**Reflectieverslag**

**Ontwikkeling Local**

**01-07-18**

**Project Vrije Ruimte**

**XCMH6PR15**

**Ivo Kuijer***307263*

Inleiding

In dit reflectieverslag kijk ik terug het verloop van het keuzeblok ‘Project vrije ruimte’ die ik heb gevolgd van 23 april tot 2 juli 2018. Tijdens dit keuzeblok heb ik gewerkt aan het vervolg van een app waaraan ik tijdens het voorgaande keuzeblok ‘AIT’ ben begonnen. Tijdens AIT waren de concept fase, de designfase en het begin van de ontwikkelingsfase afgerond.

Het doel van dit keuzeblok was om een vervolg te maken op de app en uiteindelijk tot een werkend en volledig eindproduct te komen. Dit eindproduct zal vervolgens toegevoegd kunnen worden aan mijn portfolio. Om dit doel te bereiken is er gewerkt met Scrum en zijn er vier sprint opgezet van elk twee weken. Om de week was er een coaching moment ingepland om de voortgang te bespreken.

Allereerst zal ik in dit verslag mijn voortgang tijdens alle sprints bespreken. Hierna zal ik evalueren op de gebruikte technieken. Als laatste zal ik een algemene conclusie geven en zal ik opsommen wat ik de volgende keer beter kan doen wanneer ik een nieuwe taal leer kennen.

Inhoudsopgave

[Planning 2](#_Toc518252796)

[Sprints 3](#_Toc518252797)

[Sprint 1 3](#_Toc518252798)

[Sprint 2 5](#_Toc518252799)

[Sprint 3 6](#_Toc518252800)

[Sprint 4 7](#_Toc518252801)

[Gebruikte technieken 8](#_Toc518252802)

[React native 8](#_Toc518252803)

[Expo 8](#_Toc518252804)

[Zo veel mogelijk eigen componenten 8](#_Toc518252805)

[Firebase 8](#_Toc518252806)

[Github 9](#_Toc518252807)

[Conclusie 10](#_Toc518252808)

# Planning

Aan het begin van het keuzeblok is een plan van aanpak gemaakt en een Trello board opgezet. In dit Trello board werden alle taken genoteerd die uitgevoerd moeten worden om de app te realiseren. Aan deze taken werd vervolgens een tijd gekoppeld die ik verwacht nodig te hebben om deze taak uit te voeren. Uit mijn eigen ervaring weet ik dat ik vaak taken te krap inschat. Dit komt doordat ik geen ruimte inplan voor eventuele tijdrovende bugs waar veel onderzoek voor nodig is. Hierdoor kom ik vaak in tijdnood. Ik had hierom besloten om de tijd altijd ruim in te schatten.

In totaal bevatte het keuzeblok vier sprints van elk twee weken. Na elke sprint werd er een demo gegeven om aan de andere studenten te laten zien hoe het project er nu bij staat. Bij elke sprint werd uitgegaan van 64 uur (32 uur per week). Toen de tijden aan de taken waren gekoppeld kon ik voor elke sprint de 64 uur vol maken met de taken. De ene sprint zat ik hier iets onder, en de andere sprint iets erboven. Dit compenseerde elkaar op de lange termijn goed.

De volgende planning was gemaakt in het plan van aanpak:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Van | Tot | Inhoud |
| Sprint 1 | 07-05 | 18-05 | Business case opstellen  Re-design uitvoeren  Vervolg ontwikkeling app |
| Sprint 2 | 21-05 | 01-06 | Vervolg ontwikkeling app |
| Sprint 3 | 04-06 | 15-06 | Vervolg ontwikkeling app |
| Sprint 4 | 18-06 | 29-06 | Vervolg ontwikkeling app  Exporteren en ombouwen naar APK en IPA |
| Uitloop | 02-07 | - | Eventuele uitloop |

Deze planning heb ik goed doorlopen op het exporteren en ombouwen naar APK en IPA na. Na sprint 3 had ik de keuze tussen óf zo veel mogelijk (belangrijke) functionaliteiten in de app te bouwen, óf om maar een paar functionaliteiten toe te voegen en dan de app te exporteren. Ik heb er uiteindelijk voor gekozen om voor de eerste optie te gaan. Ik wou graag een zo compleet mogelijk app hebben die ik in mijn portfolio kon toevoegen. Voor mij betekende dat dat de app compleet functioneel is en geen belangrijke functionaliteiten miste. Ik besloot om tijdens de dagen binnen dit keuzeblok de functionaliteiten toe te voegen en vervolgens in mijn eigen tijd verder te bouwen aan de app en deze vervolgens te exporteren.

# Sprints

Elke sprint bracht zijn eigen problemen en succesmomenten met zich mee. In dit hoofdstuk ga ik alle sprints bij langs en bespreek ik wat ik er goed ging, wat er fout ging en hou ik dat heb opgelost.

## Sprint 1

In de eerste sprint was ik voornamelijk bezig met het maken van business case en een re-design van de app. Ik begon met de business case. Hierbij heb ik de Business Model Canvas ingevuld. Tijdens het invullen hiervan bekroop mij steeds meer het gevoel dat dit veel te diepgaand is voor een app die je maakt om een nieuwe taal te leren. Ik deed alsof ik een geheel bedrijf aan het opzetten was terwijl ik daar in de rest van het keuzeblok helemaal geen aandacht meer aan ga schenken.

Het idee was aan het begin dat ik er eventueel geld mee kon gaan verdienen wanneer de app echt beschikbaar is en door de consumenten gebruikt ging worden. Om deze reden zou het slim zijn om in een zo vroeg mogelijk stadium hierover na te denken en eventuele aanpassingen op het gebied van functionaliteiten en design door te voeren in de planning en plan van aanpak. Hierdoor zou ik geen dubbel werk verrichten. Toen ik klaar was met de Business Model Canvas trok ik de conclusie dat ik eigenlijk alleen maar dingen heb lopen opschrijven die bij wijze van uit de lucht zijn gegrepen.

Daarna heb ik gewerkt aan de re-design van de app. Ik had tijdens het keuzeblok ‘AIT’ al een design gemaakt voor de app maar daar was ik nog niet helemaal tevreden mee. Het kleurenpalet, de naam en het logo was snel in elkaar gezet en was nog voor verbetering vatbaar. Ook wou ik een andere sfeer creëren rond de app. Uiteindelijk heb ik hier bijna een week tijd ingestoken en ben ik de tel kwijtgeraakt hoeveel logo’s ik ontworpen heb.

Ik ben erachter gekomen dat je eigenlijk nooit tevreden bent het een ontwerp als je het maakt voor jezelf. Wanneer je iets ontwerpt voor een klant kan je je nog optrekken aan de eisen en wensen van de klant en is dat de richtlijn. Als je ontwerpt voor jezelf kan je bij wijze van alle kanten op en dat zorgt bij mij voor veel chaos en besluiteloosheid. Op gegeven moment heb ik de knoop doorgehakt met een ontwerp omdat ik anders nog meer tijd hieraan kwijt zou zijn.

Naast de business case en de re-design waren er nog enkele bugs die ik uit de app wou halen. Het ging hierbij over de formulieren binnen de app. Hierin zaten nog twee bugs die ik er graag uit zou hebben voordat ik in sprint 2 met een schone lei, dus bug-loos, kon beginnen. De eerste bug zorgt ervoor dat de inhoud van de pagina verdwijnt achter het toetsenbord wanneer deze zich opent. Hierdoor is een groot gedeelde van de pagina niet meer zichtbaar. Verder was er een bug die mij niet toeliet om bij de formulieren tussen de inputvelden te ‘tabben’ (denk aan de ‘tab’ toets op het toetsenbord van een pc).

Het probleem rond de inhoud van de pagina die overschaduwd werd door het toetsenbord was in vergelijking met het andere probleem snel opgelost. Het was een hoop gedoe om de juiste componenten hiervoor te zoeken en deze op de juiste manier met elkaar te laten samenwerken, maar uiteindelijk was het gelukt.

Het probleem rond het ‘tabben’ door de inputvelden bleek een groter probleem. Na veel onderzoek op internet had ik eindelijk een oplossing gevonden die zou moeten werken, echter gooide een package die ik gebruik tijdens het ontwikkelen van de app roet in het eten. Alle oplossingen die ik dus online vond waren óf voor een verouderde versie van React Native, óf waren in conflict met de package. Ik besloot deze taak te laten liggen tot de volgende sprint.

Al met al was deze eerste sprint dus een beetje een domper voor mij. Ik had gehoopt om met een fris gevoel de sprints in te gaan, maar na deze sprint had ik vooral het idee dat ik twee weken heb weggegooid.

## Sprint 2

Ik begon deze sprint met het oplossen van het probleem rond de formulieren binnen de app. Corné had mij na de demo van sprint 1 aangeraden om eerst geheel uit de omgeving van de app een kleinschalige test omgeving te maken en daarin het formulier eerst werkend te krijgen. Wanneer dat is gelukt kan er worden gekeken wat de reden is waardoor het in de testomgeving wel lukt en in de echte app niet. Dit kan gedaan worden door steeds kleine stukken code over te zetten en te kijken op welk moment de app ermee ophoudt. Ik heb deze aanpak uitgeprobeerd en het beviel mij goed. Waar ik eerst dagen bezig was om het op te lossen was het met deze aanpak binnen een paar uur opgelost.

Door de bovenstaande aanpak te gebruiken ben ik mezelf erg tegen gekomen. Ik ben erachter gekomen dat ik maar blijf proberen en proberen om een probleem op te lossen en daarbij steeds gefrustreerder raak. Ik heb dan het idee dat het te veel tijd kost om opnieuw te beginnen in een aparte omgeving. Ik heb nu het besef dat het juist minder tijd kost om dat te doen. Bovendien scheelt het een hoop frustratie. Dit is zeker een aanpak die ik goed zal onthouden en vaak zal toepassen bij problemen waar ik niet uit kom.

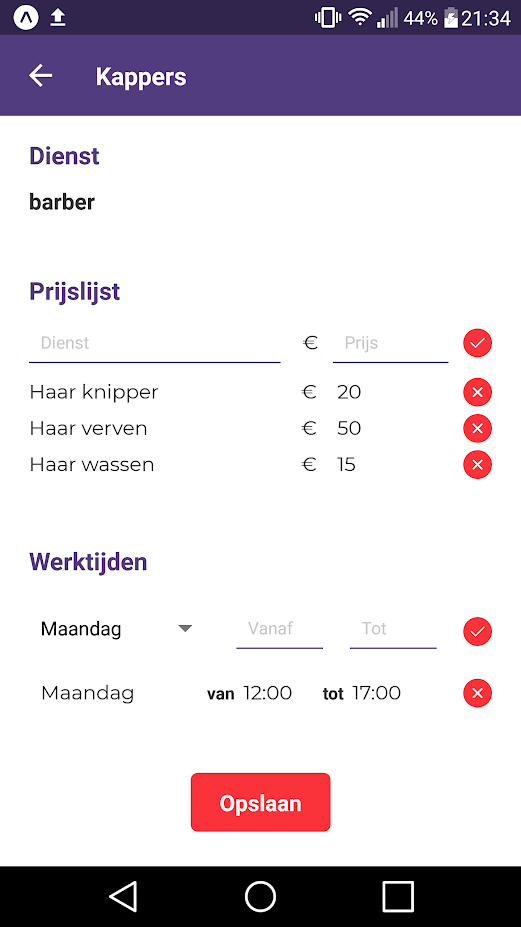
Ik was blij dat ik na het oplossen van de bug rond de formulieren met een schone lei verder kon. Ik ging bezig met het maken van login functionaliteit en het maken van de ‘Mijn Diensten’ pagina. Tijdens het ontwikkelen hiervan kwam ik erachter dat ik nog wat kennis miste van redux, het state management systeem achter React Native. Ik had hiervan wel wat kennis opgedaan door een Udemy cursus die ik tijdens het keuzeblok ‘AIT’ heb gevolgd maar dat is een andere context dan de app waar ik zelf aan werk en ik was niet helemaal comfortabel met de manier van coderen die mij daar werd aangeleerd. Ik besloot om een andere kleinere cursus online te volgen die speciaal gericht is op redux zodat ik hierin een eigen manier van coderen kon ontwikkelen. Ik ben erachter gekomen dat ik veel consistentie en een logische opbouw nodig heb in mijn code om me er prettig bij te voelen. Als dat ontbreekt merk ik dat de chaos wat toeneemt waardoor ik minder efficiënt kan werken.

Nadat ik meer kennis had opgedaan in redux kon ik verder met mijn taken. Deze verliepen hierna verassend voorspoedig. Door de keuze te maken om even een stap terug te doen en meer kennis te verzamelen, in tegenstelling tot door te werken tegen beter weten in, heb ik voor mijzelf zowel tijd bespaard als een prettiger gevoel gecreëerd. Aan deze keuze merk ik dat ik vooruitgang boek in het leren van een nieuwe taal. Waar ik eerst veel door bleef werken tegen beter weten in neem ik steeds vaker dus een stap terug om te kijken wat een betere oplossing is.

## Sprint 3

Aan het einde van de derde sprint was het de bedoeling dat de MVP (Minimal Viable Product) zo goed als af was. Met andere woorden, de app moest nu de kernfunctionaliteiten bevatten en deze moeten naar behoren werken. In mijn plan van aanpak had ik alle functionaliteiten van de app geprioriteerd op basis van de MoSCoW regels. Ik had voor mijzelf bepaald dat ik aan het einde van deze sprint alle functionaliteiten onder de categorie Must-haves af zou hebben. Deze categorie had ik gelijkgesteld aan de MVP. Dit is gelukkig goed gelukt.

In deze sprint heb ik ook gewerkt aan het herinrichten van de navigatiestructuur. Ik kwam erachter dat elke gebruikersrol zijn eigen pagina’s nodig heeft en tot nu toe had ik als het ware alle pagina’s op 1 bult gegooid. Elke gebruiker kon dus ook op elke pagina komen en elke gebruiker zag dezelfde menustructuur. Ik heb deze sprint drie verschillende navigatiestructuren gemaakt. De dienstverlener, de consument en de niet-ingelogde gebruiker hebben nu elk een eigen navigatiestructuur. Achteraf gedacht had ik dit wel kunnen voorspellen en hierop kunnen anticiperen door vanaf het begin al gebruik te maken van deze drie verschillende structuren. Nu heeft het redelijk wat tijd gekost om het om te bouwen. Dit is aan de andere kant wel een goede les geweest aangezien ik nu extra aandacht heb kunnen schenken aan wat er mogelijk is en wat er allemaal mogelijk is. Deze kennis kan ik meenemen naar volgende projecten om daar vanaf het begin hier de juiste keuzes in te maken.

Aan het einde van de sprint was ik bezig met het ontwikkelen van de pagina waarop de dienstverleners een nieuwe dienst kunnen toevoegen. Hierbij kan de dienstverlener een dienst selecteren (kapper, autowasser, etc.), verschillende prijzen toevoegen en verschillende zijn/haar beschikbaarheid aangeven. Nadat er een nieuwe prijs of beschikbaarheid is toegevoegd kan deze ook weer verwijderd worden door op het kruisje erachter te klikken (zie afbeelding). Op dit moment werd de prijs of beschikbaarheid wel verwijderd, maar werd dit niet direct zichtbaar. Pas nadat een andere actie werd ondernomen, zoals het invullen van een inputveld, werd het zichtbaar. Dit moest nog opgelost worden en werd doorgeschoven naar de volgende sprint.

## Sprint 4

Tijdens de vierde sprint had ik de taal goed genoeg onder de knie dat ik redelijk complexe functionaliteiten in minder tijd dan verwacht kon afronden. Het gaat hierbij om de chat en recensies functionaliteiten. Ik had verwacht hier veel tijd aan kwijt te zijn maar uiteindelijk kon ik voor deze functionaliteiten, vooral voor de recensies, veel eerder geschreven code hergebruiken. De methode rond het maken van een object en die opslaan in de database had ik al toegepast bij de functionaliteiten rond het maken van het toevoegen van een dienst of het registreren van een gebruiker.

Er was tijdens deze sprint nog één probleem waar ik helaas niet uit ben gekomen. Namelijk het uploaden van een afbeelding naar Firebase. Dit is een functionaliteit waar ik tijdens AIT ook al aandacht had besteed maar toen ook niet lukte. Ik had de hoop dat ik met alle nieuwe kennis dit probleem nu wel op zou kunnen lossen. Toch mocht het niet zo zijn. Er bleef een fout ontstaan wanneer de afbeelding op wou slaan in Firebase. Ik gebruikte hiervoor een component van Expo, aangezien de normale componenten niet met Expo werken, en deze geeft de verkeerde data terug wanneer een afbeelding wordt geüpload. Deze verkeerde data wil Firebase niet accepteren wanneer ik het daar probeer op te slaan.

Ik heb dit probleem geprobeerd op te lossen door eerst alle andere taken af te ronden en vervolgens een stap terug te nemen en een losse testomgeving te maken. Toch liep ik er ook hier tegenaan dat ik dus zelfs in de testomgeving nog steeds de verkeerde data terugkreeg. Vanwege tijdstekort heb ik besloten deze functionaliteit voor de laatste demo eruit te halen. Wanneer ik hier later mee verder ga heb ik meer tijd om dit probleem om te lossen.

# Gebruikte technieken

## React native

Van alle talen die ik tot nu toe heb geleerd vind ik React Native wel de makkelijkste taal om te leren. De structuur op basis van componenten spreekt mij heel erg aan. Voor mij is het heel logisch en dat zorgde er ook voor dat ik het sneller onder de knie kreeg. Het aanspreken van de native APIs viel ook reuze mee. React Native handelt alles voor je af zodat je er zelf eigenlijk niet veel aandacht aan hoeft te besteden. De voorgeprogrammeerde componenten spelen hierin een grote rol.

## Expo

Ik heb Expo gebruikt voor het creëren van een ontwikkelingsomgeving. Het voordeel van Expo is dat je met weinig moeite de app realtime kan testen op je eigen toestel. Ook heeft expo zijn eigen set met voorgeprogrammeerde componenten. Het nadeel van Expo is dat sommige functionaliteiten of componenten van React Native zelf niet meer werken. Je moet hiervoor dus omwegen bouwen binnen Expo. Voor de volgende keer dat ik een app ga ontwikkelen in React Native zal ik waarschijnlijk er niet voor kiezen om Expo te gebruiken. Dat komt puur door de omwegen die je dan moet gaan bouwen. Ik zal er dan liever voor zorgen dat ik zonder Expo een goede ontwikkelingsomgeving opzet waar ik met pure React Native kan werken en daarin ook alles van React Native kan gebruiken.

## Zo veel mogelijk eigen componenten

Zoals eerder vermeld hebben zowel React Native als Expo hun eigen voorgeprogrammeerde componenten. Deze zijn noodzakelijk voor het bouwen van de applicatie aangezien deze componenten worden omgezet naar native code. Ik vindt sommige componenten hiervan echter wat onhandig en dus heb ik veel eigen componenten geschreven en deze als het ware als laag bovenop de voorgeprogrammeerde componenten te leggen. Zo worden ze nog steeds op dezelfde wijze omgezet naar native code maar zijn ze voor mij een stuk prettiger in het gebruik. Ook zorg ik er hiermee voor dat de code compact en overzichtelijk blijft.

## Firebase

Als database achter de app maak ik gebruik van Firebase. Dit is een realtime datastore die werkt op basis van JSON. Veel andere databases werken met relaties, iets wat niet mogelijk is in React Native. Met relaties is het heel makkelijk, vooral bij een app als deze, om bijvoorbeeld dienstverleners aan diensten te koppelen. Is Firebase is dit een stuk lastiger.

Aan het begin van het keuzeblok ‘AIT’ had ik de keuze om óf Firebase, óf een database op basis van onderlinge relaties te gebruiken. Hoewel Firebase niet het beste geschikt was voor deze app had ik er toch voor gekozen omdat ik er al kennis van had opgedaan vanuit de Udemy cursus en omdat de Firebase heel goed te combineren is met React Native. Bovendien is mijn kennis van back-end development te mager om zelf APIs te schrijven. Dit zou dus een hoop tijd hebben gekost om dit goed te leren.

Als ik de volgende keer een project start in React Native waarbij ik zelf ook voor de database moet zorgen zal ik waarschijnlijk toch de tijd erin gaan steken om gebruik te maken van een relatie-gebaseerde database. De voordelen hiervan zijn heel groot en de kennis die ik ervan opdoe is alleen maar mooi meegenomen. De voorwaarde hiervan is wel dat ik genoeg tijd heb om dit goed te leren.

## Github

Als versiebeheersysteem heb ik gebruik gemaakt van Github. Het voordeel van een versiebeheersysteem spreekt voor zich en Github is de grootste speler op dit gebied. Bovendien heb ik tijdens mijn stage kennis opgedaan van Github en deze kennis wou ik graag uitbreiden door het ook te gebruiken in dit project.

Een medestudent maakte gebruik van BitBucket en toen ik dat zag sprak het mij ook erg aan. In Github is het bijvoorbeeld niet mogelijk om je eigen repository prive te houden zonder hiervoor te betalen. Bij BitBucket is dit wel mogelijk. Bovendien werkt BitBucket nauw samen met Trello wat ook veel voordelen met zich mee brengt.

# Conclusie

Ik kijk met veel tevredenheid terug op de afgelopen paar weken. Ik heb veel geleerd over het ontwikkelen van een applicatie in React Native, maar boven dat heb ik geleerd hoe ik het beste een nieuwe taal kan leren. Ik ben van mening dat ik in deze weken een goed product heb neergezet waar ik trots op mag zijn. Het eindproduct is uiteindelijk goed in de richting gekomen van hoe ik het voor ogen had. Er zitten nog wel wat kleine bugs en inconsistenties in maar als ik kijk naar het grote geheel mag ik zeker niet klagen.

Wat mij wel opvalt als ik terugkijk naar de code is dat ik ben geëindigd met veel inconsistenties in de syntax en in de opbouw van functies en componenten. Naarmate ik door de sprints heen ging leerde ik de taal beter en leerde ik ook ES6 beter. Je kan dus ook zien dat de bestanden die ik aan het begin van de blok heb geschreven veel inefficiënter zijn dat de bestanden die ik in de laatste week heb geschreven. Ik heb wel geprobeerd om zoveel mogelijk de oude code te herschrijven maar op gegeven moment zit er alleen maar tijd in zonder direct resultaat, anders dan nettere code. Ik heb voor mijzelf geaccepteerd dat de code wat inconsistent is aangezien ik van mijzelf weet dat ik het volgende project in React Native veel beter zal gaan schrijven.