applicatie gericht op een ouder publiek voorziet best grote tekst en duidelijke iconen.

Het product zelf moet uiteindelijk ook nog bruikbaar zijn en alle functionaliteiten moeten eenvoudig bereikbaar zijn. Een hele boterham voor de User Experience designer om onderzoek naar te doen voor zijn use case.

Morville (2004) verdeelde User Experience op een andere manier. Hij maakt gebruik van de User Experience honingraat (figuur 2.3) die User Experience opsplits in zeven onderdelen.

- **Nuttig.** Alle producten moeten een zeker nut hebben. Een applicatie mag niet zomaar een tool zijn van het management maar moet een zekere waarde hebben voor de eindgebruiker.
- **Bruikbaar.** De bruikbaarheid of usability van een product is een van de belang-

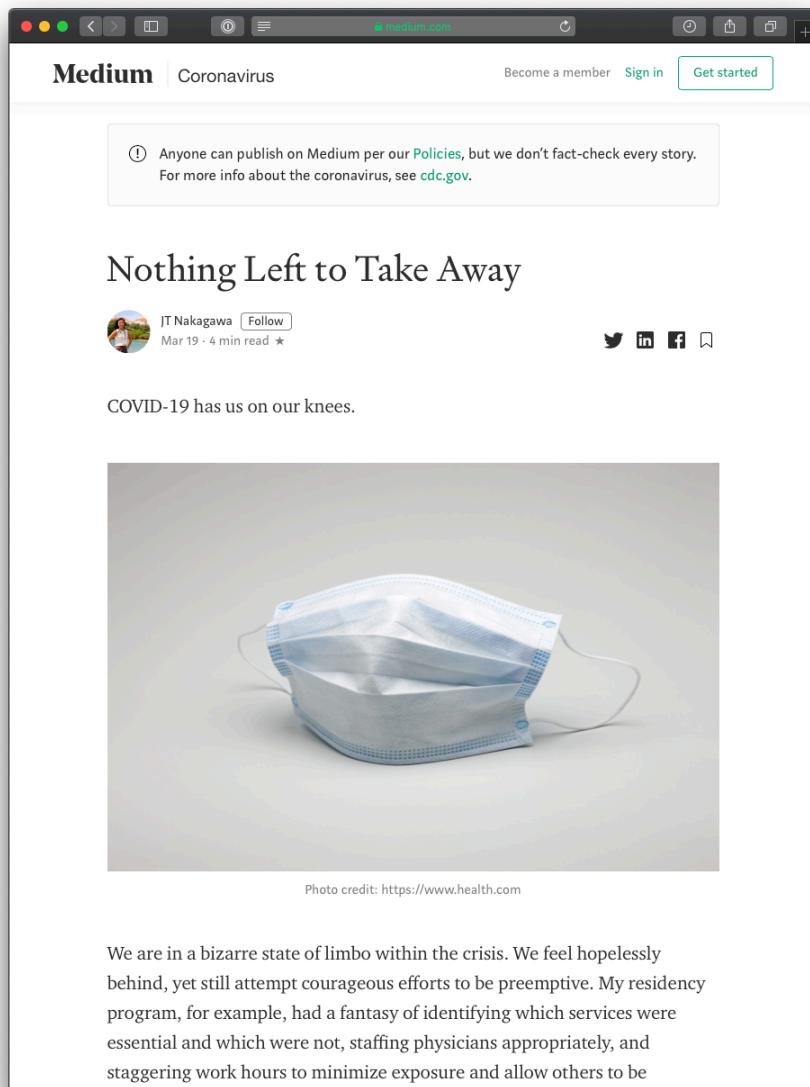


Figuur 2.3: De User Experience honingraat

rijkste kenmerken van de User Experience. Het is echter niet het enige kenmerk. Bruikbaarheid en gebruiksgemak zijn dus essentieel maar niet voldoende.

- **Gewenst.** De zoektocht naar een efficiënte applicatie mag de branding, het image en de esthetiek van de applicatie niet achterwege laten. Hoe wenselijker het product is, hoe meer de gebruiker erover zal opscheppen tegen potentieel nieuwe gebruikers.
- **Vindbaar.** Software moet eenvoudig te navigeren zijn. Gebruikers moeten vlot kunnen vinden wat ze nodig hebben.
- **Toegankelijk.** Software moet toegankelijk zijn voor alle doelgroepen. Een gebruiker met een handicap mag geen hindernissen ondervinden bij het gebruik ervan.
- **Geloofwaardig.** De design elementen gebruikt in de software moeten ervoor zorgen dat de gebruikers vertrouwen hebben in de informatie die we hen meedelen.
- **Waardevol.** De software moet waarde leveren voor de organisatie. De organisatie zal er naar streven dat de winst en klanttevredenheid sterk toenemen.

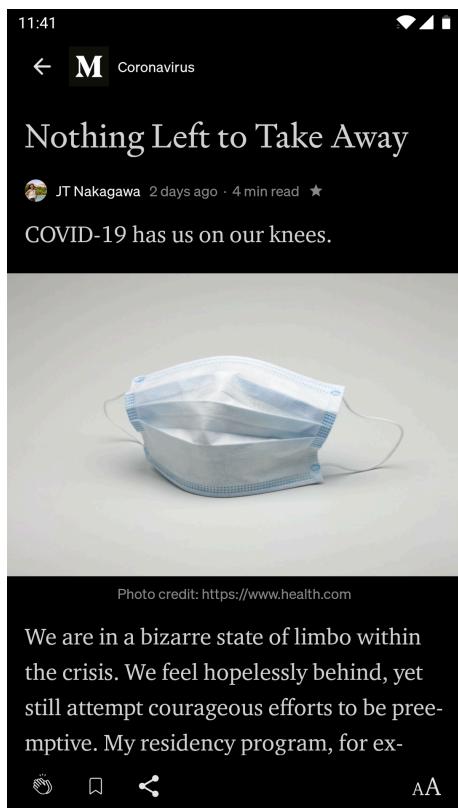
User Experience Design is een zeer creatief concept. Door deze creativiteit zijn er uiteraard verschillende meningen over hoe men User Experience moet definiëren, omschrijven en indelen. Gezien de belangrijkste manieren vermeld zijn, gaan we hier niet verder op in.



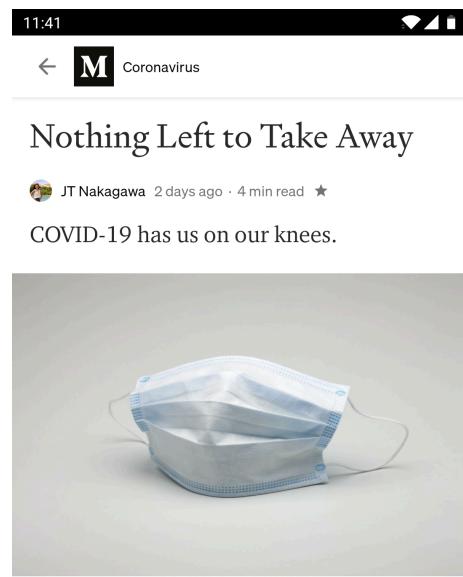
Figuur 2.4: Artikel op Medium weergegeven in een desktop-omgeving

2.2.1 User Experience design in de praktijk

Een eenvoudige applicatie moet simpel in gebruik zijn, en dit is waar de website **Medium** (<https://medium.com/>) op inzet. Medium is een online platform voor schrijvers en lezers. Bij een blog-artikel moet men focussen op de inhoud van het artikel. Door een gebrek aan kleurgebruik en een goede keuze van het lettertype is Medium gebruiksvriendelijker dan de papieren krant. Afbeeldingen zijn groot en duidelijk, de titel springt eruit en op enkele iconen na zijn er weinig tot geen afleidingen te bespeuren (figuur 2.4). Medium trekt deze lijn door naar hun mobiele applicatie (figuur 2.5). Hier implementeerde men ook een donkere variant. Deze variant zorgt voor leescomfort in donkere omgevingen en in sommige gevallen ook voor batterijbesparing (Jin e.a., 2017).

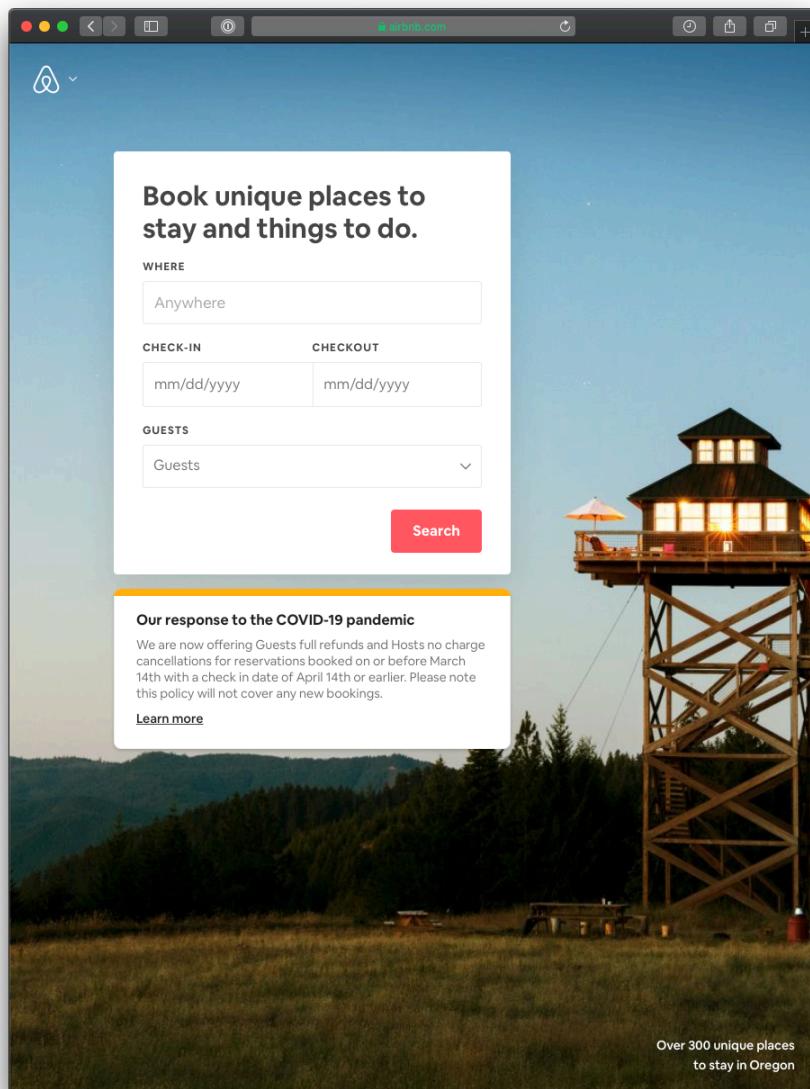


(a) Donkere gebruikersomgeving



(b) Lichte gebruikersomgeving

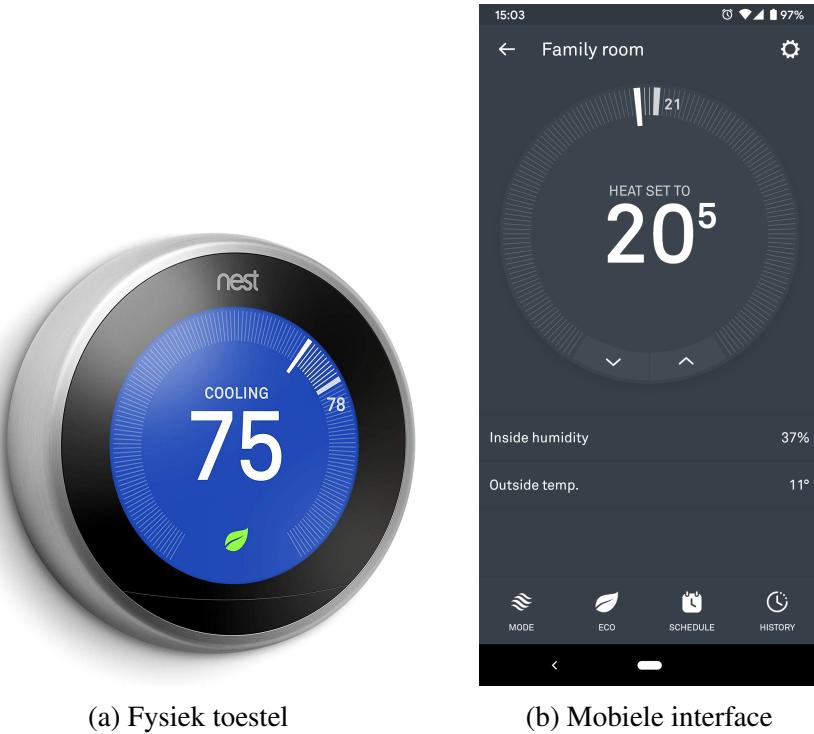
Figuur 2.5: Artikel op Medium weergegeven in een mobiele omgeving



Figuur 2.6: Airbnb homepagina

Airbnb wil verkopen, dat is merkbaar van zodra je de website opent. Airbnb (<https://www.airbnb.com/>) is een platform waarmee je een kamer of woning van iemand anders kan huren voor een korte periode. Het is een razend populair platform bij mensen die een plezier- of werkreis plannen en iets unieks zoeken of de kosten van hun reis willen drukken. Van zodra je op de homepagina komt zorgt Airbnb ervoor dat je onmiddellijk kan zoeken naar een geschikte plaats om te verblijven op een locatie en tijdstip naar keuze. Gepaard met een uitnodigende titel en een buitengewone afbeelding is de verleiding bij de gebruiker groot om hun ideale trip te beginnen plannen. Door deze directe aanpak vergeet de gebruiker als het ware de concurrentie van Airbnb, wat uiteraard net het doel was van bij het begin.

Slimme apparaten in het huishouden zijn bezig aan een opmars. Eén van de bekendste ap-



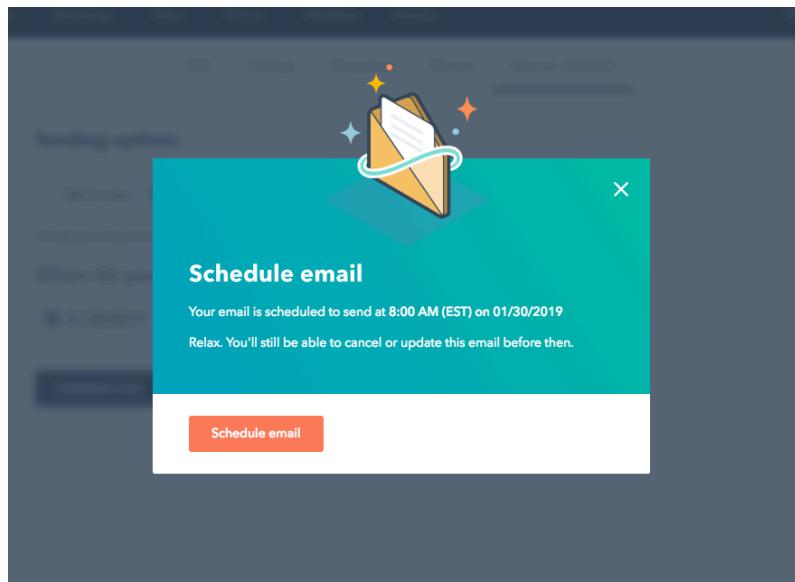
Figuur 2.7: Nest Smart Thermostat

paraten is de **Nest Smart Thermostat**, een slimme thermostaat die verbonden is met het internet. De thermostaat leert wanneer de ruimte moet opwarmen of afkoelen. Het fysieke toestel zelf is de simpliciteit zelve. Het is een grote, ronde knop met centraal de huidige temperatuur, hij vermeldt of de ruimtes opwarmen of afkoelen en ook de temperatuur waar men naartoe werkt. Een draai naar rechts en je verhoogt de gewenste temperatuur, een draai naar links en je verlaagt deze. De mobiele applicatie bootst deze werking na zodat de gebruiker zich direct comfortabel voelt met de interface (figuur 2.7).

Hubspot (<https://www.hubspot.com/>) maakt producten voor marketing, sales en klantenservice. Hubspot richt zich tot bedrijven van elke omvang. In dit voorbeeld bekijken we slechts een klein deel van de software. Binnen Hubspot is er de mogelijkheid om een massa-e-mail te plannen die een deel van of alle klanten van de gebruiker bereikt. Uiteraard is dit een actie met een aanzienlijke impact op het bedrijf van de gebruiker. Een verzonden e-mail kan echter niet ongedaan gemaakt worden. Wanneer de gebruiker de e-mail inplant komt Hubspot met een pop-up melding die toch nog even bevestiging vraagt (figuur 2.8). Deze melding zorgt ervoor dat de kans op fouten verkleind wordt, maar ook dat de gebruiker zich gerust voelt bij zijn acties.

2.3 Usability testing

Usability testing is een methode om alle onderdelen van de User Experience honingraat (figuur 2.3) in een applicatie of website te testen. Men test de software in kwestie door er echte gebruikers op te laten werken en ondertussen hun handelingen te observeren. Het



Figuur 2.8: Hubspot pop-up melding

doel van usability testing is om de algemene gebruikerservaring te verbeteren (Hotjar, 2020).

Bij het creëren van software vergeet de ontwikkelaar al snel dat de gewone eindgebruiker vaak meer moeilijkheden zal ondervinden bij het gebruik van zijn creatie dan hijzelf. Doordat de ontwikkelaar hier zelf vaak blind voor is, voert men testen uit met de eindgebruiker. Hierdoor kan men een beter inzicht krijgen over hoe bruikbaar de software is. In dit proces worden vaak vele euvels opgemerkt die anders in de productiesoftware zouden aanbelanden. Zonder usability testing zou men vaak vast komen te zitten met een product dat door het team van ontwikkelaars wordt begrepen, maar door de doelgroep niet.

In onderstaande testmethoden verwijst men vaak naar het “think aloud” protocol. Dit think aloud protocol houdt in dat de participant al zijn of haar acties luidop verklaart. Zo heeft de moderator een beter beeld over de gedachtegang van de participant.

Uit een interview met Seppe Vereecken (Product & Growth bij Cardify) blijkt dat men bij Cardify de gebruiker eerst wat laat werken met de software om nadien vragen te stellen over de acties van de gebruiker. Om te controleren indien de gebruiker de software begrepen heeft laat men deze nadien nog enkele opdrachten uitvoeren (bijlage C).

2.3.1 Laboratory en field testing

In voorgaand onderzoek tonen Kaikkonen e.a. (2005) aan dat usability testing onderverdeeld kan worden in twee categorieën, namelijk laboratory en field testing. Bij field testing worden te testpersonen uitgenodigd om de applicatie in kwestie te testen in de omgeving waar de applicatie normaal zou gebruikt worden. Een mobiele applicatie zoals Strava die statistieken van lopers en fietsers bijhoudt zou bijvoorbeeld getest worden tijdens een trainingssessie. Sinds de opmars van de smartphone kan field testing steeds

meer ingezet worden (Kjeldskov e.a., 2004). De camera die gebruikt wordt om de testsessie op te nemen wordt steeds kleiner, waardoor het field testing een pak aangenamer wordt.

Laboratory testing houdt in dat men enkele testpersonen uitnodigt in testlabo's, dit is gewoonlijk op kantoor. Dit labo is een rustige ruimte waarin afleidingen beperkt zijn zodat de concentratie gegarandeerd kan worden. Ook al bestaan er veel twijfels rond laboratory testing, volgens Kjeldskov en Graham (2003) wordt het nog steeds vaker verkozen boven field testing. De reden achter deze keuze is vaak omdat men moeilijkheden ondervindt bij field testing. Het is eenvoudiger om testpersonen op te nemen en te observeren bij technieken zoals "think aloud" wanneer men gebruik maakt van laboratory testing.

In beide gevallen van usability testing werd geopteerd voor het "think aloud" protocol gebaseerd op het werk van Ericsson en Simon (1984). Hierbij zal de testpersoon luidop denken. Dit zorgt ervoor dat er veel informatie vrijkomt over de gedachtegang van de gebruiker bij het gebruik van de applicatie. Dit kan makkelijk opgenomen worden om later conclusies uit te trekken.

Kaikkonen e.a. (2005) hebben ook kunnen afleiden dat zowel laboratory testing als field testing quasi dezelfde resultaten vertonen. In hun use case werden alle usability fouten bij beide testmethodes gevonden. Bij field testing vond men de fout soms wel sneller of kwam men deze meerdere keren tegen.

2.3.2 Andere testmethoden

Laboratory en field testing zijn slecht twee van vele methodes waarmee de bruikbaarheid van een applicatie kan getest worden. In een artikel van Babich (2019) worden de zeven belangrijkste methodes opgesomd.

Guerilla testing

Guerilla testing is de eenvoudigste testmethode om zo snel mogelijk resultaten te krijgen. Bij guerilla testing gaat de moderator simpelweg op een publieke plaats aan willekeurige personen vragen om even het prototype van de applicatie te gebruiken. De moderator noteert dan de bevindingen. Guerilla testing voert men het best uit in het begin van het ontwikkelproces, zo weet men snel of men in de juiste richting werkt. Vaak krijgt de testpersoon een kleine attentie na het uitvoeren van de test. Zo kan de moderator bijvoorbeeld in een koffiebar plaatsnemen en testpersonen uitnodigen met een gratis koffie.

Lab usability testing

Deze testmethode werd eerder besproken in hoofdstuk 2.3.1. In contrast met geurilla testing kan men hier een gerichter publiek aantrekken en kan men zo relevantere informatie verkrijgen.

Unmoderated remote usability testing

Dit is een variant van field testing (hoofdstuk 2.3.1) zonder moderator. Hierbij gebruikt de testpersoon de applicatie in zijn omgeving en op zijn toestel. De informatie die men hierdoor verkrijgt is door het gebrek van een moderator uiteraard beperkter. Het voordeel van deze methode is dat de kost lager is dan andere testmethoden.

Contextual inquiry

Deze methode leunt meer aan bij een observatiemethode dan bij een testmethode voor usability testing. De testpersonen krijgen hierbij een lijst met vragen die ze moeten beantwoorden over het product. Achteraf worden ze vaak gevraagd de applicatie te gebruiken in hun omgeving met een moderator (zoals bij field testing).

Phone interview

Deze methode is een variant van laboratory testing waarbij men de testpersoon telefonisch opdrachten geeft. Dit gesprek kan opgenomen worden om later conclusies uit te trekken. Bij deze testmethode worden de communicatievaardigheden van de moderator op proef gesteld.

Card sorting

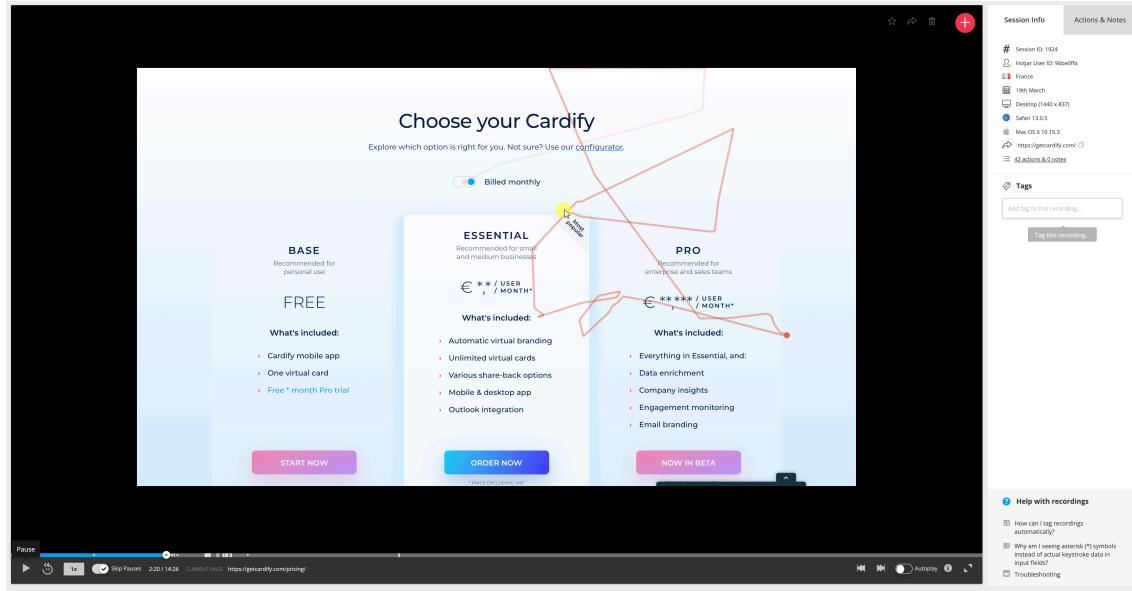
Hierbij worden alle functionaliteiten van de applicatie op kaartjes geschreven. De testpersoon plaats deze kaarten dan in categorieën en geeft deze kaartjes een bepaalde prioriteit. Van zodra de testpersoon aan het werk gaat met de kaartjes moet de moderator uitleg vragen waarom de testpersoon bepaalde kaartjes op een bepaalde plaats legt.

Session recording

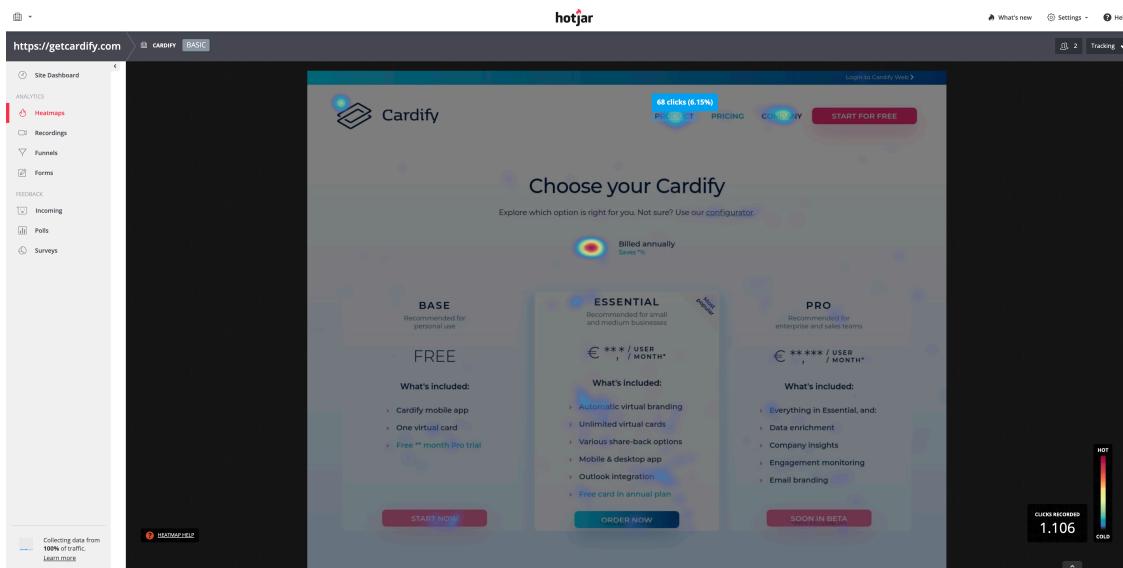
Deze methode legt de acties vast van echte (anonieme) gebruikers. Zo kan men nadien bekijken hoe de sessie bij de gebruiker verlopen is en waar er eventueel User Experience fouten zitten. Ook kan men door deze opnames beter begrijpen welke inhoud en functies belangrijk zijn voor de gebruiker. Deze bevindingen kan men makkelijk aflezen van een heatmap-analyse. Een veelgebruikte tool om sessies op te nemen en heatmap-analyses te genereren is Hotjar. Figuur 2.9 geeft een voorbeeld van een opname en een heatmap-analyse van de website van Cardify.

2.3.3 Testpersonen

De testpersonen bij usability testing moeten zorgvuldig gekozen worden. De doelgroep moet nauwkeurig vastgelegd worden. Zo zal een bejaarde dame geen meerwaarde bieden bij een applicatie gericht op schoolgaande jeugd. In onderzoek van Alnashri e.a. (2016) bekijkt men ook de verschillen tussen introverte en extroverte personaliteiten.



(a) Opname handelingen van een gebruiker



(b) Heatmap-analyse

Figuur 2.9: Voorbeeld session recording bij Cardify

Zoals Faulkner (2003) aanhaalde heeft het weinig nut om honderden testpersonen te zoeken. Twintig personen zou volstaan om 95% van de usability problemen op te merken. In een interview met Quentin Braet, Lead Solution Architect bij In The Pocket, haalde ook hij aan dat de meeste UX problemen op te merken zijn bij het testen bij een tiental personen (bijlage B).

In voorgaand onderzoek van Marcus (2006) toont men aan dat er culturele verschillen een grote invloed hebben op UX. Hieruit kunnen we concluderen dat variëteit in de testpersonen zal lonen.

2.3.4 De System Usability Scale

De System Usability Scale is een snelle manier om, met behulp van een vragenlijst, te weten te komen of de usability van het product goed zit. De vragenlijst bestaat uit tien vragen met telkens vijf antwoordmogelijkheden; gaande van “helemaal mee eens” tot “helemaal niet mee eens”. Voor de testpersonen is dit eenvoudig in te vullen. De berekening nadien is echter complex.

Bijgevoegd aan deze scriptie is een template voor de System Usability Scale (Calisto & Nascimento, 2018) (bijlage D).

In deze vragenlijst bevinden zich twee vragen die testen op learnability in plaats van op usability (Lewis & Sauro, 2009).

2.4 Learnability

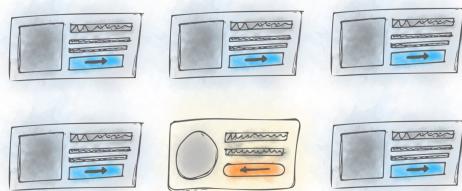
Er is nog steeds een verschil tussen hoe bruikbaar een applicatie is en hoe goed de eindgebruiker deze applicatie kent. Coyle en Peterson (2016) merkten op dat naast de bruikbaarheid (usability) van hun ontwerpen, er ook moet gekeken worden naar de leerbaarheid (learnability) ervan. Na een hele reeks testen en vragenlijsten met gebruikers uit de doelgroep kon men effectief afleiden dat, indien de leerbaarheid van een applicatie ook in acht genomen wordt, men diepere gebruiksproblemen gemakkelijker kan vaststellen. Bij hun onderzoek maakten ze gebruik van onder andere open vragen, de System Usability Scale (hoofdstuk 2.3.4) en vertrouwensratings gebruikmakend van de Likert schaal.

In onderzoek van Haramundanis (2001) werd beschreven welke stappen men moet ondernemen om leerbare materie te ontwikkelen:

- Analytisch denken
- Logische ontwikkeling
- Usability testen
- Consistentie van aanpak
- Visualisatie

Visualisatie is hierbij een belangrijk onderdeel. In de beginjaren van de personal computer

moest men nog door de mappenstructuur van het besturingssysteem bladeren met behulp van de Command Line Interface. De leercurve was toen uiteraard zeer steil waardoor de computer een product was waarmee weinig mensen mee overweg konden. Tegenwoordig kan iedereen door zijn of haar mappen bladeren in een besturingssysteem door middel van een Grafische User Interface. Deze verandering bij het gebruik van de personal computer zorgde er ook voor dat het gebruik van een computer makkelijk aan te leren was voor nieuwe gebruikers. Deze evolutie in het ontwerp van software zorgde ervoor dat bijna elk modern huishouden een computer ter beschikking heeft.



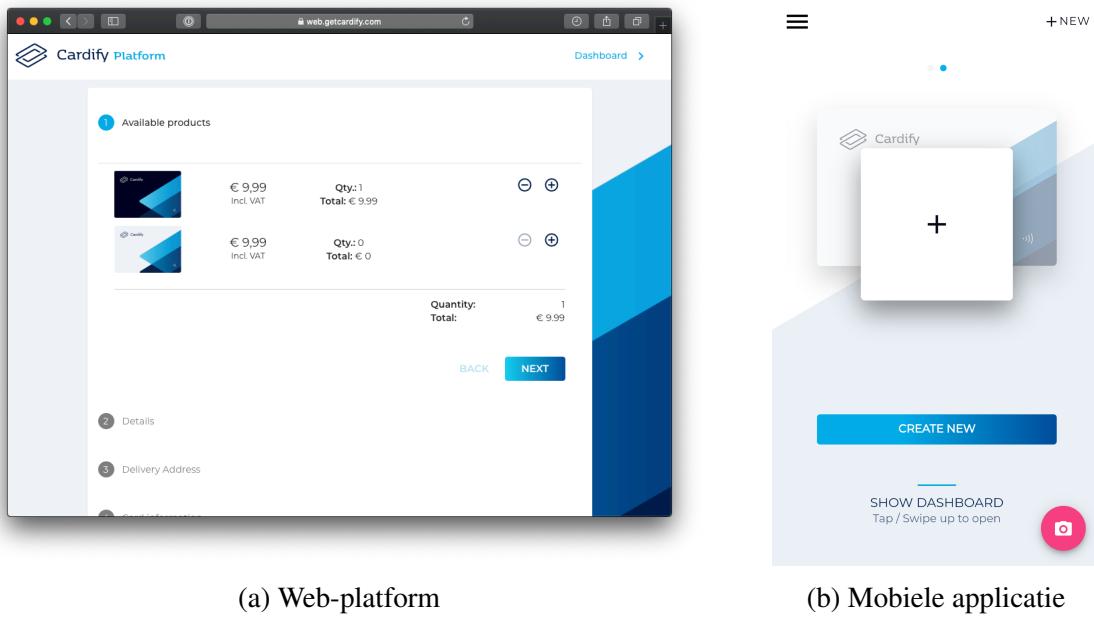
Figuur 2.10: Het belang van consistentie doorheen UI-elementen

Ook consistentie is belangrijk bij zowel learnability als UX design. Zoals Nikolov (2017) aangeeft is bestaan er vier verschillende types van consistentie: visuele, functionele, interne en externe consistentie. De visuele consistentie (figuur 2.10) leunt aan bij het hierboven genoemde visualisatie. Interne consistentie is een combinatie van visuele en functionele consistentie. Bij het behouden van interne consistentie kan men makkelijk een nieuwe functionaliteit of UI-element toevoegen aan bestaande software zonder dat eindgebruikers opnieuw moeten bijleren hoe de applicatie werkt. Externe consistentie trekt de lijn door naar andere systemen en producten. Bij Cardify blijft men consistent doorheen hun volledige aanbod aan producten (figuur 2.11). Zo zal een gebruiker van de mobiele applicatie zich snel vertrouwd voelen met de verscheidene webplatformen die Cardify te bieden heeft. Vaak gebruikt een bedrijf ook design guidelines. Grote bedrijven zoals Google gaan daarin natuurlijk wat verder. Zo ontwikkelde Google het Material Design systeem. Bij Cardify kan je de design guidelines vinden op <https://getcardify.com/brand/>.

2.5 Onboarding

Onboarding is slechts één element van de vele die de User Experience in software kunnen beïnvloeden. Het is echter wel één van de bekendste en belangrijkste. Voorgaand onderzoek toonde aan hoe onboarding een positieve invloed had op de tevredenheid van de gebruiker over de applicatie (Cardoso, 2017). Onboarding vindt doorgaans plaats wanneer de gebruiker de applicatie voor de eerste maal opstart en gebruikt. Het is een soort van gids die de gebruiker de belangrijkste functionaliteiten van de applicatie laat zien. Vaak wordt onboarding ook ingezet voor het verkrijgen van essentiële informatie van de gebruiker, bijvoorbeeld wanneer deze een account moet aanmaken.

Een onboarding kan de ervaring van de gebruiker zeer positief beïnvloeden. Een slecht



Figuur 2.11: Consistentie doorheen Cardify producten

geïmplementeerde onboarding kan de ervaring weliswaar heel snel negatief beïnvloeden. Er mag dus zeker niet overdreven worden. Zowel Quentin als Seppe, Product & Growth bij Cardify, bevestigen de gouden regel: “keep it simple” (bijlage B en C).

Voorgaand onderzoek (Desai, 2019) toonde aan dat een goed geïmplementeerde onboarding (een onboarding gebaseerd op wat de gebruiker wil bereiken) wel degelijk een heilboel voordelen heeft. Zo zijn klanten met een positieve perceptie minder geneigd om te stoppen met het gebruik van de applicatie in de eerst 21 dagen dat ze klant zijn. In ideale omstandigheden stijgt de bereidheid om te betalen voor de software met 27%, wat dus resulteert in een gunstigere Return on Investment (ROI).

In dit hoofdstuk zullen de verschillende implementaties van onboarding verder verduidelijkt worden. De verschillende types alsook de UX elementen die in omloop zijn zullen worden besproken. Er wordt aangevangen met hoe men best start met de implementatie van onboarding.

2.5.1 Starten met de implementatie van onboarding

Starten aan de implementatie van de onboarding is geen eenvoudige klus. Bij Cardify werd er geluisterd naar wat de gebruikers nodig hadden. De onboarding werd per iteratie verfijnd. Bij In The Pocket ontwikkeld men vaak complexe applicaties. Om alle functionaliteiten duidelijk over te brengen naar de gebruiker start men al met het uitdenken van onboarding in de ontwerp fase. In The Pocket gaat zeer frequent feedback vragen bij testgebruikers, dit zelf voor men start met de effectieve ontwikkeling van de software. Door vroeg te testen komen snel problemen boven die in latere fasen moeilijker op te lossen zijn. Ook nadat de applicatie uitgebracht is, volgt In The Pocket de gedragingen van de



Figuur 2.12: Bij de ingebruikname van de nieuwe Apple iPad Pro werd de onboarding onderbroken door een programmeurfout

gebruiker nog op. Wanneer men problemen opmerkt, probeert men die onmiddellijk aan te pakken. Er wordt ook gewerkt met een phased rollout systeem; hierbij lanceert men (een update van) de applicatie voor slechts een deel van de gebruikers, zodat men bij complicaties vroegtijdig kan ingrijpen zonder dat alle gebruikers er hinder van ondervinden. Om snel aan de slag te gaan met onboarding heeft Apple een gids voorzien in hun Human Interface Guidelines. Apple blijft bij de vijf basisprincipes van onboarding, deze worden besproken verder in dit hoofdstuk.

De onboarding zorgt er uiteraard voor dat gebruikers weten hoe de applicatie werkt. Er mag echter niet vergeten worden dat de gebruiker van de applicatie moet kunnen genieten. Onboarding mag dus niet aanvoelen als een verplicht stappenplan waar de gebruiker doorheen moet. De gebruiker moet de onboarding zien als een toffe, leerrijke initiatie tot het gebruik van de applicatie. Het gebruik van animaties en interactie kan ervoor zorgen dat de gebruiker de onboarding op een positieve manier ervaart. Zoals vermeld in het interview met Quentin (bijlage B) zijn er soms verplichte secties in de onboarding die niet mogen ontbreken, zoals het vragen van permissies of een legal onderdeel (terms & conditions en privacy disclaimer).

De gebruiker wil snel aan de slag met de applicatie. Maak de onboarding dus niet onnodig lang. Bij initieel gebruik volstaat het om de basisfunctionaliteit aan te halen, er kan later uitgebreid worden naar complexere functionaliteiten of functionaliteiten die verder verborgen zitten.

Er moet proactief gezocht worden naar stukken in de software waar de gebruiker vast kan komen te zitten. Bij deze stukken kan doorheen de applicatie hulp voorzien worden in de vorm van een onboarding. Dit kan bestaan uit elementen zoals bijvoorbeeld tooltips of hotspots (hoofdstuk 2.5.3 en 2.5.3).

Kort samengevat moet men rekening houden met volgende vijf basisprincipes:

1. Zorg dat gebruikers van de applicatie kunnen genieten.
2. Ga snel naar de actie.
3. Anticipeer op de behoefte aan hulp.
4. Blijf bij de essentie.
5. Maak leren leuk.

2.5.2 Verschillende types van onboarding

Er bestaan verschillende manieren om een gebruiker aan boord te krijgen van de applicatie. Zo zal een onboarding van een applicatie die een product of dienst probeert te verkopen er anders uit zien dan een onboarding van software die gebruikt wordt op de werkvloer of onboarding om de werking van een computerspel uit te leggen. Bufe (2020) toonde in voorgaand onderzoek aan dat onboarding gesplitst kan worden in drie soorten. Bij de effectieve implementatie van deze onboarding is het wel mogelijk dat men hier een combinatie van neemt.

Onboarding gericht op de voordelen van de applicatie

Wanneer men zich voornamelijk richt op de voordelen van de applicatie zal men in de onboarding uitleggen welk probleem de gebruiker kan oplossen met behulp van de applicatie. Er wordt bij deze manier van onboarding dus niet gefocust op hoe de functionaliteit werkt. Deze onboarding wordt ingezet om de gebruiker ervan te overtuigen de applicatie te blijven gebruiken.

Functiegerichte onboarding

Functiegerichte onboarding focust zich, zoals de benaming al deed vermoeden, op de functionaliteiten binnen de applicatie. Deze manier van onboarding wordt vaak ingezet wanneer de applicatie enkele complexe functies heeft. Het is belangrijk dat initieel enkel de essentiële en belangrijkste functionaliteiten uitgelegd worden. Gedetailleerde uitleg kan in latere fases voorzien worden.

Progressieve onboarding

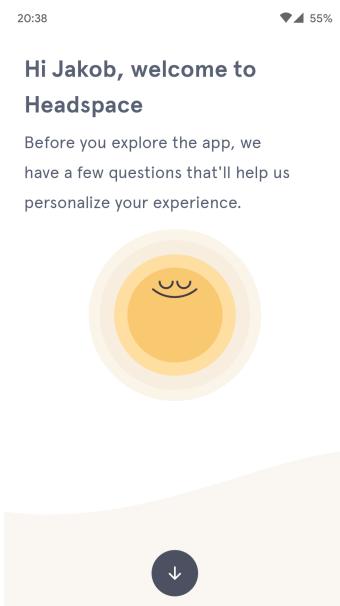
Progressieve onboarding laat de gebruiker kennis maken met de applicatie door deze te gebruiken. Aan de hand van een rondleiding leert de gebruiker wat de mogelijkheden en voordelen zijn van de software. Progressieve onboarding is een uitstekende aanpak als de applicatie verborgen functionaliteiten, verborgen gebaren of ingewikkelde workflows heeft.

2.5.3 Verschillende technieken en implementaties van onboarding

Balboni (2018) groepeerde verschillende implementaties van onboarding in acht UI/UX patronen. Deze worden hieronder besproken.

Welkomstberichten

Wanneer je je net registreerde of wanneer je een nieuwe app open doet op een mobiel toestel heb je zeker al eens een welkomstbericht ontvangen. Negen op tien onboarding-ervaringen starten met een bericht waarin ze de nieuwe klant verwelkomen. Vaak wordt dit bericht ondersteund door een CTA-knop (Call-To-Action) waarin men bijvoorbeeld de gebruiker kan uitnodigen om deel te nemen aan een rondleiding doorheen de applicatie.

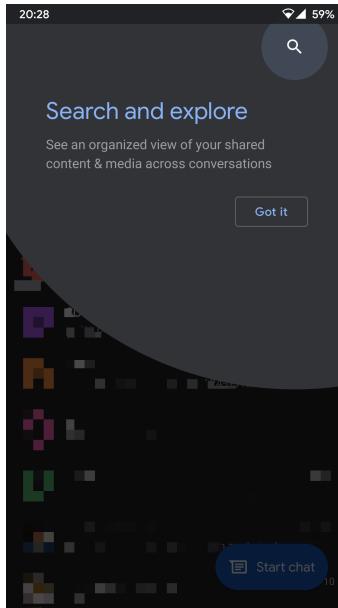


Figuur 2.13: Een welkomstbericht na registratie bij Headspace

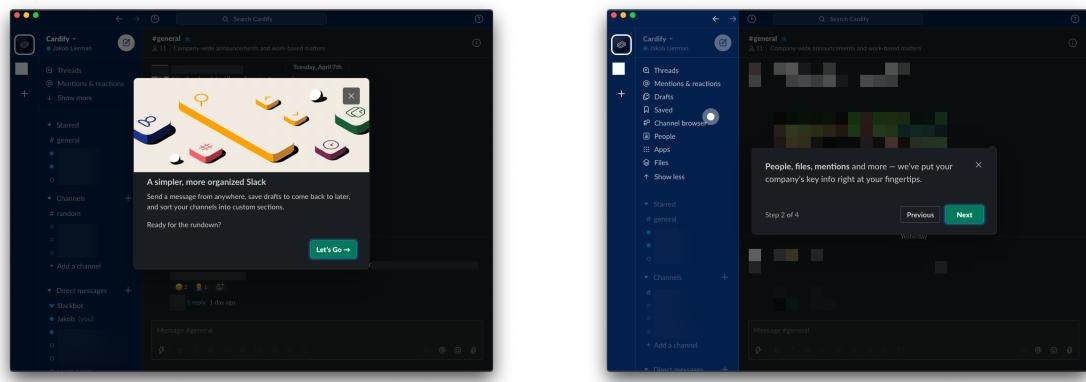
Rondleidingen

Een rondleiding doorheen de applicatie toont alle belangrijke elementen en functionaliteiten even aan met een beknopte uitleg. De rondleiding moet zich echter beperken tot de belangrijkste onderdelen zodat de eindgebruiker zich niet verveeld of geïrriteerd voelt. Bij de rondleiding maakt men veelal gebruik van tooltips (figuur 2.14). De rondleiding wordt voornamelijk gebruikt bij progressieve onboarding (hoofdstuk 2.5.2).

Bij grote applicaties of software met een aanzienlijke hoeveelheid aan functies worden de rondleidingen gewoonlijk opgesplitst. Zo leert de gebruiker stap voor stap alle functionaliteit. Wanneer een applicatie een nieuwe functionaliteit toevoegt of een deel van de UI wijzigt door middel van een update wordt er vaak een nieuwe rondleiding aangeboden. Zo werd de communicatie-tool Slack in maart 2020 geüpdateet en werd er voor bestaande gebruikers opnieuw een rondleiding doorheen de wijzigingen gegeven (figuur 2.15).



Figuur 2.14: Een tooltip om de zoekfunctie te benadrukken in de Messages applicatie van Google



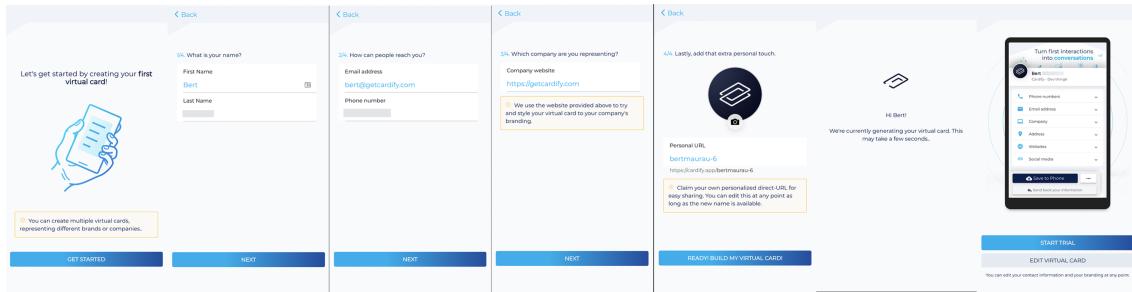
(a) Melding over de update

(b) Slack duidt de belangrijkste wijzigingen aan

Figuur 2.15: Slack (<https://slack.com/>) gaf bestaande gebruikers de kans om nieuwe functies te ontdekken door middel van een rondleiding na een update van de applicatie

Voortgangsindicatoren

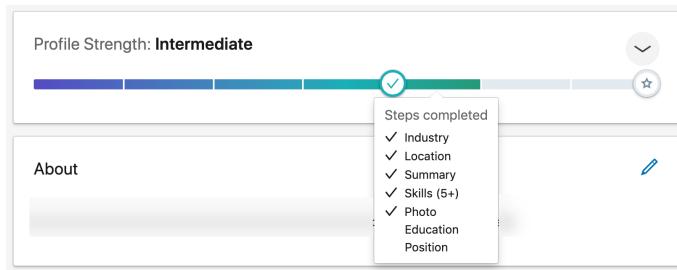
De mens wil opdrachten voltooien. Zeker wanneer men de voortgang kan opvolgen. Doorheen het volledige onboarding-proces kan de voortgang dus weergegeven worden in voortgangsbalken of voortgangsstippen (figuur 2.16). Wanneer een onboarding lang zou worden is dit een goede manier om de gebruiker aan te tonen wanneer deze flow ten einde komt, deze techniek is ook terug te vinden bij Cardify en applicaties ontwikkeld door In The Pocket.



Figuur 2.16: Weergave van resterende stappen bij de registratie in de Cardify applicatie

Checklists

Checklists zetten net zoals voortgangsindicatoren aan tot het voltooien van opdrachten. Het afwerken van een item in de checklist geeft de gebruiker een gevoel van voldoening. Checklists worden naar gewoonte ingezet voor complexere taken met meerdere stappen.



Figuur 2.17: Checklist bij het vervolledigen van een profiel op LinkedIn

Hotspots

Hotspots zijn gelijkaardig aan de tooltips uit de rondleidingen (hoofdstuk 2.5.3). De tooltips zijn niet vrijblijvend, een gebruiker die snel aan de slag wil zal hier dus snel door klikken zonder veel informatie opgenomen te hebben. Hotspots spelen hierop in door zones te creëren waarin de gebruiker kan klikken om meer informatie te verkrijgen over dat UI-element of die functionaliteit. Doordat deze hotspots vrijblijvend zijn verhinderen ze de workflow van de gebruiker niet.

Actie-gedreven tooltips

Actie-gedreven tooltips hebben veel weg van de tooltips uit hoofdstuk 2.5.3. Het verschil zit hem in de interactie ermee. Een eenvoudige tooltip bevat een omschrijving of uitleg en een knop om de tooltip te sluiten of om naar de volgende te gaan in de rondleiding. Bij actie-gedreven tooltips moet men een kleine opdracht uitvoeren, zoals op een knop klikken of tekst invoeren. De bedoeling hiervan is dat men bijleert hoe de applicatie in zijn werk gaat door deze ook echt te gebruiken. In onderzoek van Oliveira (2019) toont hij zelfs aan dat deze tooltips een beter effect hebben dan de eenvoudige tooltips van een rondleiding.



Figuur 2.18: Hotspots bij ingebruikname van Dropbox Paper

Zoals Seppe vermeldde in het interview gebruikt ook Cardify actie-gedreven tooltips. Deze worden voornamelijk ingezet in de webapplicaties omdat deze complexere functionaliteiten bevatten.

Dit kan ook ingezet worden in een bedrijfsomgeving. Een nieuwe werknemer die nog moet leren werken met de vele functionaliteiten van (de software van) een machine zal met deze hands-on mentaliteit veel vlotter zelfstandig aan de slag kunnen gaan.

Uitgestelde accountcreatie

Uitgestelde accountcreatie stelt, zoals je uit de benaming al kon afleiden, de accountcreatie uit. Hierbij worden eerst de functionaliteiten van de software aangehaald voordat de gebruiker zicht hoeft te registreren. Dit beperkt het aantal velden dat de gebruiker moet invullen bij het eerste contact met de applicatie. De gebruiker moet nadien wel in de juiste richting geduwd worden zodat deze toch de registratie voltooid.

Gepersonaliseerde onboarding

Gepersonaliseerde onboarding zorgt ervoor dat de gebruiker bij de eerste ingebruikname van de applicatie meteen een gepersonaliseerde software voor handen heeft. Dit kan door bijvoorbeeld in een kennismakingsformulier de interesses van de gebruiker op te vragen zodat men enkel relevante inhoud krijgt.

Ook Cardify speelt hierop in. De gebruiker krijgt in de finale stap van de onboarding zijn of haar kaart te bezichtigen. Deze kaart werd al gepersonaliseerd gebruik makend van de informatie die men heeft kunnen verkrijgen uit de vorige stappen van de onboarding en registratie. Het is de bedoeling hiermee een 'aha-moment' te creëren bij de gebruiker en deze als het ware te verrassen.

41%



Time to create a profile!

Create a profile to save your progress and
continue learning for free.

CREATE PROFILE

LATER

Figuur 2.19: Duolingo vereist geen account voor de werking van de applicatie, er worden op bepaalde momenten wel meldingen gegeven van de voordelen van een account



Figuur 2.20: Een vragenlijst om de verdere ervaring bij Headspace te personaliseren

2.5.4 Testen van onboarding

Zoals Strahm e.a. (2018) verduidelijkten, is de creatie van een onboardingsflow verschillend voor elke applicatie. Het ontwerpen zelf is een iteratief proces waarbij men afwisselt tussen implementatie en testen. Hier komt laboratory testing zeker van pas (hoofdstuk 2.3.1).

De onboarding wordt net zoals andere UX en UI-elementen getest door middel van usa-

bility testing (hoofdstuk 2.3). Hierbij kan rekening gehouden worden met de snelheid waarop een gebruiker een taak kan uitvoeren met behulp van uitleg uit onboarding-technieken. Ook moet gekeken worden naar het gevoel dat de gebruiker heeft bij deze onboarding. Indien bijvoorbeeld het grootste deel van de testpersonen te snel door de rondleiding klikt is deze hoogstwaarschijnlijk te lang. Door te werken met het “think aloud” protocol en achteraf gerichte vragen te stellen aan de testpersoon kan men snel inzichten creëren over hoe men deze onboarding-technieken kan verbeteren.

Quentin liet weten dat men bij In The Pocket deze testen vaak uitvoert met uitgeprinte mockups of een klikbaar prototype (bijlage B). Zo merken ze snel wanneer een flow onlogisch is of wanneer de inhoud van een bepaald scherm onduidelijk is voor de testgebruiker. Verder in het ontwikkelproces wordt de applicatie vaak uitgebracht als bèta-software. In de bèta-software test het gehele team (of zelfs alle medewerkers van het kantoor) de applicatie. Ook zo komen er soms nog problemen naar boven die men moeilijk of niet zou gevonden hebben bij een test van een half uurtje. Het voordeel van bèta-software is dat men ook verschillende varianten van de applicatie kan uitbrengen en nadien bekijken welke variant de meest positieve impact had.

Cardify werkt als start-up anders dan In The Pocket. Bij Cardify start men met de implementatie van alle functionaliteiten en haalt men de belangrijkste nog eens aan in de onboarding. Als nadien uit usability testing of uit interne data blijkt dat de gebruiker moeilijkheden ondervindt bij andere functionaliteiten, dan worden deze nog in de onboarding verwerkt.

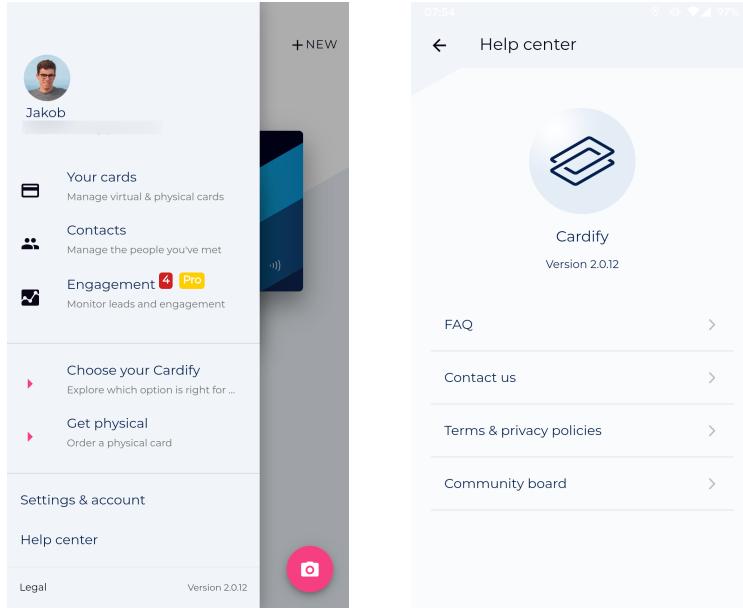
Wanneer de software al in productie is kan men de onboarding blijven verfijnen zoals men bij de rest van de functionaliteiten zou doen. Testtechnieken zoals session recording spelen hierbij een belangrijke rol (hoofdstuk 2.3.2).

2.6 In-app training

Eenmaal de onboarding voorbij is, kan de gebruiker uiteraard terugvallen op de in-app training. Dit zorgt ervoor dat de gebruiker na de onboarding niet in de steek wordt gelaten. In-app user training kan meerdere vormen aannemen. De meest gekende is een help-sectie in de applicatie of online. Hier kan men antwoorden vinden op veelgestelde vragen. Vaak is er ook een formulier aanwezig om complexere vragen op te sturen.

Een help-sectie kan echter wel voor verwarring zorgen bij de gebruiker. Wanneer men UX-problemen opmerkt, zouden deze moeten opgelost worden met UX-elementen. In het interview met Quentin benadrukte hij dat men bij applicaties die ontwikkeld worden door In The Pocket een help-sectie koste wat het kost wil vermijden. Klanten van In The Pocket zetten vaak nog een FAQ op hun website. Ook bij Cardify heeft men dit voorzien. Een verwijzing naar de FAQ op de website is te vinden in het navigatiemenu in de mobiele applicatie (figuur 2.21).

Tooltips (hoofdstuk 2.5.3) komen doorheen de applicatie terug. Omdat te veel tooltips



(a) Help in de navigatie

(b) Help submenu

Figuur 2.21: Help bij Cardify, te vinden op <https://help.getcardify.com/>

bij de ingebruikname van de applicatie geen goed idee zijn, verspreidt men deze over de verschillende functionaliteiten van de applicatie. Zo leert de gebruiker gaandeweg meer over de verschillende mogelijkheden.

Vaak is er ook de mogelijkheid om (een deel van) de onboarding opnieuw te starten. Zo kan de gebruiker altijd terugvallen op de initiële uitleg over de functionaliteit waarvan de gebruiker de werking is vergeten.

2.7 De verschillen tussen UX en UI

User Interfaces (UIs) omvatten de “looks” van de applicatie. Een UI designer vult zijn of haar dag dus met het ontwerpen en schetsen van de interface en verscheidene elementen daarvan (Lamprecht, 2019).

UX is focused on the users journey to solve a problem, UI is focused on how a products surfaces look and function.

Ken Norton - Partner at Google Ventures, Ex-Product Manager at Google

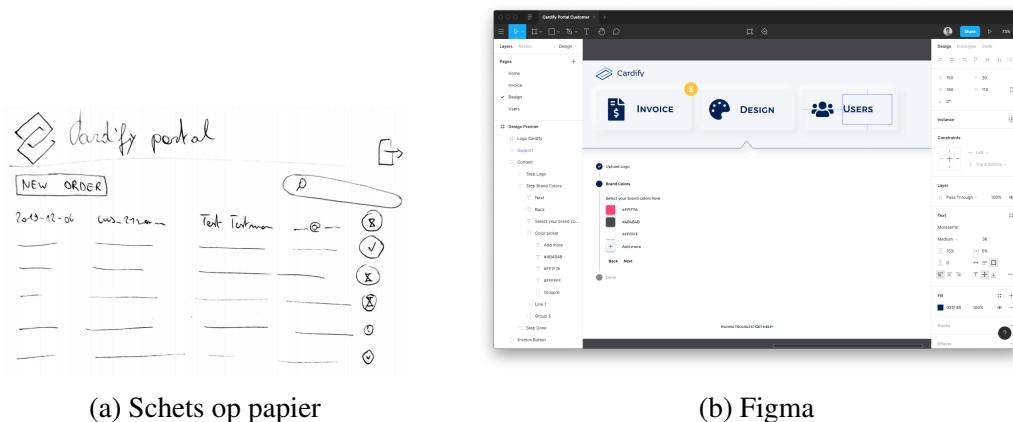
Terwijl het bij UX design steeds om het vinden en oplossen van gebruikssproblemen gaat, heeft UI design daar weinig mee te maken. Bij UI design ligt de focus op het creëren

van esthetisch mooie interfaces. Het ontwerpen van UIs wordt meestal vooraf gegaan door het UX ontwerpproces. Terwijl UX van toepassing is op alle soorten producten en diensten, is UI enkel van toepassing op digitale producten.

User Experience (UX) en User Interface (UI) gaan hand in hand. Zonder een goed uitgewerkt UX design mag de UI nog zo goed zijn, de gebruikers zullen steeds problemen ondervinden bij het gebruik van de applicatie. Het omgekeerde geldt natuurlijk ook.

2.7.1 Software voor UI design

Ontwerpen van interfaces kan vele vormen aannemen. Van een schets op papier tot een prototype waarin men zelfs enkele navigatie elementen kan nabootsen. Softwarepakketten populair bij UI designers bevatten momenteel Adobe Photoshop, Sketch en Figma.



Figuur 2.22: UI design van Cardify Office

3. Methodologie

In dit hoofdstuk zal besproken worden hoe het experiment in zijn werk gaat, hoe de steekproef werd getrokken alsook welke variabelen er gemeten werden.

3.1 Het experiment

Om te bekijken of een implementatie van bepaalde learnability technieken invloed heeft op de eindgebruiker zullen er usability tests worden uitgevoerd bij testpersonen. Welke deze testpersonen zijn en hoe er participanten werden verzameld is te raadplegen in hoofdstuk 3.1.1.

Bij dit experiment zal er vooral gefocust worden op laboratory testing (hoofdstuk 2.3.1). Dit wil zeggen dat de testpersoon de applicatie zal testen in het bijzijn van een moderator, maar niet strikt noodzakelijk in de omgeving waarin de applicatie in een reëel scenario gebruikt zou worden. De moderator noteert hierbij alle bevindingen van de testpersoon, alsook waar deze eventueel moeilijkheden ondervindt.

Het experiment zal uitgevoerd worden bij twee groepen. De ene groep krijgt een proof-of-concept applicatie waarin alle learnability technieken in verwerkt zijn. De controlegroep krijgt een applicatie zonder deze learnability elementen. Tijdens deze usability tests wordt van de testpersonen verwacht dat deze een reeks taken tot een goed einde proberen te brengen. Gedurende het experiment worden een aantal variabelen gemeten (hoofdstuk 3.1.2). De procedure die gevuld wordt in dit onderzoek is te raadplegen in hoofdstuk 3.3.

3.1.1 De populatie en steekproef

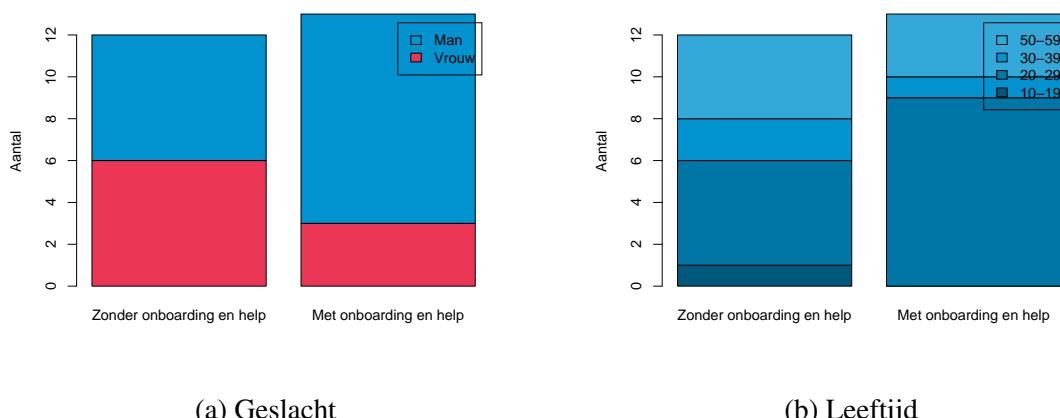
Aan dit experiment kan quasi iedereen deelnemen die een basiskennis heeft over een smartphone. Leeftijd, afkomst, achtergrond, kennis en andere factoren zijn van weinig belang. Er moet echter wel voldoende variatie zijn. Wanneer de groep participanten enkel uit jonge personen met een technische achtergrond bestaat zal het resultaat sterk beïnvloed worden. Het is dus de bedoeling om zowel jong, oud, technisch onderlegd en technisch leek op te nemen in de poule van participanten.

Participanten worden gezocht zowel in de kring van familie en vrienden als daarbuiten. De resultaten van de laboratory tests met deze participanten gingen normaal aangevuld worden met resultaten uit guerilla tests (hoofdstuk 2.3.2). Deze scriptie werd opgesteld tijdens de Covid-19-situatie en het was dus niet mogelijk en/of veilig om guerilla testing uit te voeren.

Omdat uit veiligheidsoverwegingen geopteerd werd om de test op afstand uit te voeren (de moderator is dus niet fysiek aanwezig bij de testpersoon) moet de testpersoon de proof-of-concept applicatie installeren op hun eigen toestel. De applicatie is geschreven voor het mobiele besturingssysteem van Apple (iOS 13 of hoger) en werd geoptimaliseerd voor gebruik op een smartphone (iPhone of iPod touch). Bij het verzamelen van gegevens van potentiële participanten werd er gevraagd indien deze in het bezit zijn van een toestel dat aan deze voorwaarden voldoet.

Een deelnameformulier werd via verschillende kanalen (sociale netwerken, mond-totmondreclame, ...) verspreid om zoveel mogelijk potentiële participanten te bereiken. Het deelnameformulier werd opgesteld in Microsoft Forms. Het volledige formulier is bijgevoegd in bijlage E.

In totaal werd het formulier 45 keer ingevuld. Daaruit werden 25 personen geselecteerd om deel te nemen aan de test. In figuur 3.1 is te zien in welke leeftijds groep de participanten zich bevinden en welk geslacht deze hebben.



Figuur 3.1: Beschrijving van de steekproef

3.1.2 Gemeten variabelen

Als eerste worden enkele demografische variabelen gemeten, namelijk:

- Geslacht
- Leeftijd

De afhankelijke variabelen die worden gemeten zijn:

- Gebruikstijd bij bepaalde opdrachten
- Slaagkans bij bepaalde opdrachten
- De SUS-score
- Gebruikstijd bij herhaling van bepaalde opdrachten
- Slaagkans bij herhaling van bepaalde opdrachten
- Een kwalitatieve variabele met betrekking tot de voorkeur voor één van beide applicaties
- Een kwalitatieve variabele met betrekking tot het blijven gebruiken van (houden van) de applicatie

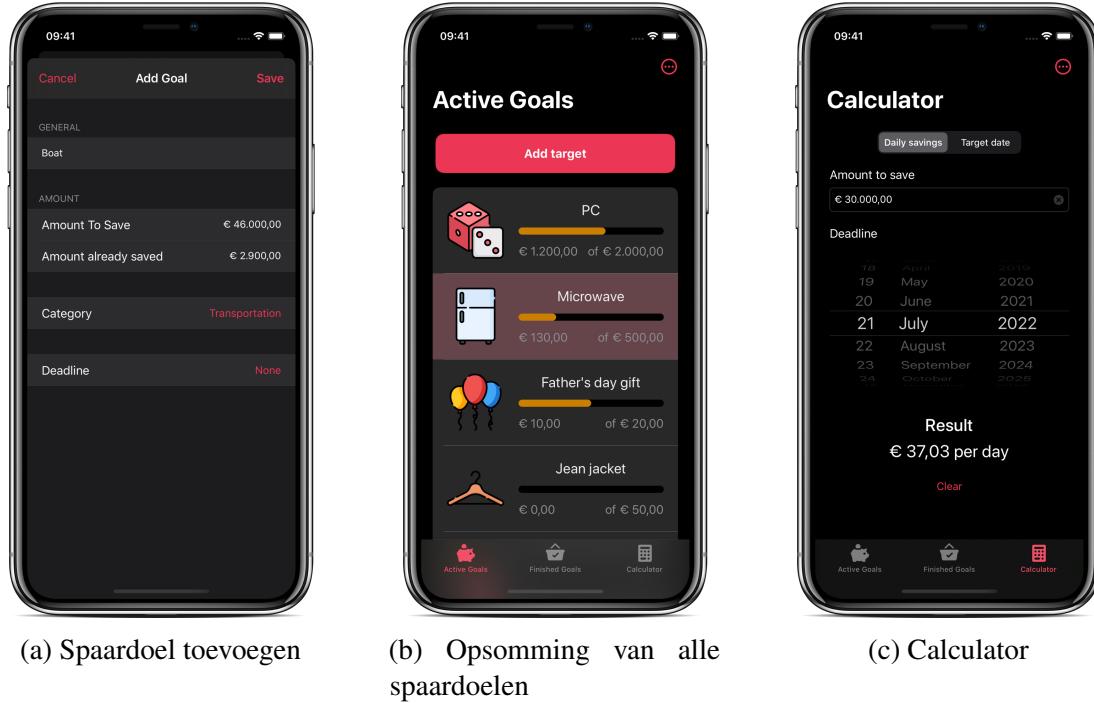
3.2 De proof-of-concept applicatie

De proof-of-concept applicatie is een tool waarmee je je huidige spaardoelen kan bijhouden. Van zodra je de applicatie opent kan je een spaardoel toevoegen (figuur 3.2a), je kan dit eventueel aanvullen met een bedrag dat je al gespaard hebt, een passende categorie en een deadline. Verder bestaat de applicatie uit drie onderdelen; je huidige spaardoelen, je voltooide spaardoelen en een handige calculator. Je huidige en voltooide spaardoelen worden weergegeven in een overzichtelijke lijst (figuur 3.2b). Met de calculator kan je snel berekenen hoeveel je op dagelijkse basis moet sparen om een bepaalde deadline te halen of wanneer je je spaardoel zal behalen als je dagelijks een bepaald bedrag opzij legt (figuur 3.2c).

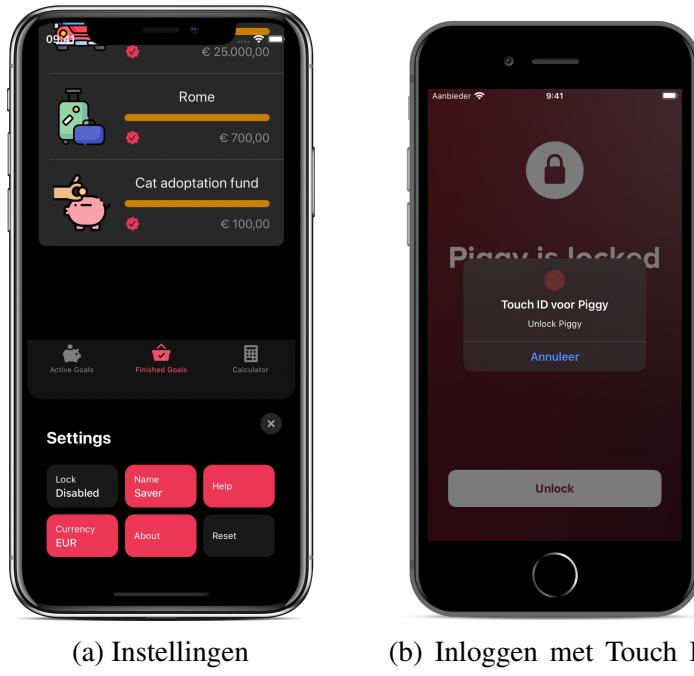
Eenmaal je een spaardoel hebt toegevoegd, kan je telkens een bedrag toevoegen dat je al bijeen gespaard hebt. Dit bedrag wordt bij het totaal opgeteld. Het is mogelijk dit snel te doen door middel van een slider die tot 50 gaat of je kan een extra popover openen waarmee je gedetailleerdeerder je bedrag kan specificeren.

Er zijn ook nog enkele snufjes die iets verder in de menu's verborgen zitten, zoals de mogelijkheid om de applicatie te vergrendelen met een pincode (figuur 3.3b). Bij de instellingen is het verder ook mogelijk de munteenheid te wijzigen die doorheen de applicatie gehanteerd wordt. Indien je terug van nul wil starten kan je daar ook alle gegevens van de applicatie wissen (figuur 3.3a).

De applicatie heeft de naam Piggy gekregen, een verwijzing naar piggy bank, de Engelse vertaling voor spaarvarken. Hierbij is ook een bijhorend logo ontworpen (figuur 3.4). Doorheen de applicatie maakt men verder gebruik van een roze tint die verwijst naar de kleur van het varkentje verwerkt in het logo. Er werd gekozen voor een alternatieve, gele



Figuur 3.2: Basisfunctionaliteiten van de proof-of-concept applicatie



Figuur 3.3: Uitgebreide opties van de proof-of-concept applicatie

tint voor elementen die moeten opvallen.



Figuur 3.4: Icoon van Piggy, de proof-of-concept applicatie

3.2.1 De learnability elementen

Zoals vooraf aangehaald in hoofdstuk 3.1 zullen er twee varianten van de proof-of-concept applicatie gemaakt worden. De ene variant bevat alle functionaliteiten maar hierbij is geen uitleg voorzien in de vorm van onboarding en/of in-app training.

De andere applicatie is functioneel niet verschillend aan de applicatie zonder learnability elementen. In deze applicatie worden echter wel enkele aanpassingen gedaan om de learnability van de applicatie te verbeteren in de hoop een positieve invloed te hebben op het klantbehoud. Bij het openen van deze applicatie wordt de gebruiker gegroet met een welkomstbericht en start er direct een rondleiding aan de hand van tooltips die voorzien zijn van uitleg (hoofdstukken 2.5.3, 2.5.3 en figuur 3.5). Wanneer de gebruiker in de applicatie meer uitleg wenst kan hij/zij de rondleiding van bepaalde onderdelen herstarten door het vraagteken-icoon aan te klikken in de navigatiebalk. Als laatste redmiddel is er ook een uitgebreide help-sectie voorzien (hoofdstuk 2.6 en figuur 3.6).

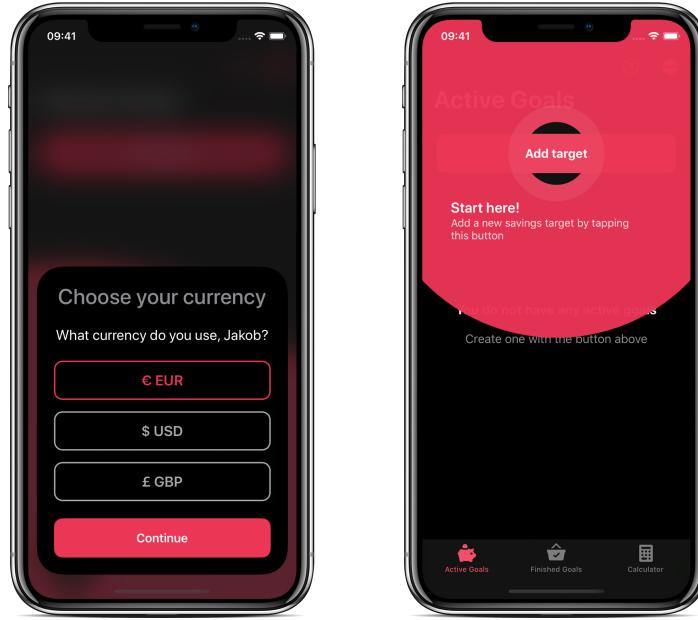
Deze onboarding en hulp doorheen de applicatie beperkt zich tot de essentiële onderdelen ervan. De gebruiker kan vlot starten met het gebruik van de applicatie. Gaandeweg komen er nog enkele tooltips tevoorschijn om zo de aandacht te vestigen op een onderdeel dat de gebruiker net ontdekt heeft (bijvoorbeeld: de gebruiker wil een nieuw spaardoel creëren en de bijhorende interface komt tevoorschijn) (hoofdstuk 2.5.2). Er werd bij het uitdenken van de onboarding aandacht gehecht aan de vijf basisprincipes uit hoofdstuk 2.5.1.

Deze onboarding is een voorbeeld van een functiegerichte onboarding. De focus ligt voornamelijk bij het uitleggen van functionaliteiten aan de gebruiker zodat deze vlot met de applicatie overweg kan (hoofdstuk 2.5.2).

Vooraleer er werd gestart aan de usability tests werd deze applicatie grondig getest op onduidelijkheden doorheen de onboarding. Een testversie van de applicatie werd voorgelegd aan enkele gebruikers. Zij konden onduidelijkheden doorheen de software aantonen.

3.2.2 Ontwikkeling

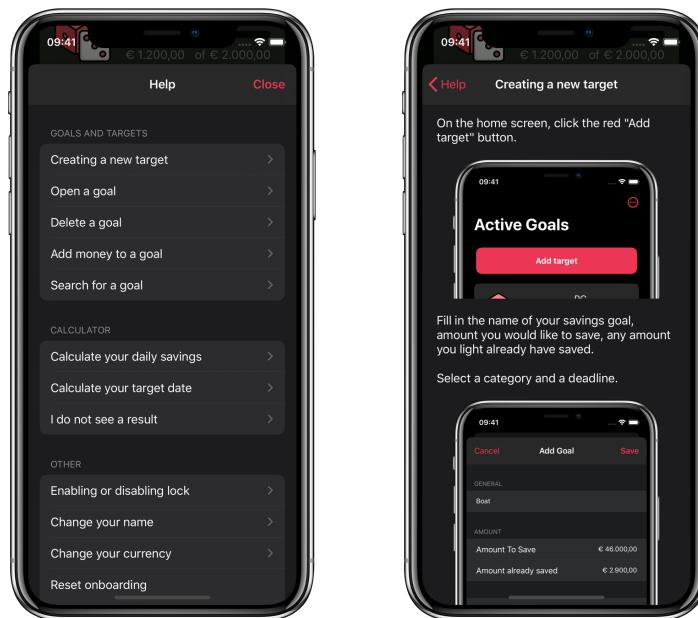
Er werd gekozen om de applicatie te schrijven voor het mobiele platform van Apple (iOS) omdat veel gebruikers met grote diversiteit hiermee bekend zijn. De applicatie werd geschreven in de programmeertaal Swift 5 en draait op iOS 13 of hoger.



(a) Hulp bij het starten met het gebruik wanneer de applicatie voor de eerste maal start

(b) De rondleiding toont alle belangrijke functionaliteiten

Figuur 3.5: Een rondleiding doorheen de proof-of-concept applicatie



(a) Een lijst met alle onderwerpen

(b) Een bepaalde functie grondig uitgelegd

Figuur 3.6: Uitgebreide help-sectie in de proof-of-concept applicatie

De volledige broncode van de applicatie is te raadplegen op <https://github.com/JakobLierman/piggy>.

3.3 De test afnemen

Vooraleer de test wordt afgenoem, vulden de participanten al enkele vragen in. Enerzijds gingen deze vragen over of ze al dan niet in aanmerking kwamen om deel te nemen aan dit onderzoek, anderzijds gingen deze vragen over hun demografische gegevens (bijlage E).

De standaardprocedure voor het afnemen van deze test gaat als volgt: aan de participanten wordt via een videogesprek de procedure uitgelegd. Afhankelijk van de groep waarvan de participant deel uitmaakt, krijgt deze de applicatie met of zonder de learnability elementen. De participanten voeren achtereenvolgens een aantal opdrachten uit om de usability van de applicatie in kaart te brengen. Tijdens deze opdrachten observeert de moderator het gedrag van de participanten. Na afloop vullen ze de SUS vragenlijst in (bijlage D).

Tenslotte wordt de opzet van het onderzoek uitgelegd. Hierna krijgt de participant nog even tijd om de andere applicatie uit te testen (afhankelijk van in welke groep de participant zich bevindt). Op basis van deze twee ervaringen mag de participant aangeven welke van de twee applicaties hij of zij prefereert.

4. Over de metingen

4.1 De gebruikstijd bij bepaalde opdrachten

Eén van de gemeten variabelen was de gebruikstijd bij bepaalde opdrachten. Er werd gemeten hoe lang de participant erover deed om bepaalde opdrachten te voltooien. De opdrachten waren bij alle participanten gelijk:

1. Enkele instellingen wijzigen
2. Een nieuw spaardoel toevoegen
3. Een bedrag toevoegen aan een spaardoel
4. Een reeds voltooid spaardoel verwijderen
5. Een berekening met de ingebouwde calculator
6. Een groot bedrag toevoegen aan een spaardoel

De laatste opdracht werd pas na een korte onderbreking uitgevoerd. Deze opdracht is in essentie een herhaling van opdracht 3 (een bedrag toevoegen aan een spaardoel). Door deze achteraf nog eens uit te voeren werd getest of de participant de werking van die functionaliteit nog steeds onder de knie had.

De timer werd gepauzeerd wanneer de participant doorheen de onboarding ging. Ook wanneer er een help-element tevoorschijn kwam werd deze timer even stopgezet. Participanten kregen geen hulp van de moderator. De timer werd stopgezet nadat de participant voltooid had of wanneer de participant na een hele poos zijn of haar zoektocht opgaf.

In tabellen 4.1 en 4.2 worden de gemiddelden en standaardafwijkingen van de gemeten tijden weergegeven. Hier kan men waarnemen dat er eventueel een verschil zal zijn in tijden tussen de groep die de opdrachten zonder onboarding en help-elementen voltooij-

	<i>Tijd in seconden</i>						
	<i>Min</i>	<i>Q₁</i>	<i>Mdn</i>	<i>M</i>	<i>SD</i>	<i>Q₃</i>	<i>Max</i>
Instellingen	7.00	15.00	19.50	24.08	12.87	31.25	45.00
Spaardoel toevoegen	27.00	39.00	42.50	48.83	20.70	52.50	106.00
Bedrag toevoegen	12.00	24.50	30.00	35.67	18.52	44.00	71.00
Spaardoel verwijderen	11.00	14.25	39.50	65.25	57.46	121.50	148.00
Berekening	22.00	34.75	38.00	40.33	17.5	41.50	91.00
Groot bedrag toevoegen	12.00	25.50	34.00	47.25	35.76	50.75	137.00

Tabel 4.1: Beschrijving van tijden zonder het gebruik van onboarding en help-elementen

	<i>Tijd in seconden</i>						
	<i>Min</i>	<i>Q₁</i>	<i>Mdn</i>	<i>M</i>	<i>SD</i>	<i>Q₃</i>	<i>Max</i>
Instellingen	5.00	12.00	15.00	15.31	7.39	15.00	30.00
Spaardoel toevoegen	23.00	26.00	31.00	34.92	13.77	40.00	75.00
Bedrag toevoegen	15.00	16.00	20.00	22.62	8.03	30.00	40.00
Spaardoel verwijderen	6.00	8.00	15.00	16.38	11.94	20.00	48.00
Berekening	12.00	20.00	25.00	26.15	9.33	30.00	42.00
Groot bedrag toevoegen	10.00	12.00	17.00	24.77	27.88	26.00	115.00

Tabel 4.2: Beschrijving van tijden met het gebruik van onboarding en help-elementen

den en de groep die de opdrachten met voltooiden. Het verschil wordt in hoofdstuk 5.1 bewezen.

4.2 Het vragen om hulp

Niet elke participant kon elke opdracht succesvol voltooien. In figuur 4.1 wordt procentueel weergegeven indien men hulp nodig had bij het voltooien van de opdrachten. Er wordt een onderverdeling gemaakt indien men al dan niet gebruik kon maken van onboarding en help-elementen.

Uit figuur 4.1 kan men afleiden dat er slechts voor sommige opdrachten hulp gevraagd werd. De andere opdrachten waren voor alle participanten duidelijk, ongeacht de groep waarin ze zich bevonden.

4.3 Gebruik van de voorziene functionaliteiten

Men kan op verschillende manieren een bedrag toevoegen aan een spaardoel (figuur 4.2). De snelste manier is door een slider te verschuiven tot het bedrag correct is en dan op de grote knop “Add money” te drukken. Deze slider gaat slechts tot een maximum van 50. Wanneer de gebruiker een hoger bedrag wil toevoegen moet deze op de knop met het “+”-symbool drukken. Dit werd uitgelegd in de rondleiding doorheen de applicatie.

In figuur 4.3 wordt procentueel weergegeven hoeveel participanten gebruik hebben gemaakt van het “+”-symbool bij de opdrachten die vereisten om een bedrag toe te voegen.

4.4 De SUS-score

Aan iedere participant werd gevraagd om de SUS-vragenlijst in te vullen (bijlage D). De gemiddelde resultaten hiervan worden weergegeven in tabel 4.3. Hier kan men waarnemen dat het eventuele verschil in SUS-score zich tot een minimum zal beperken.

	<i>Min</i>	<i>Q₁</i>	<i>Mdn</i>	<i>M</i>	<i>SD</i>	<i>Q₃</i>	<i>Max</i>
Zonder	62.50	71.25	78.75	78.54	10.08	85.00	92.50
Met	57.50	62.50	85.00	78.65	16.13	95.00	100.00

Tabel 4.3: Beschrijving van de SUS-score zonder of met learnability-elementen

Zoals vermeld in hoofdstuk 2.3.4 bevat de SUS-vragenlijst enkele vragen die testen op learnability in plaats van op usability. Het gaat hier om vraag 4 (*Ik denk dat ik de steun van een technisch persoon nodig heb om dit systeem te kunnen gebruiken*) en vraag 10 (*Ik moest veel dingen leren voordat ik met dit systeem aan de slag kon*). De antwoorden op deze vragen worden visueel weergegeven in figuur 4.4. Een beschrijving van deze antwoorden wordt weergegeven in tabellen 4.4 en 4.5. Hierbij stellen de waardes 1, 2, 3, 4 en 5 respectievelijk *volledig oneens*, *oneens*, *neutraal*, *eens* en *volledig eens* voor.

	<i>Min</i>	<i>Q₁</i>	<i>Mdn</i>	<i>M</i>	<i>SD</i>	<i>Q₃</i>	<i>Max</i>
Vraag 4	1.00	1.00	1.50	1.67	0.78	2.00	3.00
Vraag 10	1.00	1.00	2.00	1.83	0.85	2.00	4.00

Tabel 4.4: Beschrijving van SUS-vragen omtrent learnability zonder onboarding en help-elementen

	<i>Min</i>	<i>Q₁</i>	<i>Mdn</i>	<i>M</i>	<i>SD</i>	<i>Q₃</i>	<i>Max</i>
Vraag 4	1.00	1.00	1.00	1.31	0.83	1.00	4.00
Vraag 10	1.00	1.00	1.00	1.62	1.04	2.00	4.00

Tabel 4.5: Beschrijving van SUS-vragen omtrent learnability met onboarding en help-elementen

4.5 Voorkeur voor onboarding en help-elementen

De participanten werden na afloop gevraagd of ze de proof-of-concept applicatie lieft gebruikten indien de onboarding en help-elementen aanwezig waren. De resultaten van deze vraag worden procentueel weergegeven in figuur 4.5.

Participanten die de applicatie prefereren zonder de onboarding en help-elementen maken deze keuze omdat de applicatie voor hun eenvoudig en duidelijk genoeg was. Indien de

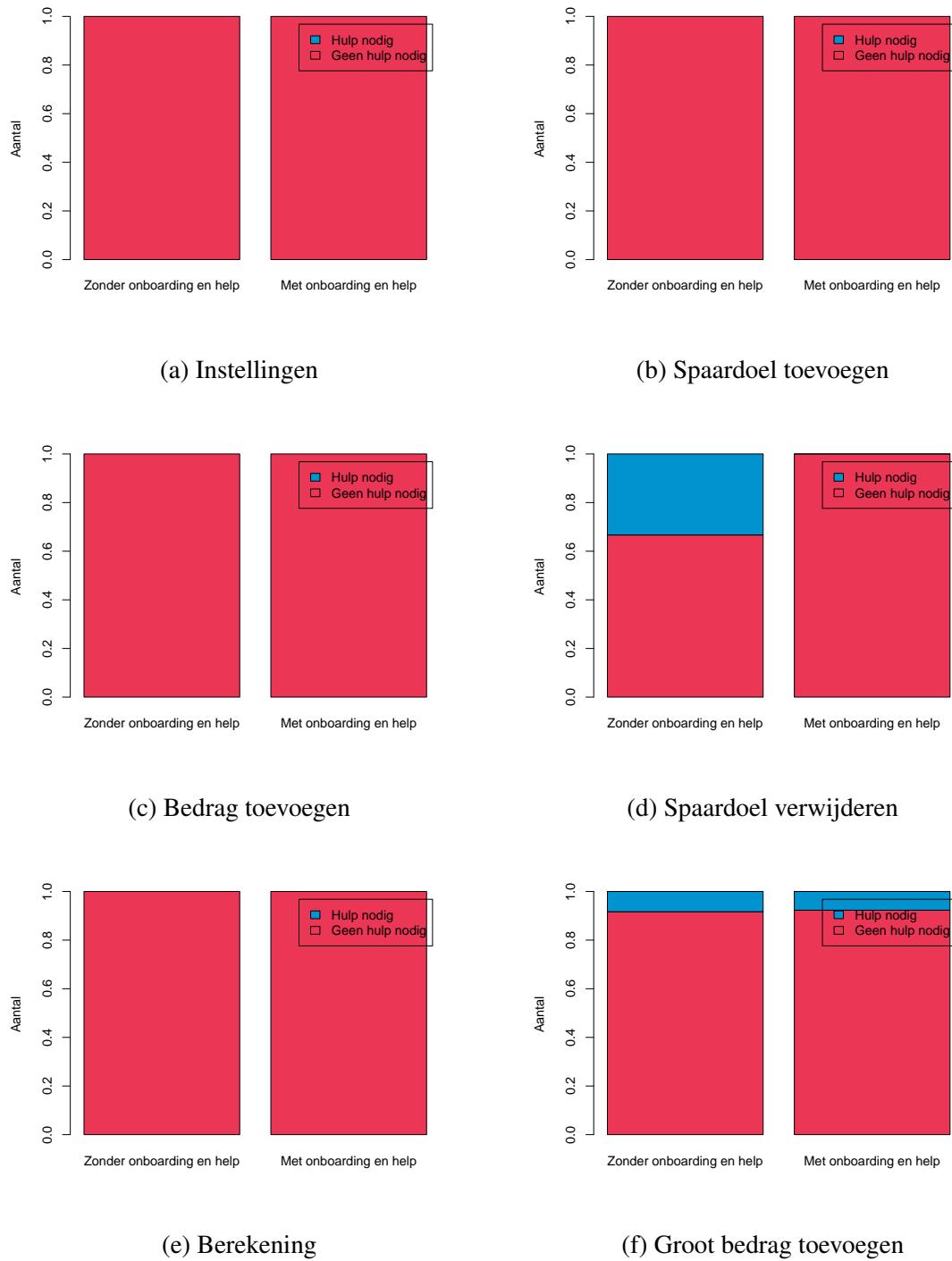
applicatie enkele meer ingewikkelde functionaliteiten zou hebben, zouden ook zij liever geholpen worden doorheen de applicatie.

4.6 Voorkeur voor help-sectie

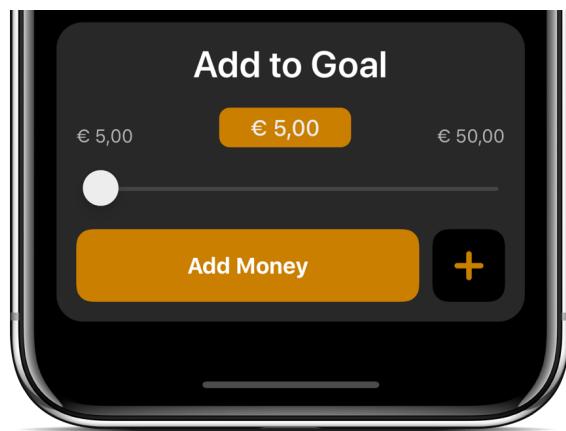
Er werd de participanten een voorbeeld getoond van de help-sectie in de applicatie (figuur 3.6). Hierover werd aan de participant gevraagd of deze gebruik zou maken van een gelijkaardige help-sectie in deze of andere applicaties. De resultaten van deze vraag worden procentueel weergegeven in figuur 4.6.

4.7 De gebruiksduur en/of levensduur van de applicatie

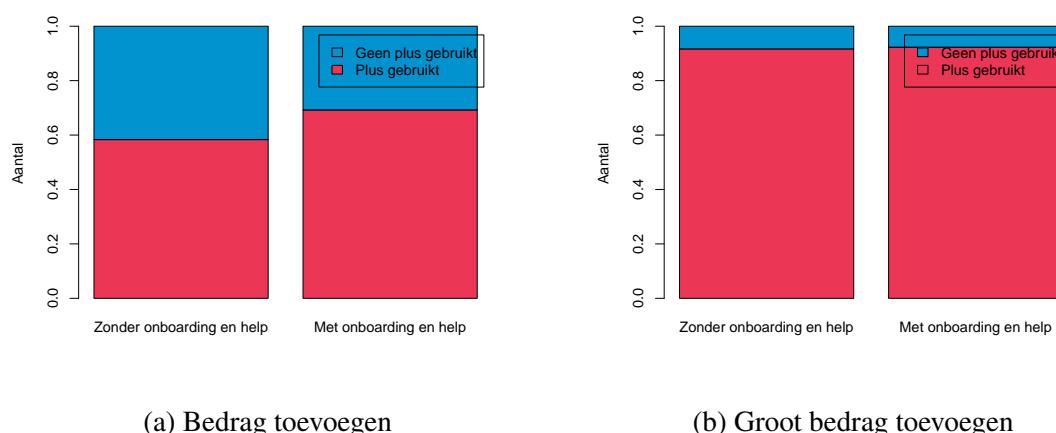
Om te meten of de onboarding en help-elementen al dan niet invloed hebben op de gebruiksduur en/of levensduur van de applicatie werd de participant gevraagd of deze de applicatie zou houden. Andere mogelijkheden waren dat de applicatie niet goed was, niet duidelijk was of dat men liever nog eens vergelijkt met andere beschikbare applicaties met gelijkaardige functionaliteiten. Wanneer de gebruiker voor één van deze mogelijkheden zou kiezen, zou hij of zij de applicatie dus verwijderen. De resultaten van deze vraag worden procentueel weergegeven in figuur 4.7.



Figuur 4.1: Had de participant hulp nodig bij het voltooien van de opdrachten?



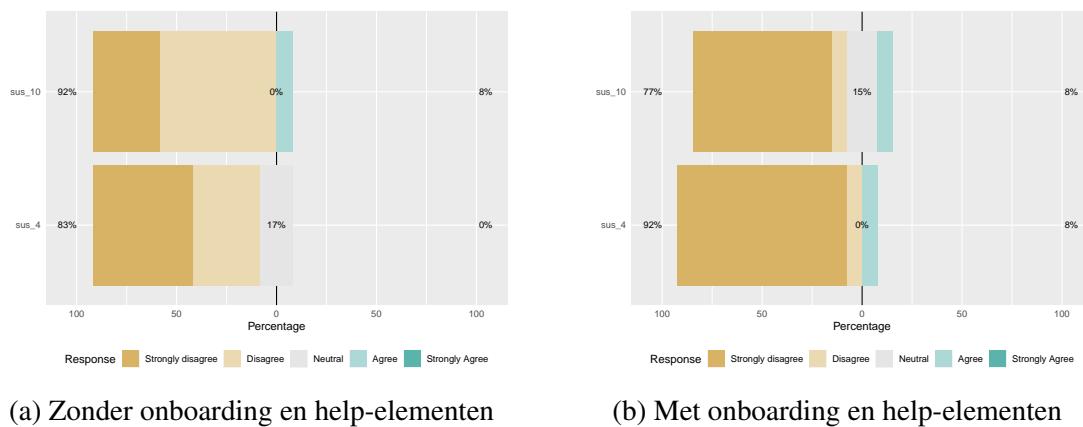
Figuur 4.2: Verschillende manieren om een bedrag aan een spaardoel toe te voegen



(a) Bedrag toevoegen

(b) Groot bedrag toevoegen

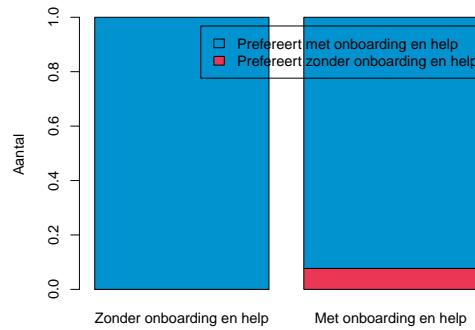
Figuur 4.3: Of de participant het “+”-symbool gebruikt bij het toevoegen van een bedrag



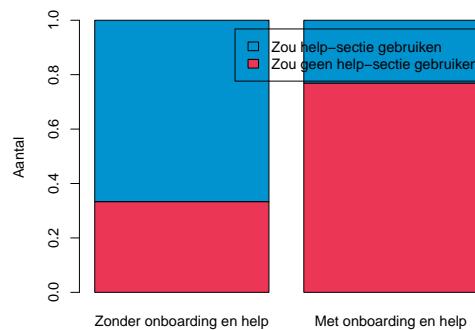
(a) Zonder onboarding en help-elementen

(b) Met onboarding en help-elementen

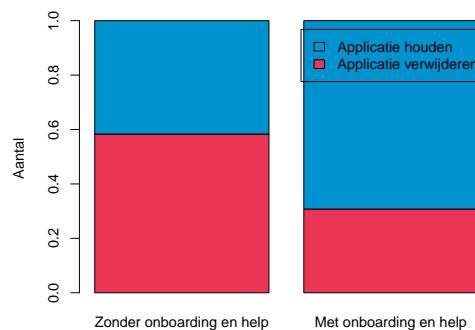
Figuur 4.4: Antwoorden op vraag 4 en 10 van de SUS-vragenlijst



Figuur 4.5: De voorkeur van de participant: een applicatie met of zonder onboarding en help-elementen



Figuur 4.6: Ziet de participant zichzelf de help-sectie gebruiken



Figuur 4.7: Of de participant de applicatie zou houden

5. Resultaten van de proef

5.1 Onderzoeksvraag 1: Kan een (betere) onboarding en in-app user training ervoor zorgen dat de eindgebruiker een beter inzicht heeft op de totale functionaliteit van een grote applicatie?

Om een antwoord te formuleren op de eerste onderzoeksvraag kan men een aantal afhankelijke variabelen in overweging nemen. Enerzijds kan men kijken naar alle tijden op de zes taken, anderzijds naar de SUS-score en de vragen met betrekking tot learnability uit de SUS-vragenlijst, maar men kan ook in overweging nemen of de participant al dan niet om hulp vroeg tijdens één van de taken.

Deze laatste variabelen, namelijk de zes dichotome hulp-gevraagd variabelen, werden onderzocht met behulp van een χ^2 -test (chi-kwadraat-test). In deze test werd de groep waarvan de participant deel uitmaakt als onafhankelijke variabele opgenomen. De resultaten zijn waar te nemen in tabel 5.1. Voor de opdrachten *instellingen, spaardoel toevoegen, bedrag toevoegen en berekening* hebben alle participanten het doel bereikt zonder hulp.

Uit de resultaten van de χ^2 -test valt af te leiden dat er een significant verband is tussen de twee variabelen bij de opdracht *spaardoel verwijderen*, $\chi^2(1, N = 25) = 5.16$, $p = 0.02$,

	χ^2	df	p	N
Spaardoel verwijderen	5.16	1	0.02	25
Groot bedrag toevoegen	0.003	1	0.95	25

Tabel 5.1: χ^2 resultaten indien de participant hulp nodig had

Opdracht	<i>t</i>	<i>df</i>	<i>p</i>
Instellingen	-8.55	24.10	< 0.001
Spaardoel toevoegen	-11.11	24.04	< 0.001
Bedrag toevoegen	-9.27	24.05	< 0.001
Spaardoel verwijderen	-4.19	24.01	< 0.001
Berekening	-10.55	24.05	< 0.001
Groot bedrag toevoegen	-5.27	24.01	< 0.001
<i>SUS-score</i>	-29.36	24.07	< 0.001

Tabel 5.2: *t*-testen van alle opdrachten

maar niet bij de opdracht *groot bedrag toevoegen*, $\chi^2(1, N = 25) = 0.003$, $p = 0.95$, ns. Bij enkele functionaliteiten zal de gebruiker dus minder hulp nodig hebben wanneer men bepaalde vormen van onboarding implementeert in de applicatie. Echter zal de gebruiker niet bij elke functionaliteit een voordeel halen uit de onboarding. Om te weten wanneer de onboarding van pas komt kan men best de applicatie laten testen door enkele personen en noteren waar deze problemen ondervinden (hoofdstuk 2.3).

Om de overige afhankelijke variabelen op te nemen in de analyses, werd eerst een *t*-test uitgevoerd. In deze *t*-test werd opnieuw als onafhankelijke variabele de groep waarvan de participant deel uitmaakt opgenomen. Als afhankelijke variabele werd gekozen voor een gemiddelde van de tijden op alle taken. Participanten voltooiden de opdrachten sneller wanneer deze de proof-of-concept applicatie hadden waarbij de onboarding en help-elementen beschikbaar waren ($M = 23.36$, $SD = 9.95$) in vergelijking met wanneer deze geen onboarding en help-elementen ter beschikking hadden ($M = 43.57$, $SD = 21.76$), $t(24.033) = -8.426$, $p < .001$.

Deze analyse toont een verschil aan tussen wanneer men wel of geen gebruik kon maken van de learnability-elementen, maar de assumptie wordt gemaakt dat het effect van learnability-elementen op de zes taken gelijkaardig genoeg is dat er een gemiddelde van kan worden genomen. Omwille van deze reden wordt voor elke taak apart nog eens een *t*-test uitgevoerd. De resultaten van deze *t*-test zijn waar te nemen in tabel 5.2. Per opdracht is ook hier duidelijk dat participanten die gebruik maakten van onboarding en help-elementen significant minder tijd nodig hadden om de opdracht te voltooien. De gemiddelde tijden zijn te vinden in tabellen 4.1 en 4.2.

De berekening van deze waarden in R is bijgevoegd in bijlage F.

Een betere onboarding kan er zeker en vast voor zorgen dat de eindgebruiker sneller met de applicatie overweg kan. Hoe goed deze applicatie is in de ogen van de gebruiker hangt echter van meerdere variabelen af. Zo is een matige applicatie met een goede learnability niet rechtstreeks een betere applicatie. Waar en wanneer er in-app help elementen moeten geïmplementeerd worden, hangt sterk af van de gebruiker. Wat vaak werd opgemerkt bij het afnemen van deze proef is dat elke gebruiker verschillend is en de ene gebruiker een bepaalde functionaliteit begrijpt zonder hulp terwijl de andere gebruiker sterk leunt op de hulp. Bij het bouwen van een applicatie moet dus zeker rekening gehouden worden met het doelpubliek bij het implementeren van onboarding en help-elementen. Usability tests

voor en na die implementatie zijn sterk aan te raden voor betere inzichten.

5.2 Onderzoeksvraag 2: Hoe een grote hoeveelheid aan functionaliteiten beheersbaar houden voor de eindgebruiker?

Het beheersbaar houden van een aanzienlijke hoeveelheid aan functionaliteiten is niet enkel een taak voor de onboarding en andere learnability elementen. Dit is een breder probleem waarvoor men idealiter een verder onderzoek in het UX-vakgebied houdt. Zoals in hoofdstuk 2.4, 2.5 en 2.6 te lezen valt, is het voorzien van bepaalde onboarding en help-elementen wel een stap in de goede richting. Bij het afnemen van de proef viel ook op te merken dat voor de herhaling van de opdracht waar men een bedrag moest toevoegen de participant zelfverzekerder was wanneer deze de proof-of-concept applicatie had waarin de learnability-elementen aanwezig waren. Zo waren minder twijfels merkbaar.

Waarop zeker gefocust moet worden bij de implementatie van learnability-elementen in software van een aanzienlijke omvang is dat men de gebruiker initieel niet overspoelt met uitleg (hoofdstuk 2.5.1). De gebruiker verliest zo snel de aandacht en zal niet alles kunnen onthouden. Hotspots (hoofdstuk 2.5.3) zijn zo vaak een betere keuze dan een rondleiding (hoofdstuk 2.5.3), omdat de gebruiker zo zelf kiest wanneer deze wil starten met de uitleg.

Men kan de beheersbaarheid meten met de vragen uit de SUS-vragenlijst die betrekking hebben tot learnability. Participanten antwoorden gunstiger op deze vragen wanneer deze de proof-of-concept applicatie hadden waarbij de onboarding en help-elementen beschikbaar waren (vraag 4: $M = 1.31$, $SD = 0.85$, vraag 10: $M = 1.62$, $SD = 1.04$) in vergelijking met wanneer deze geen onboarding en help-elementen ter beschikking hadden (vraag 4: $M = 1.67$, $SD = 0.78$, vraag 10: $M = 1.83$, $SD = 0.83$), vraag 4: $t(40.07) = -4.96$, $p < .001$, vraag 10: $t(37.09) = -5.63$, $p < .001$.

De berekening van deze waarden in R is bijgevoegd in bijlage G.

5.3 Onderzoeksvraag 3: Heeft (het gebrek aan) in-app user training effect op de gebruiksduur en/of levensduur van de applicatie?

Of de gebruiker de applicatie voor een langdurige periode zou gebruiken is moeilijk te voorspellen. De belangrijkste fase voor de gebruiksduur is het initieel gebruik. Wanneer de gebruiker niet overtuigd is van de applicatie zal hij of zij die sneller terug verwijderen. Als afsluitende vraag in de proef werd aan de participanten gevraagd als ze deze applicatie zouden houden op hun toestel of ze deze zouden verwijderen of vergelijken met concurrerende software.

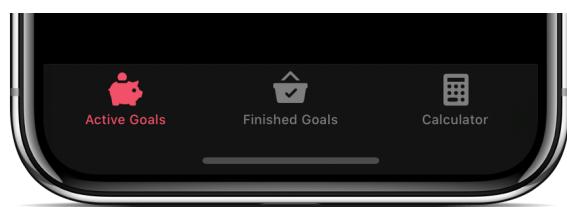
In figuur 4.7 is te merken dat er eventueel een verschil zal zijn tussen de groep met en zonder onboarding en usability elementen. Na uitvoering van de χ^2 -test is er echter

geen significante relatie merkbaar tussen het al dan niet aanwezig zijn van learnability-elementen en indien de participant de applicatie zou houden op zijn of haar persoonlijk toestel, $\chi^2(1, N = 25) = 1.92, p = 0.17, ns$. De gebruiker die toegang had tot onboarding en help-elementen was dus niet meer geneigd de applicatie te houden dan de gebruiker die hiertoe geen toegang had. Een verklaring hiervoor kan zijn dat een hele hoop meer factoren hierop inspelen. Deze proef is bijvoorbeeld uitgevoerd bij participanten waarvan niet iedereen op zoek was naar een applicatie met deze functionaliteit. Verder onderzoek kan hier meer duidelijkheid en een gunstiger resultaat bieden.

De berekening van deze waarden in R is bijgevoegd in bijlage H.

5.4 Onderzoeksraag 4: Hoe de eindgebruiker wegwijs maken in een grote applicatie?

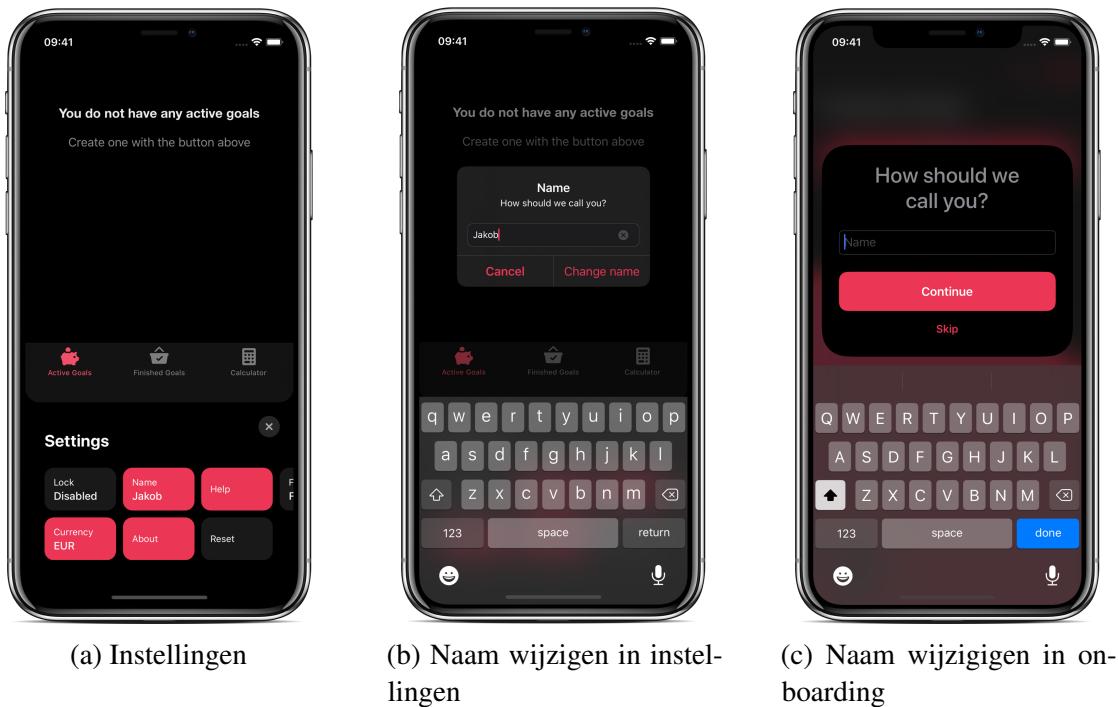
Hoe een gebruiker omgaat met de functionaliteiten binnen de applicatie is één ding, maar de gebruiker moet uiteraard ook overweg kunnen met de navigatie binnen deze applicatie om zo bij de gewenste functionaliteiten uit te komen. Net zoals bij onderzoeksraag 2 (hoofdstuk 5.2) is dit een breder probleem uit het UX-vakgebied. Participanten uit deze proef hebben wel aangetoond dat deze het handig en bruikbaar vonden dat bij de initiële opstart van de applicatie er een uitleg was voorzien voor de elementen in de tab-bar (figuur 5.1). Een uitleg voorzien zorgt ervoor dat de iconen uit de navigatie een duidelijke betekenis krijgen. Hierdoor gaat de navigatie vlotter.



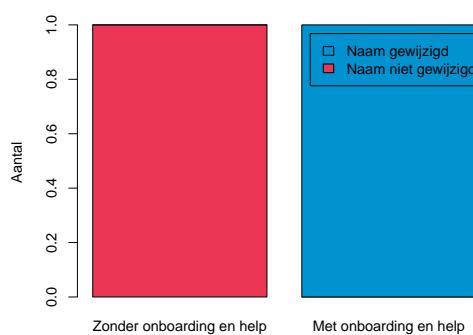
Figuur 5.1: Navigatie in de proof-of-concept applicatie

Om de gebruiker niet aan zijn of haar lot over te laten is het mogelijk om deze in de onboarding al enkele taken te laten voltooien vooraleer aan het echte werk te beginnen. In de proof-of-concept applicatie was het mogelijk om een naam in te stellen, dit kon zowel in de instellingen als in de onboarding (figuur 5.2). In de proef viel op dat enkel participanten die de proof-of-concept applicatie met onboarding hadden de naam ook effectief wijzigden (figuur 5.3). De relatie tussen het instellen van een naam en de aanwezigheid onboarding-elementen is significant, $\chi^2(1, N = 25) = 25, p = < .001$. De gebruiker is dus meer geneigd bepaalde opdrachten te voltooien wanneer hij of zij deze de kans krijgt om dit in de onboarding van de applicatie te doen.

De berekening van deze waarden in R is bijgevoegd in bijlage I.



Figuur 5.2: De naam van de gebruiker wijzigen in de proof-of-concept applicatie



Figuur 5.3: Of de participant de naam wijzigde bij het verkennen van de applicatie

6. Conclusie

In dit onderzoek werd een antwoord gegeven op de onderzoeksfrage “Kan een (betere) onboarding en in-app user training ervoor zorgen dat de eindgebruiker een beter inzicht heeft op de totale functionaliteit van een grote applicatie?”. Hiervoor werd een proof-of-concept applicatie opgesteld met en zonder learnability-elementen. Deze werd getest door twee groepen participanten.

De resultaten van deze proef gaven aan dat de learnability-elementen zeker hun nut hebben. Echter moet men zeer voorzichtig omspringen met waar en wanneer men deze plaatst. Zoals vooraf werd besproken is het belangrijk om zich te beperken tot de essentie, een gebruiker leert de details van de applicatie pas achteraf kennen. Zo wordt de gebruiker niet meteen overspoeld met informatie. Frequent de applicatie laten testen om zo te bekijken waar er hulp moet worden, is één van de belangrijkste stappen bij het implementeren van learnability in software.

Er bestaan verschillende technieken om onboarding op een correcte manier te implementeren. Niet elke techniek kan voor elke applicatie of use case gebruikt worden. De ontwikkelaar van de software bekijkt best welke technieken geschikt zijn voor zijn of haar specifieke use case. Deze verschillen in UX voor verschillende applicaties kunnen evenwel onderzocht worden in een verder onderzoek.

De verspreiding van help-elementen doorheen de applicatie is een volgend belangrijk aspect bij de implementatie van onboarding en help-elementen. Door de gebruiker niet initieel te overspoelen met info, maar deze pas geleidelijk aan nieuwe functionaliteiten aan te leren zal de gebruiker de werking van deze functionaliteiten beter onthouden. De gebruiker zal ook minder snel de aandacht verliezen dan tijdens bijvoorbeeld één grote rondleiding.

Er werd voor deze applicatie geen significante relatie gevonden tussen (het gebrek aan) in-app user training en de gebruiksduur en/of levensduur van de applicatie. Wanneer een bedrijf hun software test bij gebruikers die ook effectief een meerwaarde hebben aan de functionaliteiten van deze software kan dit resultaat met hoogste waarschijnlijkheid variëren. Dit kan eventueel bewezen worden in een verder onderzoek.

Een interessante ontdekking uit deze scriptie is het feit dat het mogelijk is de gebruiker aan te zetten tot het gebruik van bepaalde functionaliteiten. Door deze functionaliteiten aan te halen in de onboarding is de gebruiker meer geneigd deze daar uit te proberen en verder te gebruiken. Dit kan interessant zijn om bijvoorbeeld accountcreatie in de kijker te zetten.

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Technologie neemt overal de overhand. Meer leeftijdsgroepen maken gebruik van web- en mobiele applicaties. Deze softwarepakketten worden voor alle mogelijke doeleinden gebruikt, denk maar aan een management platform in een bedrijf, of een mobiel spel om tegen je vrienden te quizzen.

Deze applicaties moeten op een bepaalde manier aan de eindgebruiker uitgelegd worden. Er bestaan verscheidene technieken om alle functionaliteiten binnen de software aan de man te brengen. Boeken deze technieken wel het gewenste resultaat? Is de ene techniek beter dan de andere?

In deze bachelorproef onderzoeken we de onboarding en in-app user training aan de hand van volgende onderzoeksvragen en deelvragen:

- Kan een (betere) onboarding en in-app user training ervoor zorgen dat de eindgebruiker een beter inzicht heeft op de totale functionaliteit van een grote applicatie?
- Hoe een grote hoeveelheid aan functionaliteiten beheersbaar houden voor de eindgebruiker?
- Heeft (het gebrek aan) in-app user training effect op de gebruiksduur en/of levensduur van de applicatie?
- Hoe de eindgebruiker wegwijs maken in een grote applicatie?

A.2 Literatuurstudie

A.2.1 Wat is 'onboarding'?

Er bestaan al verscheidene technieken die mogelijk geïmplementeerd kunnen worden in een applicatie om het de gebruiker eenvoudiger te maken deze applicatie te gebruiken. Eén van de bekendste technieken om een gebruiker wegwijs te maken binnen een softwarepakket is 'onboarding'. Onboarding vindt plaats wanneer u de software voor de eerste maal opstart. Het geeft de eindgebruiker een duidelijk inzicht van hoe de applicatie werkt, hoe de functionaliteiten in zijn werk gaan en welke voordelen er worden aangeboden (Cardoso, 2017).

A.2.2 Voorgaand onderzoek

Bestaand onderzoek bestudeert een manier van onboarding en in-app user training door middel van een 'Learning Center' (Camacho Herrero, 2019). Hierbij maakt men gebruik van een in-app onderwijsfunctie die de eindgebruiker alle mogelijke functionaliteiten van de software kan uitleggen. Dit platform wil de gebruiker meer betrekken en aanmoedigen om alle elementen te voltooien door middel van gamification elementen zoals checklists of voortgangsbalken. Per onderdeel zelf werd grotendeels gebruik gemaakt van video's die alle uitleg voorzien. In deze bachelorproef zal verder onderzocht worden als het voordeleger is om dit platform op te splitsen in kleine stukken per functionaliteit. Pas wanneer de gebruiker in contact komt met bepaalde functionaliteiten zal de hij de in-app training te zien krijgen.

Onboarding is vaak ook meer dan enkel uitleg over de applicatie. Ook elementen zoals de registratie zijn een stap in het proces om met de software overweg te kunnen (Renz e.a., 2014). Ook een welkomstmail is een veelgebruikt middel om de gebruiker vertrouwd te maken met de applicatie.

Onboarding is verschillend voor webapplicaties als mobiele applicaties (Ramirez Alvarez, 2018). In beide gevallen moet grondig onderzocht worden wat de gebruiker wil bereiken en het gedrag van deze gebruikersgroep is. Applicaties gericht op een jeugdig publiek zullen de onboarding anders aanpakken dan softwarepakketten gericht op het vervullen van een taak in een bedrijfsomgeving.

A.2.3 UX testing

Voorgaand onderzoek maakt duidelijk dat er veel voorbereiding nodig is op vlak van user experience (UX) en usability testing. Om te weten wat de eindgebruiker wil bereiken en welke acties die daarvoor zal ondernemen is er genoeg inlevingsvermogen nodig (Gualtieri, 2009). Ook mag men er nooit vanuit gaan dat de ontwikkelaar het bij het rechte eind heeft. Hij is immers niet de eindgebruiker en bekijkt de probleemstelling op een andere manier.

A.3 Methodologie

Het onderzoeken van technieken zoals in-app user training en bepaalde manieren van het implementeren van deze technieken zal vooral gebeuren aan de hand van experimenten in het kader van usability testing.

Hierbij zal gebruik gemaakt worden van enkele varianten van dummy-applicaties (het zij met of zonder gebruik van bovenstaande technieken). Gebruikers van de dummy-applicatie zullen fysiek de software doorlopen en enkele opdrachten voltooien. Aan de hand van metingen op basis van een aantal criteria zal kunnen worden besloten of er over een succesvolle test mag gesproken worden. Deze criteria zijn onder meer, maar niet beperkt tot; het succespercentage, het aantal fouten per tijdseenheid, de gemiddelde tijd en de subjectieve tevredenheid van de gebruiker.

De bovengenoemde dummy-applicaties zullen mobiele en/of webapplicaties zijn waarin al dan niet de te onderzoeken technieken geïmplementeerd zullen zijn. Om het onderzoek zo breed mogelijk te houden zal er zowel een applicatie zijn die een software uit de industriële wereld nabootst (denk aan het besturen van een machine, het beheren van bestellingen, ...) als een applicatie voor alledaags gebruik (denk aan een boodschappenlijstje, sociaal netwerk, ...).

Het te onderzoeken doelpubliek zal ook zo breed mogelijk gehouden worden aangezien iedereen ooit wel eens in aanraking komt met software. Het doelpubliek zal nadien opgesplitst worden in doelgroepen op basis van leeftijd en technische kennis.

De gebruikers krijgen enkele opdrachten die ze tot een goed einde moeten brengen op de te testen applicatie. Aan de hand van usability testing zal gemeten worden hoe kwaliteitsvol de software was. De belangrijkste criteria bij deze meting zijn het succespercentage, het aantal fouten per tijdseenheid, de gemiddelde tijd en de subjectieve tevredenheid van de gebruiker. Deze opdrachten zullen formatief verlopen. Er zal dus een moderator/observator aanwezig zijn die noteert hoe de gebruiker de opdrachten uitvoert. Na deze opdrachten zal de gebruiker nog een vragenlijst invullen om zo de SUS-score (System Usability Scale) te berekenen.

A.4 Verwachte resultaten

Ik verwacht dat de eindgebruiker er gemiddeld langer over zal doen om te wennen aan bepaalde functionaliteiten bij een gebrek aan in-app user training of onboarding. Ook verwacht ik dat een onboarding waarbij alle functionaliteiten aan bod komen een negatieve impact geeft in vergelijking met in-app user training per functionaliteit wanneer de gebruiker hiermee in contact komt.

A.5 Verwachte conclusies

Het nut van een goede onboarding zal bewezen worden. Alsook mag de onboarding enkel betrekking hebben op de meest essentiële functionaliteit.

In-app user training zal ervoor zorgen dat de eindgebruiker het gevoel heeft de functionaliteit onder de knie te hebben. Daardoor zal de gebruiker minder vlug afstand nemen van de applicatie. Ook zal het eenvoudiger zijn voor gebruikers met een beperkte technische achtergrond om gebruik te maken van de applicatie.

B. Interview In The Pocket

B.1 Interview informatie

Datum: 2 mei 2020.

Respondent: Quentin Braet (Lead Solution Architect bij In The Pocket)

B.2 Hoe start je bij de ontwikkeling van een nieuwe applicatie aan de onboarding?

Neem je de onboarding op in het designproces?

Heel veel hangt af van het project. We gaan altijd eerst samen met de klant op zoek naar de requirements. Hiervoor starten we van de business requirements waaruit we zowel de functionele als niet functionele requirements afleiden. Dit geldt ook voor de onboarding van een app. We vertrekken van de doelen (requirements) en kijken wat we nodig hebben om die te bereiken. Veel hangt af van hoe we gebruikers authenticeren en welke vereisten de klant daarbij heeft.

Bij de Payconiq by Bancontact app hebben we bijvoorbeeld heel zware security en legal constraints, maar ook complexiteit die geïntroduceerd wordt door de partners waarmee we samenwerken. Dat maakt dat we voor een heel grote uitdaging stonden om deze technisch zware onboarding flow toch behapbaar te houden voor de gebruiker.

Ook heb je bij mobile apps vaak meerdere permissies die je moet vragen, ook deze moeten worden meegenomen in het designproces. Je wil de gebruiker hier zoveel mogelijk

begeleiden en informeren, maar het mag ook niet overweldigend zijn, teveel tekst op een scherm is vaak geen goed idee, de copy is vaak een belangrijk aandachtspunt.

Al deze requirements worden bij ons door de product manager en architect opgenomen in de intake gesprekken. Eens we een duidelijk beeld hiervan hebben, gaan we hiermee aan de slag om samen met een designer de eerste mockups te maken. Deze worden dan afgecheckt op allerlei vlakken, ook met de klant.

B.3 Hoe test je het gehele onboarding proces binnen ITP?

Test je direct bij echte gebruikers of is een rondje rond kantoor reeds voldoende?

Onze mensen zijn getraind op interfaces ontwerpen, maar doordat wij zelf vaak power users zijn, is het niet altijd evident om te zien waar minder digitaal aangelegde gebruikers moeite mee gaan hebben. Dat is waarom wij vaak gaan “user testen”, zeker voor grote projecten waar een foute implementatie van een complexe onboarding flow grote gevolgen kan hebben.

Bij user test leggen we de mockups voor aan mensen die we extern recruten. Vaak een heel gevarieerd publiek, maar aan een 5 tot 10 tal mensen heb je vaak genoeg om de belangrijkste issues te spotten. Hierbij leggen we ze ofwel een printout van de mockups voor, maar vaak kijken we of we toch een clickable prototype kunnen maken om het geheel echter te laten aanvoelen. In deze sessies vergaar je vaak heel veel feedback over onlogische flows of copy en icoontjes die onduidelijk zijn.

Maar ook dit dekt niet alles: wanneer gebruikers langs komen voor zo een user test, dan zijn ze daar met een doel om al je opdrachten te voltooien. In het echt haken mensen gewoon af op het moment dat dingen onduidelijk zijn en hebben ze niet altijd de mogelijkheid om extra uitleg te vragen. Vandaar dat ook de data, vaak in de vorm van funnels, een grote rol spelen in het bijsturen van onboarding flows die live staan. Bovenop deze funnels kunnen ook A/B tests gemaakt worden om het effect van kleine aanpassingen te meten om zo tot een optimale flow te komen.

Naast user testing, doen we ook vaak beta testing, waarbij iedereen van het team (inclusief de klant), maar bij uitbreiding ook gans ITP in een beta programma zit voor onze apps. Dit zijn onze zogenaamde friendly users, die de productie app testen voor hij naar het brede publiek gaat. We krijgen nu eenmaal makkelijker feedback van onze collega's dan van mensen die we niet kennen. Eens de beta goedgekeurd is, gaan we over tot een phased rollout, wat betekent dat we geleidelijk aan meer productie gebruikers toegang geven tot de nieuwste versie. In deze periode houden we alle statistieken nog extra in de gaten.

B.4 Welke elementen zijn zeker terug te vinden bij een goede onboarding volgens ITP?

Denk aan: tooltips, welkomstbericht, ...

De boodschap is altijd: keep it simple. Hoe korter de flow hoe beter. Toch is dit niet altijd mogelijk. Een aantal dingen die we zeker altijd doen:

- Bij het vragen van permissies gaan we altijd eerst een scherm tonen die uitlegt waarom deze permissie gebruikt wordt.
- Vaak is er een legal scherm (terms & conditions + privacy disclaimer), vraag de toestemmingen niet stiekem maar heel expliciet.
- Als je onboarding flow toch lang wordt, geef de gebruiker indicaties van hoe ver hij in de flow zit.
- Bij een lange onboarding flow: probeer eventueel de flow toch op te splitsen. Sommige functionaliteiten kan je misschien pas later activeren met een extra stukje onboarding achteraf. Dit helpt de gebruiker om snel een bruikbare app te zien.
- Introduceer geen nieuwe wachtwoorden als dat niet nodig is, gebruik indien mogelijk bestaande accounts.
- Een handleiding bij de app zou niet nodig mogen zijn, maar korte coachmarks kunnen wel helpen. Ideaal is ook om die niet allemaal tegelijk te geven bij het eerste gebruik, maar de gebruiker enkel het belangrijkste mee te geven in het begin, en meer geavanceerde features later pas te introduceren indien dat nog nodig is (wanneer je merkt dat een gebruiker die bijvoorbeeld nog niet gebruikt heeft).
- Maak het visueel, niemand leest graag veel tekst.

B.5 Wanneer merk je dat een gebruiker moeilijkheden ondervindt bij een bepaalde functionaliteit?

Zie je dit aan gedragingen van de gebruiker bij bijvoorbeeld een usability test?

Ook hier hangt het af van welke fase we in zitten.

Ontwerp fase: de eerste stap is dat het team het zelf moeten begrijpen, dat klinkt logisch maar is soms al een enorme uitdaging (denk aan apps als Itsme & Payconiq by Bancontact).

User testing fase: hier let je vooral op het gedrag van de testgebruikers en de vragen die ze stellen.

Development & QA fase: ook hier komt vaak nog wat feedback van het team, door alles dieper te gaan bekijken komen soms nog rare kronkels naar boven of dingen die plots technisch moeilijker blijken dan gedacht.

Beta fase: de periode waarin we feedback krijgen van onze klant en andere collega's, als

blijkt dat bepaalde zaken echt niet werken, kunnen die nog herwerkt worden voor we naar het brede publiek gaan.

Phased rollout: hier monitoren we allerhande statistieken en reviews. Zijn er flows die plots minder gebruikt worden? Komt dit door een technisch probleem? Vinden de mensen een nieuwe feature wel? Moeten we die duidelijker aangeven met een coachmark of een what's new? Indien nodig stoppen we de rollout en wordt het probleem eerst opgelost.

Productie fase: Hier spreekt de data van de onboardingsfunnel het meest, bij welk scherm of bij welke actie haken mensen af? Dit is natuurlijk enkel een indicator, hiermee weet je nog niet noodzakelijk de oorzaak.

B.6 Hoe voorzie je een help sectie bij zeer complexe functionaliteiten?

Volstaat een FAQ sectie? Neem je als het ware het handje van de gebruiker vast en doorloop je alle stappen in de applicatie?

In principe vermijden we die koste wat het kost, help secties zijn vaak maar een lapmiddel om een minder goede UX toch bruikbaar te maken. In principe moet een UI zichzelf uitwijzen. Dit kan met een aantal kleine tricks zoals coachmarks op gepaste momenten (zie hierboven), een informatieve maar subtile banner die tips & tricks geeft... In sommige gevallen beperken we de info op het scherm zelf, waar goede verstaanders in principe genoeg mee hebben, maar zetten we toch ergens een link naar meer uitleg. Veel klanten zetten toch nog een FAQ op hun website en linken er naar via de app, al wordt die vaak aangevuld door informatie van hun first line support. Wij bekijken continu of we enkele van die opmerkingen niet beter kunnen verwerken in de UX van de app. Al hebben al die dingen natuurlijk wel een prijs, en moeten er trade offs gemaakt worden. In principe is alles oplosbaar, maar sommige technische implicaties zijn te kostelijk en die oplossen levert misschien ook maar een beperkte meerwaarde voor de gebruiker. Ook al vinden we dat soms zelf jammer omdat we weten dat het eigenlijk beter zou moeten kunnen, maar het is een constante evenwichtsoefening.

C. Interview Cardify

C.1 Interview informatie

Datum: 4 mei 2020.

Respondent: Seppe Vereecken (Product & Growth bij Cardify)

C.2 Hoe start je bij de ontwikkeling van een nieuwe applicatie aan de onboarding?

Neem je de onboarding op in het designproces?

Bij de ontwikkeling van Cardify start men nog niet met onboarding. Men start eerst met de implementatie van de effectieve functie en achteraf bekijkt men waar en wanneer er onboarding zou voorzien moeten worden.

Omdat Cardify nog steeds een startup is, leren we uit elke ervaring bij. Moest je me deze vraag binnen en jaar opnieuw stellen, zal het antwoord waarschijnlijk volledig anders zijn.

Wat wel onmiddellijk geïmplementeerd werd bij onze app was een flow voor de registratie. Nadat de gebruiker alle gegevens heeft ingevuld, wordt er een kaart getoond. Het is belangrijk dat de gebruiker het product zo snel mogelijk te zien krijgt.

C.3 Hoe test je het gehele onboarding proces binnen Cardify?

Test je direct bij echte gebruikers of is een rondje rond kantoor reeds voldoende?

Wij gebruiken daar vooral “superusers” voor. Nu je het zo vraagt denk ik er aan dat we dat misschien toch iets officiëler moeten maken, zodat we inderdaad een vaste groep gebruikers hebben die zich opgegeven hebben om nieuwe features te testen. In de Birdhouse (co-working kantoor) ging ik soms gewoon rond om features te laten testen door andere startups, maar tot nu toe beperken we ons dus tot ‘friends, family, fools’ en superusers waarvan we weten dat ze engaged zijn.

Wanneer een gebruiker moeite ondervindt met bepaalde functies passen we de applicatie aan in de volgende iteratie. Dat kan gaan van kleine communicatieve aanpassingen, zoals popups of kleine UI veranderingen zoals kleuren of buttons maar dat kan ook als gevolg hebben dat we die feature volledig anders positioneren in de app.

Het uploaden van content was bijvoorbeeld een feature die in het begin minder belangrijk was voor ons en daardoor iets verder weg zat maar dus ook weinig gebruikt werd. We hebben dan een extra tussenscherm gemaakt alvorens ze contactgegevens kunnen aanpassen waardoor het uploaden van content een pak naar voor geschoven werd en nu dus ook meer gebruikt wordt.

Alles bij elkaar wordt er vaak nog “op het gevoel” gewerkt.

C.4 Welke elementen zijn zeker terug te vinden bij een goede onboarding volgens Cardify?

Denk aan: tooltips, welkomstbericht, ...

Bij Cardify is het doel nog steeds om een product/dienst te verkopen. De klant maakt in de onboarding zo snel mogelijk kennis met de Cardify kaart. Deze kaart is na de onboarding ook meteen gepersonaliseerd, dit moet bij de klant een ‘aha-moment’ creëren. We laten de gebruiker ook kennis maken met alles wat mogelijk is, hierdoor leert de gebruiker meteen ook met deze functionaliteiten werken.

Bij Cardify willen we alles zo makkelijk mogelijk maken. Een goede onboarding bestaat dus uit veel visuals en weinig tekst. De gebruiker weet hoogstwaarschijnlijk graag hoever ze in de onboarding zitten, dus daarvoor tonen we voortgangsindicatoren.

C.5 Wanneer merk je dat een gebruiker moeilijkheden ondervindt bij een bepaalde functionaliteit?

Zie je dit aan gedragingen van de gebruiker bij bijvoorbeeld een usability test?

Om initieel een onboarding te testen laten we een gebruiker de applicatie even gebruiken. Wanneer de gebruiker de applicatie sluit stellen we deze enkele vragen. Alsook vragen we achteraf om enkele taken uit te voeren. Wanneer deze mislukken zit er een fout in de onboarding, de gebruiker heeft niet goed geleerd met de app te werken.

In de app zitten ook enkele “verplichte” flows. Dit zijn flows die de gebruiker normaal gezien direct zou moeten doorlopen (zoals een kaart personaliseren en zijn/haar kaart laten scannen). Wanneer van zo’n flow afgeweken werd, is er iets niet duidelijk voor de gebruiker en moeten we daarop anticiperen.

Op ons intern platform kunnen we beperkt monitoren welke functionaliteiten van de applicatie worden gebruikt. Wanneer een functionaliteit niet het gewenste resultaat behaalt spelen we ook hierop in. We plaatsen deze bijvoorbeeld voorop in de navigatie of we “duwen” de gebruiker als het ware naar de functionaliteit.

C.6 Hoe voorzie je een help sectie bij zeer complexe functionaliteiten?

Volstaat een FAQ sectie? Neem je als het ware het handje van de gebruiker vast en doorloop je alle stappen in de applicatie?

Bij Cardify hebben we een beknopt help-center. Hier worden alle functionaliteiten grondig uitgelegd. Uit persoonlijke ervaring weten we wel dat zo’n help-center zelden gebruikt zal worden, daarom proberen we alles zo duidelijk mogelijk te maken in de app zelf.

In de web-platformen zitten vaak iets complexere stukken van het softwarepakket. Deze worden uitgelegd aan de hand van een tour. Deze tour loodst de gebruiker doorheen de belangrijkste flows door middel van actie-gedreven tooltips. Dit willen we echter niet verplichten.

D. System Usability Scale template

System Usability Scale (SUS) Project

Institution Department

Strongly
Disagree Strongly
Agree

1. I think that I would like to use this system frequently 1 2 3 4 5
2. I found the system unnecessarily complex 1 2 3 4 5
3. I thought the system was easy to use 1 2 3 4 5
4. I think that I would need the support of a technical person to be able to use this system 1 2 3 4 5
5. I found the various functions in this system were well integrated 1 2 3 4 5
6. I thought there was too much inconsistency in this system 1 2 3 4 5
7. I would imagine that most people would learn to use this system very quickly 1 2 3 4 5
8. I found the system very cumbersome to use 1 2 3 4 5
9. I felt very confident using the system 1 2 3 4 5
10. I needed to learn a lot of things before I could get going with this system 1 2 3 4 5

Information

The following information describes the detailed data regarding the user tests of the research work. The data will therefore be important to demographic and versioning analysis.

Participant Name

Date

Test

Location

Prototype

Version

Support

List of sponsors, honors and donors:

LOGO

LOGO

LOGO

LOGO

LOGO

LOGO

E. Deelnameformulier

E.1 Inleiding

Inschrijving applicatie testing

Hey!

Bedankt voor je interesse in mijn bachelorproef. Voor mijn onderzoek moet ik enkele usability tests van software uitvoeren. In het kort wil dit dus zeggen dat ik een kleine mobiele applicatie loslaat op echte gebruikers! Na het invullen van deze enquête kom je te weten als je in aanmerking komt voor dit onderzoek.

Wat houdt deze test exact in? Je zal enkele opdrachten moeten uitvoeren op een applicatie die ik zelf ontwikkeld heb, ondertussen bekijk ik aandachtig alle stappen dat je daarvoor onderneemt (no stress, alles wat je doe is juist). Achteraf moet je ook een kleine vragenlijst invullen en stel ik mogelijks nog enkele vragen.

Alvast bedankt!

Vanwege de aanhoudende Covid-19-situatie zal de usability test worden gehouden via een videogesprek.

E.2 Mobieltje toestel

Type smartphone in bezit

Gelieve iPhone"te selecteren als je tijdelijk een iPhone kan lenen van een vriend, huisgenoot, ...

- iPhone
- Android-toestel (Samsung, Oneplus, ...)
Bij keuze van deze optie wordt er direct doorverwezen naar sectie E.3.
- Anders
Bij keuze van deze optie wordt er direct doorverwezen naar sectie E.3.

Huidige iOS versie

U vindt uw versie onder Instellingen > Algemeen > Info > Softwareversie

- iOS 13.x.x
- iOS 12.x.x
- Lager dan iOS 12.x.x

E.3 Over jezelf

Gelieve enkele gegevens over jezelf achter te laten zodat ik jou kan contacteren.

Contactgegevens worden niet gedeeld.

Volledige naam

Dit is een invulveld.

Leeftijdsgroep

- 70+
- 60-69
- 50-59
- 40-49
- 30-39
- 20-29
- 10-19

Geslacht

- Vrouw
- Man
- Non-binair

E-mailadres

Dit is een invulveld.

Telefoonnummer

Dit is een invulveld.

Ik heb een computer met een webcam en microfoon

- Ja
- Nee

Ik identificeer mezelf als technisch onderlegd

Een smartphone heeft voor mij weinig tot geen geheimen. Mijn familie en vrienden komen telkens naar mij met hun computerproblemen. Ik heb interesse in alle digitale snufjes.

- Ja
- Nee

E.4 Afsluiting

Bedankt voor de interesse. Ik probeer zo snel mogelijk contact met je op te nemen. Vergeet ook niet dit formulier om te delen!

Blijf in uw kot en was uw handen.

Jakob Lierman

F. Berekeningen R onderzoeks vragen 1

```
1 # Data
2 dataSet <- read.csv(file = "dataset.csv", sep = ";")
3
4 dataSet$times_mean <- (dataSet$settings_task + dataSet$new_task +
5   dataSet$add_task + dataSet$delete_task + dataSet$calculator_task +
6   dataSet$add_task_repeat) / 6
7
8 subSetWithoutOnboarding <- subset(dataSet, onboarding_elements == FALSE,
9   select = -c(onboarding_elements))
10 subSetWithOnboarding <- subset(dataSet, onboarding_elements == TRUE,
11   select = -c(onboarding_elements))
12
13 # Hulp nodig
14
15 ## Settings Task
16 #chisq.test(x = dataSet$onboarding_elements, y = dataSet$settings_task_
17   help, correct = FALSE)
18
19 ## New Task
20 #chisq.test(x = dataSet$onboarding_elements, y = dataSet$new_task_help,
21   correct = FALSE)
22
23 ## Add Task
24 #chisq.test(x = dataSet$onboarding_elements, y = dataSet$add_task_help,
25   correct = FALSE)
26
27 ## Delete Task
28 chisq.test(x = dataSet$onboarding_elements, y = dataSet$delete_task_
29   help, correct = FALSE)
30
31 ## Calculator Task
```

```

24 #chisq.test(x = dataSet$onboarding_elements , y = dataSet$calculator_
25   task_help , correct = FALSE)
26
27 ## Add Task Repeat
28 chisq.test(x = dataSet$onboarding_elements , y = dataSet$add_task_repeat_
29   _help , correct = FALSE)
30
31 # T-test van gemiddelde van tijden
32 meanCombinedTasksWithoutOnboarding <- mean(subSetWithoutOnboarding$times_mean)
33 round(meanCombinedTasksWithoutOnboarding , 2)
34 sdCombinedTasksWithoutOnboarding <- sd(subSetWithoutOnboarding$times_mean)
35 round(sdCombinedTasksWithoutOnboarding , 2)
36
37 meanCombinedTasksWithOnboarding <- mean(subSetWithOnboarding$times_mean)
38 round(meanCombinedTasksWithOnboarding , 2)
39 sdCombinedTasksWithOnboarding <- sd(subSetWithOnboarding$times_mean)
40 round(sdCombinedTasksWithOnboarding , 2)
41
42 t.test(x = dataSet$onboarding_elements , y = dataSet$times_mean)
43
44 # T-tests per opdracht
45
46 ## Settings Task
47 meanSettingsTaskWithoutOnboarding <- mean(subSetWithoutOnboarding$settings_task)
48 round(meanSettingsTaskWithoutOnboarding , 2)
49 sdSettingsTaskWithoutOnboarding <- sd(subSetWithoutOnboarding$settings_task)
50 round(sdSettingsTaskWithoutOnboarding , 2)
51
52 meanSettingsTaskWithOnboarding <- mean(subSetWithOnboarding$settings_task)
53 round(meanSettingsTaskWithOnboarding , 2)
54 sdSettingsTaskWithOnboarding <- sd(subSetWithOnboarding$settings_task)
55 round(sdSettingsTaskWithOnboarding , 2)
56
57 t.test(x = dataSet$onboarding_elements , y = dataSet$settings_task)
58
59 ## New Task
60 meanNewTaskWithoutOnboarding <- mean(subSetWithoutOnboarding$new_task)
61 round(meanNewTaskWithoutOnboarding , 2)
62 sdNewTaskWithoutOnboarding <- sd(subSetWithoutOnboarding$new_task)
63 round(sdNewTaskWithoutOnboarding , 2)
64
65 meanNewTaskWithOnboarding <- mean(subSetWithOnboarding$new_task)
66 round(meanNewTaskWithOnboarding , 2)
67 sdNewTaskWithOnboarding <- sd(subSetWithOnboarding$new_task)
68 round(sdNewTaskWithOnboarding , 2)
69
70 t.test(x = dataSet$onboarding_elements , y = dataSet$new_task)
71
72 ## Add Task
73 meanAddTaskWithoutOnboarding <- mean(subSetWithoutOnboarding$add_task)

```

```

72 round(meanAddTaskWithoutOnboarding, 2)
73 sdAddTaskWithoutOnboarding <- sd(subSetWithoutOnboarding$add_task)
74 round(sdAddTaskWithoutOnboarding, 2)
75
76 meanAddTaskWithOnboarding <- mean(subSetWithOnboarding$add_task)
77 round(meanAddTaskWithOnboarding, 2)
78 sdAddTaskWithOnboarding <- sd(subSetWithOnboarding$add_task)
79 round(sdAddTaskWithOnboarding, 2)
80
81 t.test(x = dataSet$onboarding_elements, y = dataSet$add_task)
82
83 ## Delete Task
84 meanDeleteTaskWithoutOnboarding <- mean(subSetWithoutOnboarding$delete_task)
85 round(meanDeleteTaskWithoutOnboarding, 2)
86 sdDeleteTaskWithoutOnboarding <- sd(subSetWithoutOnboarding$delete_task)
87 round(sdDeleteTaskWithoutOnboarding, 2)
88
89 meanDeleteTaskWithOnboarding <- mean(subSetWithOnboarding$delete_task)
90 round(meanDeleteTaskWithOnboarding, 2)
91 sdDeleteTaskWithOnboarding <- sd(subSetWithOnboarding$delete_task)
92 round(sdDeleteTaskWithOnboarding, 2)
93
94 t.test(x = dataSet$onboarding_elements, y = dataSet$delete_task)
95
96 ## Calculator Task
97 meanCalculatorTaskWithoutOnboarding <- mean(subSetWithoutOnboarding$calculator_task)
98 round(meanCalculatorTaskWithoutOnboarding, 2)
99 sdCalculatorTaskWithoutOnboarding <- sd(subSetWithoutOnboarding$calculator_task)
100 round(sdCalculatorTaskWithoutOnboarding, 2)
101
102 meanCalculatorTaskWithOnboarding <- mean(subSetWithOnboarding$calculator_task)
103 round(meanCalculatorTaskWithOnboarding, 2)
104 sdCalculatorTaskWithOnboarding <- sd(subSetWithOnboarding$calculator_task)
105 round(sdCalculatorTaskWithOnboarding, 2)
106
107 t.test(x = dataSet$onboarding_elements, y = dataSet$calculator_task)
108
109 ## Add Task Repeat
110 meanAddTaskRepeatWithoutOnboarding <- mean(subSetWithoutOnboarding$add_task_repeat)
111 round(meanAddTaskRepeatWithoutOnboarding, 2)
112 sdAddTaskRepeatWithoutOnboarding <- sd(subSetWithoutOnboarding$add_task_repeat)
113 round(sdAddTaskRepeatWithoutOnboarding, 2)
114
115 meanAddTaskRepeatWithOnboarding <- mean(subSetWithOnboarding$add_task_repeat)
116 round(meanAddTaskRepeatWithOnboarding, 2)
117 sdAddTaskRepeatWithOnboarding <- sd(subSetWithOnboarding$add_task_repeat)

```

```
118 round(sdAddTaskRepeatWithOnboarding, 2)
119
120 t.test(x = dataSet$onboarding_elements, y = dataSet$add_task_repeat)
121
122 # T-test SUS-score
123 meanSUSWithoutOnboarding <- mean(subSetWithoutOnboarding$sus)
124 round(meanSUSWithoutOnboarding, 2)
125 sdSUSWithoutOnboarding <- sd(subSetWithoutOnboarding$sus)
126 round(sdSUSWithoutOnboarding, 2)
127
128 meanSUSWithOnboarding <- mean(subSetWithOnboarding$sus)
129 round(meanSUSWithOnboarding, 2)
130 sdSUSWithOnboarding <- sd(subSetWithOnboarding$sus)
131 round(sdSUSWithOnboarding, 2)
132
133 t.test(x = dataSet$onboarding_elements, y = dataSet$sus)
```

G. Berekeningen R onderzoeksvraag 2

```
1 # Data
2 dataSet <- read.csv(file = "dataset.csv", sep = ";")
3 dataSet$sus_4 <- factor(dataSet$sus_4,
4                           levels = c("Strongly\_disagree", "Disagree", "Neutral", "Agree", "Strongly\_Agree"),
5                           ordered = TRUE)
6 dataSet$sus_10 <- factor(dataSet$sus_10,
7                           levels = c("Strongly\_disagree", "Disagree", "Neutral", "Agree", "Strongly\_Agree"),
8                           ordered = TRUE)
9
10 subSetWithoutOnboarding <- subset(dataSet, onboarding_elements == FALSE,
11                                     select = -c(onboarding_elements))
11 subSetWithOnboarding <- subset(dataSet, onboarding_elements == TRUE,
12                                     select = -c(onboarding_elements))
12
13 # SUS 4
14 meanSUS4WithoutOnboarding <- mean(as.numeric(subSetWithoutOnboarding$ sus_4))
15 round(meanSUS4WithoutOnboarding, 2)
16 sdSUS4WithoutOnboarding <- sd(as.numeric(subSetWithoutOnboarding$sus_4))
17 round(sdSUS4WithoutOnboarding, 2)
18
19 meanSUS4WithOnboarding <- mean(as.numeric(subSetWithOnboarding$sus_4))
20 round(meanSUS4WithOnboarding, 2)
21 sdSUS4WithOnboarding <- sd(as.numeric(subSetWithOnboarding$sus_4))
22 round(sdSUS4WithOnboarding, 2)
23
24 t.test(x = dataSet$onboarding_elements, y = as.numeric(dataSet$sus_4))
25
26 # SUS 10
```

```
27 meanSUS10WithoutOnboarding <- mean(as.numeric(subSetWithoutOnboarding$  
28   sus_10))  
29 round(meanSUS10WithoutOnboarding, 2)  
30 sdSUS10WithoutOnboarding <- sd(as.numeric(subSetWithoutOnboarding$sus_  
31   10))  
32 round(sdSUS10WithoutOnboarding, 2)  
33 meanSUS10WithOnboarding <- mean(as.numeric(subSetWithOnboarding$sus_10))  
34 round(meanSUS10WithOnboarding, 2)  
35 sdSUS10WithOnboarding <- sd(as.numeric(subSetWithOnboarding$sus_10))  
36 round(sdSUS10WithOnboarding, 2)  
37 t.test(x = dataSet$onboarding_elements, y = as.numeric(dataSet$sus_10))
```

H. Berekeningen R onderzoeksvraag 3

```
1 # Data
2 dataSet <- read.csv(file = "dataset.csv", sep = ";")
3
4 # Would keep
5 chisq.test(x = dataSet$onboarding_elements, y = dataSet$would_keep,
   correct = FALSE)
```


I. Berekeningen R onderzoeksvergadering 4

```
1 # Data
2 dataSet <- read.csv(file = "dataset.csv", sep = ";")
3
4 # Changed name
5 chisq.test(x = dataSet$onboarding_elements, y = dataSet$uses_personal_
    initial, correct = FALSE)
```


Bibliografie

- Alnashri, A., Alhadreti, O. & Mayhew, P. (2016). The influence of participant personality in usability tests. *International Journal of Human Computer Interaction (IJHCI)*, 7(1), 1.
- Babich, N. (2019, november 11). *Top 7 Usability Testing Methods*. Verkregen 29 maart 2020, van <https://xd.adobe.com/ideas/process/user-testing/top-7-usability-testing-methods/>
- Balboni, K. (2018). *We categorized over 500 user onboarding experiences into 8 UI/UX patterns*. Verkregen 23 februari 2020, van <https://www.appcues.com/blog/user-onboarding-ui-ux-patterns>
- Bufe, A. (2020, februari 6). *How not to lose users during app onboarding*. <https://uxcam.com/blog/ways-to-app-onboard/>
- Calisto, F. M. & Nascimento, J. C. (2018). *SUS Survey*. <https://doi.org/10.13140/rg.2.2.25301.06883>
- Camacho Herrero, S. (2019). *Gamified Learning of Software Tool Functionality : Design and implementation of an interactive in-app learning interface for complex software* (masterscriptie 2019:459). KTH, School of Electrical Engineering en Computer Science (EECS).
- Cardoso, M. C. (2017). The Onboarding Effect. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '17*. <https://doi.org/10.1145/3027063.3027128>
- Carroll, J. M. & Carrithers, C. (1984). Training wheels in a user interface. *Communications of the ACM*, 27(8), 800–806. <https://doi.org/10.1145/358198.358218>
- Coyle, C. L. & Peterson, M. (2016). Learnability Testing of a Complex Software Application. In A. Marcus (Red.), *Design, User Experience, and Usability: Novel User Experiences* (pp. 560–568). Springer International Publishing.

- Desai, N. (2019, juni 30). *Solid Customer Onboarding Drives Higher Retention, Willingness to Pay*. Verkregen 8 april 2020, van <https://www.profitwell.com/blog/positive-onboarding-boosts-retention-wtp>
- Ericsson, K. A. & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data*.
- Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(3), 379–383. <https://doi.org/10.3758/BF03195514>
- Gualtieri, M. (2009). Best practices in user experience (UX) design. *Design Compelling User Experiences to Wow your Customers*, 1–17.
- Haramundanis, K. (2001). Learnability in information design. *Proceedings of the 19th annual international conference on Computer documentation - SIGDOC '01*. <https://doi.org/10.1145/501516.501519>
- Harutyunyan, N. & Riehle, D. (2019). User Experience Design in Software Product Lines. *Proceedings of the 52nd Hawaii International Conference on System Sciences*. <https://doi.org/10.24251/hicss.2019.903>
- Hassenzahl, M. (2013). User experience and experience design. *The Encyclopedia of Human-Computer Interaction*, 2nd Ed.
- Hotjar. (2020, januari 14). *A beginner's guide to usability testing*. Verkregen 10 maart 2020, van <https://www.hotjar.com/usability-testing/>
- Jin, J., Lee, J., Kim, E. & Kim, Y. (2017). OPT: optimal human visual system-aware and power-saving color transformation for mobile AMOLED displays. *Multimedia Tools and Applications*, 77(13), 16699–16720. <https://doi.org/10.1007/s11042-017-5234-4>
- Kaikkonen, A., Kekäläinen, A., Cankar, M., Kallio, T. & Kankainen, A. (2005). Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. *J. Usability Studies*, 1(1), 4–16.
- Kjeldskov, J. & Graham, C. (2003). A review of mobile HCI research methods. *International Conference on Mobile Human-Computer Interaction*, 317–335.
- Kjeldskov, J., Skov, M. B., Als, B. S. & Høegh, R. T. (2004). Is It Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field. In S. Brewster & M. Dunlop (Red.), *Mobile Human-Computer Interaction - MobileHCI 2004* (pp. 61–73). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-28637-0_6
- Lamprecht, E. (2019, december 19). *The Difference Between UX And UI Design - A Laymans Guide*. Verkregen 2 april 2020, van <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/>
- Lewis, J. R. & Sauro, J. (2009). The Factor Structure of the System Usability Scale. In M. Kurosu (Red.), *Human Centered Design* (pp. 94–103). Springer Berlin Heidelberg.
- Marcus, A. (2006). Cross-Cultural User-Experience Design. In D. Barker-Plummer, R. Cox & N. Swoboda (Red.), *Diagrammatic Representation and Inference* (pp. 16–24). Springer Berlin Heidelberg. https://doi.org/10.1007/11783183_4
- Morville, P. (2004, juni 21). *User Experience Design*. Verkregen 22 maart 2020, van http://semanticstudios.com/user_experience_design/
- Nikolov, A. (2017, april 8). *Design principle: Consistency*. Verkregen 1 april 2020, van <https://uxdesign.cc/design-principle-consistency-6b0cf7e7339f>

- Oliveira, J. (2019, augustus 19). *User onboarding principles and best practices: How clear principles and goals can help your users*. Verkregen 8 april 2020, van <https://uxdesign.cc/user-onboarding-principles-and-best-practices-bb1ad848c553>
- Ramirez Alvarez, D. (2018). Guidelines for Onboarding : Developing guidelines by testing a process for onboarding.
- Renz, J., Staubitz, T., Pollack, J. & Meinel, C. (2014). Improving the Onboarding User Experience in MOOCs. *Proceedings EduLearn*.
- Strahm, B., Gray, C. M. & Vorvoreanu, M. (2018). Generating Mobile Application Onboarding Insights Through Minimalist Instruction. *Proceedings of the 2018 on Designing Interactive Systems Conference 2018 - DIS '18*. <https://doi.org/10.1145/3196709.3196727>
- White, C. (2020, januari 27). *What Does A UX Designer Actually Do?* Verkregen 23 februari 2020, van <https://careerfoundry.com/en/blog/ux-design/what-does-a-ux-designer-actually-do/#2-what-does-a-ux-designer-do>