



Faculteit Bedrijf en Organisatie

Implementatie van Analytics in een Native app gebruikmakend van React Native met Managed Expo

Billy Cottrell

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

Promotor:
Guy Dekoning
Co-promotor:
Tim Bogemans

Instelling: Cogn-IT

Academiejaar: 2019-2020

Eerste examenperiode

Faculteit Bedrijf en Organisatie

Implementatie van Analytics in een Native app gebruikmakend van React Native met Managed Expo

Billy Cottrell

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

Promotor:
Guy Dekoning
Co-promotor:
Tim Bogemans

Instelling: Cogn-IT

Academiejaar: 2019-2020

Eerste examenperiode

Woord vooraf

Deze bachelorproef werd geschreven in het kader van het voltooien van de opleiding Toegepaste Informatica afstudeerrichting Mobiele Applicaties. Ik heb dit onderzoek gekozen omdat dit mij enorm interesseert en omdat dit ook in lijn loopt met mijn afstudeerrichting.

Het schrijven van deze scriptie is een leerrijke ervaring geweest die mij zowel heeft toegestaan een verder begrip te krijgen over [React Native](#) en de desbetreffende [analytics tools](#). Zonder de hulp van volgende mensen had ik deze scriptie nooit tot een goed einde kunnen brengen. Bij deze wens ik ze dan ook allemaal te bedanken.

Ik wil dan ook mijn co-promotor Tim Bogemans bedanken voor de begeleiding en het structureren van het onderzoek, voor de aanwijzingen en verbeteringen die mij instaat hebben gesteld het werk verder te verbeteren. Verder wil ik Matthias D'haeseleer bedanken voor de technische begeleiding en hulp bij het opzetten van de applicaties.

Ook wil ik mijn promotor Guy Dekoning bedanken voor zijn feedback over deze bachelorproef.

Verder wil ik ook de coördinator Chantal Teerlinck, mijn familie en vrienden bedanken voor hun steun en eindeloos begrip.

Ik wens u veel leesplezier toe.

Billy Cottrell

Aalst, 21 januari 2020

Samenvatting

In deze scriptie wordt er onderzocht welke [analytics](#) aanbieder de beste keuze is binnen een mobiele applicatie, waarbij er technologieën zoals [React Native](#) en [Managed Expo](#) als basis gebruikt zullen worden. De [analytics tools](#) die gedurende dit onderzoek vergeleken zullen worden zijn [Google Analytics](#), [App Center Analytics](#) en [Expo Analytics](#).

Mobiele applicaties zijn tegenwoordig belangrijk geworden voor iedereen, het is echter zo dat er bij het gebruik van deze applicaties weleens problemen kunnen ontstaan. Daarom is het zo belangrijk voor bedrijven om [analytics tools](#) binnen mobiele applicaties te kunnen gebruiken. Door deze [analytics tools](#) te gebruiken krijgen de bedrijven een beter inzicht. Zo weten de bedrijven hoe de klanten met de applicatie omgaan, of waarom er bij een aankoop proces klanten plots uit dit proces stappen, op deze manier kunnen de bedrijven problemen opsporen.

Dit is dan ook de reden waarom het bedrijf Cognit (Gespecialiseerd in Microsoft SharePoint en Office 365 om medewerkers efficiënter te laten samenwerken) een [analytics tool](#) wil gaan gebruiken binnen hun mobiele applicatie van hun Involv product (dat voornamelijk inzet om mensen betrokken te maken en een betere samenwerking te stimuleren binnen bedrijven).

Tijdens dit onderzoek zijn er 3 applicaties gemaakt, waarbij er telkens 1 [analytics tool](#) gebruikt werd. De [tools](#) worden telkens gebruikt in een recepten applicatie waarbij gebruikers kunnen navigeren, recepten liken, de details bekijken en zelfs een simulatie waarbij recepten aangekocht kunnen worden. Dit zorgt ervoor dat we kunnen bepalen of alle functionaliteiten van de [analytics tool](#) werken, of alle data verkregen kan worden met een basis implementatie en welke data een uitgebreide implementatie vereisen.

De resultaten die uit het onderzoek kwamen waren bijna zoals verwacht. Weliswaar bleek dat [App Center Analytics](#) niet gebruikt kon worden, omdat de omgeving niet geschikt was hiervoor. Dit was dan ook hetzelfde resultaat voor [Google Analytics](#) wanneer de voorgestelde [tool](#) gebruikt werd, er was wel de mogelijkheid om [Google Analytics](#) te gebruiken mits een eigen implementatie werd gemaakt en de data werd verstuurd via een URL. De meest geschikte [tool](#) voor snel te gebruiken is [Expo Analytics](#), dit komt omdat het een implementatie is voor [Google Analytics](#) en enkel geschikt is binnen de gebruikte omgeving van dit onderzoek.

Uit dit resultaat kan er dus afgeleid worden dat [App Center Analytics](#) en [Google Analytics](#) eigen [tool](#) niet geschikt zijn om te implementeren binnen de gebruikte omgeving. Verder kan er afgeleid worden dat [Expo Analytics](#) de meest geschikte kandidaat is indien er een [analytics tool](#) nodig is voor op korte termijn en deze snel implementeerbaar moet zijn. Indien er een [analytics tool](#) nodig is met meer mogelijkheden en waarbij er voldoende tijd voor handen is, dan is [Google Analytics](#) met een eigen implementatie de meest geschikte [tool](#).

Er zijn zeker en vast mogelijkheden voor verder onderzoek over dit onderwerp. De focus zal weliswaar liggen op andere [analytics tools](#), of op de integratie mogelijkheden met een eigen implementatie van [Google Analytics](#).

Inhoudsopgave

1	Inleiding	25
1.1	Probleemstelling	26
1.2	Onderzoeksvraag	26
1.3	Onderzoeksdoelstelling	26
1.4	Opzet van deze bachelorproef	27
2	Stand van zaken	29
2.1	Inleiding	29
2.2	Mobiele Applicaties	29
2.2.1	Native App	30
2.2.2	Hybrid App	30
2.2.3	Progressieve Web App	30
2.2.4	Cross-Platform App	31

2.3	React Native	31
2.3.1	React	33
2.4	Expo	33
2.4.1	Bare Expo	34
2.4.2	Managed Expo	34
2.5	Analytics Tools	35
2.5.1	Google Analytics	35
2.5.2	Microsoft Visual Studio App Center Analytics	37
2.5.3	Expo Analytics	38
2.6	Google Analytics vs. App Center Analytics vs. Expo Analytics	39
3	Methodologie	41
4	Demo Applicatie	43
4.1	Inleiding	43
4.2	Architectuur	44
4.2.1	Systeem Context Diagram	45
4.2.2	Container Diagram	46
4.2.3	Component Diagram	47
4.3	Opzetten Van De Omgeving	49
4.4	Maken van de applicatie	50
5	Expo Analytics	57
5.1	Data en Functies	57

5.2	Implementatie en Architectuur	60
5.2.1	Implementatie	60
5.2.2	Architectuur	67
5.3	Documentatie	69
5.4	Prijs / Kwaliteit	70
5.5	Case studies / Getuigenissen	70
6	Google Analytics	71
6.1	Data en Functies	71
6.2	Implementatie en Architectuur	72
6.3	Documentatie	73
6.4	Prijs / Kwaliteit	73
6.5	Case Studies / Getuigenissen	74
7	Microsoft App Center Analytics	75
7.1	Data en Functies	75
7.2	Implementatie en Architectuur	76
7.3	Documentatie	76
7.4	Prijs / Kwaliteit	77
7.5	Case Studies / Getuigenissen	77
8	Conclusie	79
8.1	Conclusie onderzoek	79

8.2	Antwoorden op de onderzoeksvragen	80
8.2.1	Welke tool biedt de meeste mogelijkheden (collecteren en verwerking van data)?	80
8.2.2	Welke tool werkt het beste samen met Managed Expo?	81
8.2.3	Welke tool is het meest eenvoudige/gebruiksvriendelijke te implementeren zonder bijkomende aanpassingen?	81
8.2.4	Wat zijn de voor- en nadelen bij het gebruik van de tool?	81
8.2.5	Welke tool is integreerbaar met bestaande Analytics (momenteel Google Analytics)?	82
8.2.6	Wie heeft de ownership van de data en wat met privacy (GDPR)? ...	82
8.3	Conclusie van deze bachelorproef	83
A	Onderzoeksvoorstel	85
A.1	Introductie	85
A.2	Literatuurstudie	85
A.2.1	React Native	86
A.2.2	Managed Expo	86
A.2.3	Analytics	86
A.3	Methodologie	87
A.4	Verwachte resultaten	88
A.5	Verwachte conclusies	88
	Bibliografie	89

Lijst van figuren

2.1	Hoe kies je de juiste applicatie voor jouw project? (Korolev, 2019)	31
2.2	De architectuur van React Native (Grover, 2018)	32
2.3	De hierarchie van Google Analytics (Google, 2020)	38
4.1	Systeem Context diagram	45
4.2	Container diagram	46
4.3	Component diagram	47
4.4	Folder structuur demo applicatie	51
4.5	Startscherm Get Ideas	52
4.6	My Recipes scherm	53
4.7	Leftovers scherm	54
4.8	Recipe Detail scherm	55
5.1	Aanmaken Google Analytics Account	60
5.2	Opzetten Google Analytics Account	61
5.3	Keuze van platform voor specifieke tracking	61
5.4	Opzetten van een property	62
5.5	Verkrijgen van de tracking id	62
5.6	Custom Dimensions en Metrics	65

5.7	Pagina voor het aanmaken van Dimensies en Metrics	65
5.8	Metric aanmaken	66
5.9	Component diagram van Recipe Chef met Expo Analytics	68

Lijst van tabellen

2.1	Verschillen tussen de Analytics Tools	40
8.1	De verschillen tussen Google Analytics, Microsoft Visual Studio App Center Analytics en Expo Analytics volgens het onderzoek.	80
8.2	Samenvatting onderzoeksvragen alleen voor Managed Expo	83

Woordenlijst

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | R | S | T | U | V | W | X

A

analytics is het analyseren, ontdekken, interpreteren en communiceren van betekenisvolle patronen in gegevens en hierop in te spelen. 1, 3, 5, 6, 16–18, 20–22, 25–27, 29, 35, 38, 39, 41–45, 48, 50, 51, 62, 63, 67, 70, 71, 74, 76, 77, 81, 82

Android Studio is de officiële [IDE](#) voor het Android besturingssysteem (OS) van Google, speciaal ontworpen om Android applicaties te ontwikkelen. 1, 34, 50

Angular is een platform voor het bouwen van mobiele en desktop [web applicaties](#). Het is een [TypeScript](#) gebaseerd open-source [web applicatie framework](#) geleid door het Angular Team bij Google en door een gemeenschap van individuen en bedrijven. Angular is een volledige herwerking van AngularJS dat oorspronkelijk door dat team was gebouwd. 1, 20, 30, 49

Apache Cordova is een [framework](#) voor de ontwikkeling van mobiele applicaties. Het stelt software programmeurs in staat om applicaties voor mobiele apparaten te bouwen met CSS, HTML en JavaScript in plaats van platform specifieke API's zoals bij Android, IOS of Windows Phone. 1, 30

App Center geeft toegang tot Microsoft [Visual Studio](#) App Center, om applicaties te bouwen, beheren, testen, releasen en monitoren voor elk platform. Zo geeft het ook de mogelijkheid tot snellere [release cycli](#), hogere kwaliteit applicaties en data om te focussen op wat de mensen willen.. 1, 37–39, 76, 77

App Center Analytics (voluit Microsoft Visual Studio App Center Analytics) is een dienst van Microsoft om het gebruikersgedrag en klantbetrokkenheid te begrijpen door middel van rapporten. Met als doel om met deze informatie de applicatie te kunnen verbeteren. 1, 5, 6, 27, 35, 37–41, 75–77, 79–82

app secret is net zoals een API key, het laat toe om [events](#) en telemetrie te versturen. Weliswaar biedt het geen toegang tot een account of om API's aan te roepen. 1, 38

asynchroon verwijst naar het optreden van gebeurtenissen onafhankelijk van het hoofdprogramma stroom en manieren om met dergelijke gebeurtenissen om te gaan. 1, 32, 49

B

Bare Expo is de kale werkwijze binnen [Expo](#), deze werkwijze wordt bekomen wanneer men van [Managed Expo](#) afstapt. Hierbij heeft de ontwikkelaar de volledige controle, maar ook de complexiteit die [Managed Expo](#) beheerd. Zo moet de ontwikkelaar per [native platform](#) een applicatie maken en ontbreken er verschillende [services](#) onder andere de “[build service](#)”. 1, 34, 80–82, 84

big data extreem grote datasets die computationeel kunnen worden geanalyseerd om patronen, trends en associaties te onthullen, vooral met betrekking tot menselijk gedrag en interacties. 1, 25

build service zorgt voor het bouwen van de gemaakte applicatie zodat de ontwikkelaar het resultaat kan bekijken. 1, 16, 34

C

campagne verzameling [marketing activiteiten](#) gericht op het promoten van een product, dienst of organisatie. Zo zijn er 3 soorten campagnes namelijk reclamecampagne voor naamsbekendheid, verkiezingscampagne om stemmen te ronselen en een protestcampagne om ergens tegen te protesteren. 1, 16, 36, 57, 60, 72

client is een stuk computer hardware of software dat toegang heeft tot een [service](#) die door een server beschikbaar is gesteld. Deze server staat meestal (maar niet altijd) op een ander computersysteem, waardoor de cliënt via een netwerk connectie toegang krijgt tot die bepaalde [service](#). 1, 34

compileer is het vertalen van een doel programmeertaal naar een andere programmeertaal. Zo wordt bijvoorbeeld [TypeScript](#) gecompileerd naar JavaScript, dit is omdat [TypeScript](#) niet rechtstreeks in de browser werkt en het gebouwd is op basis van JavaScript en het werkt in de browser. 1, 17, 21, 22, 32, 36

component is een herbruikbaar software element, dit wordt gebruikt bij component gebaseerd software ontwikkeling. 1, 19, 20, 33, 44, 48, 51, 67

cross-platform kan worden gebruikt op verschillende soorten computers of met verschillende besturingssystemen. Zo kan 1 applicatie zowel op iOS als op Android werken. 1, 19, 25, 29, 31, 72

D

data stream is een stroom van data afkomstig van de gekozen applicatie of website van waar data moet gecollecteerd worden. Zo kunnen meerdere data streams opgevolgd worden in één [property](#). 1, 36

dimension is een beschrijvend attribuut of karakteristiek van een object waaraan verschillende waarden gegeven worden. Browser, landings pagina en [campagne](#) zijn allemaal voorbeelden van standaard dimensions in [analytics](#). 1, 19, 37, 39, 57, 59, 65, 66, 70, 71, 73, 74, 80

E

e-commerce ook wel elektronische handel is het elektronisch kopen of verkopen van producten op online diensten of via internet. Het maakt gebruik van technologieën

zoals mobiele handel, elektronische overboeking, internet marketing, en meer. 1, 18, 39, 59, 67

eject is het uitwerpen of verlaten van een bepaalde status. Zo kan men ook ejecten om uit een bepaalde ontwikkelingsmode te stappen en een andere mode dan weer te activeren.. 1, 34

event een softwarebericht dat aangeeft dat er iets is gebeurd zoals een toetsaanslag of muisklik. 1, 15, 17, 26, 37–39, 49, 57–60, 63, 64, 70, 71, 73–75, 80, 82

Expo is een [framework](#) en een platform voor universele [React](#) applicaties. Het is een set van [tools](#) en [services](#) gebouwd rond [React Native](#) en [native platformen](#) die helpen met het ontwikkelen, bouwen, implementeren en snel itereren op iOS, Android en [web applicaties](#) vanuit dezelfde JavaScript of [TypeScript](#) codebase. 1, 16, 17, 19, 25, 29, 33, 34, 38, 43, 49, 50

Expo Analytics is een [library](#) dat [Google Analytics](#) integreert voor het gebruik met [React Native](#) applicaties die gebouwd zijn op [Expo](#). Deze [library](#) vereist geen [linking](#)/koppeling met [Google Analytics](#), terwijl andere [libraries](#) of [SDK's](#) zoals [Firebase Analytics](#) (voor gebruik van [Google Analytics](#)) wel [linking](#)/koppeling vereisen die niet door [Expo](#) worden ondersteund. 1, 5, 6, 27, 35, 38–41, 48, 57, 58, 60–63, 69–74, 79–84

F

Firebase is Google's ontwikkelings platform voor mobiele applicaties dat helpt met het bouwen, verbeteren en groeien van een applicatie. 1, 17, 36, 71–73

Firebase Analytics is het [analytics](#) gedeelte van [Firebase](#), het is een platform van Google en dient voor mobiele ontwikkelaars. Het is gericht op het analyseren van applicaties en biedt applicatie gerichte rapportage en analyse. 1, 17, 73, 81

first-party met betrekking tot de persoon, groep of organisatie die de rechtmatige eigenaar is van de situatie of degene die de situatie heeft gecreëerd en mogelijk ook 2de of 3de partijen er bij betreft. 1, 25

Flutter is de UI toolkit van Google voor het maken van prachtige, native [gecompileerde](#) applicaties voor mobiel, web en desktop vanuit één enkele codebase. 1, 31

framework is gemaakt om het ontwikkelen van applicaties te ondersteunen en het eenvoudiger te laten lopen. Zo geeft een framework een basis en werkwijze om hierin te kunnen ontwikkelen. 1, 15, 17, 20, 21, 25, 31–33, 49

G

GDPR ook wel General Data Protection Regulation is een Europese verordening (met rechtstreekse werking) die de regels voor de verwerking van persoonsgegevens door particuliere bedrijven en overheidsinstanties in de EU standaardiseert. 1, 26, 60, 67, 75, 82, 84

Global Site Tag ook wel gtag.js is een JavaScript [tagging framework](#) en API waarmee [event](#) data kan worden verzonden naar verschillende producten van Google Suite. In plaats van meerdere [tags](#) te moeten beheren voor verschillende producten, kan de gtag.js gebruikt worden en kan er gemakkelijker geprofiteerd worden van de laatste nieuwe meetfuncties en integraties wanneer deze beschikbaar worden. 1, 37

Google Analytics is een dienst van Google om statistieken van een website of mobiele applicatie te verzamelen en weer te geven in verschillende rapporten. Het doel is om de beheerder van de website een duidelijk beeld te geven van het gebruik van de

website of mobiele applicatie op basis van onder andere bezoekersstromen, verkeersbronnen (vanwaar komt de bezoeker) en paginaweergaves. Met deze informatie kan de beheerder van de website reclame campagnes opzetten of de website aanpassen aan het gedrag van de bezoeker. 1, 5, 6, 17, 26, 27, 35–41, 48, 57–60, 63, 65, 69–74, 79–84

Google Tag Manager is een [tag](#) beheer systeem (TMS) waarmee snel en eenvoudig meetcodes en gerelateerde codefragmenten kunt bijwerken die gezamenlijk bekend staan als [tags](#) op uw website of mobiele applicatie. Wanneer de Tag Manager code is toegevoegd aan het project, is er de mogelijkheid om veilig en eenvoudig [analytics](#) en meet [tag](#) configuraties te implementeren vanuit een web gebaseerde UI. 1, 21, 37

H

hit is een interactie die ertoe leidt dat gegevens naar [analytics](#) worden verzonden. Zo zijn er hits voor het bijhouden van pagina's, gebeurtenissen en [e-commerce](#). 1, 21, 39, 66

hybrid applicatie is een software applicatie dat de elementen van zowel [native applicatie](#) als [web applicatie](#) combineert. 1, 18, 30, 31

I

IDE staat voor Integrated Development Environment en is een applicatie die helpt bij het schrijven van de code door nuttige processen te automatiseren zoals debugging (foutopsporing), refactoring (code herstructureren, vereenvoudigen en leesbaarder maken), code generation (code genereren) enz. Een IDE is dus gewoon een hulpmiddel voor programmeurs. 1, 15, 22, 23

Ionic is een open-source [SDK](#) voor het ontwikkelen van [hybrid applicatie](#). 1, 30

J

JSX ook wel JavaScript XML is een combinatie van UI templates en inline JavaScript logica. 1, 33

K

Kotlin is een gratis, open-source, statisch getypeerde “pragmatische” (praktisch) programmeertaal voor algemene doeleinden, dat oorspronkelijk ontworpen is voor de JVM (Java Virtual Machine) en Android, die objectgeoriënteerde en functionele programmeerfuncties combineert. 1, 30

L

library is een verzameling subprogramma's die worden gebruikt om software te ontwikkelen. 1, 17, 20, 25, 33, 37–39, 43, 48–50, 69, 72, 80–83

linking dit is het connecteren van een applicatie op een portaal door gebruik te maken van de applicatie ID. 1, 17, 39, 40

LTS ook wel lange termijn ondersteuning, is een beleid voor product levenscyclus beheer waarbij een stabiele versie van computersoftware wordt onderhouden voor een langere periode dan de standaardeditie. Dit geeft meestal ook aan dat die versie geen grote aanpassingen meer zal ondergaan maar eerder het oplossen van fouten die zich voordoen. 1, 49

M

- Managed Expo** is de beheerde werkwijze binnen [Expo](#), dit is de standaard werkwijze die gebruikt wordt wanneer men met [Expo](#) werkt. Hierbij probeert [Expo](#) zoveel mogelijk de complexiteit van het bouwen van applicaties te beheren voor de ontwikkelaars. 1, 5, 16, 25, 26, 29, 34, 39, 43, 72, 74, 76, 77, 79–82, 84
- marketing activiteit** heeft betrekking op de activiteiten die een bedrijf uitvoert om de verkoop van producten of diensten te bevorderen. 1, 16, 35
- marketing effectiviteit** is de graad waarin de marketing inspanning een vooropgesteld doel haalt. 1, 36
- metadata** is data dat informatie geeft over andere data. Zo zijn er vijf soorten metadata waaronder beschrijvende metadata voor ontdekking en identificatie, structurele metadata voor containers met gegevens en hoe objecten zijn samengesteld, administratieve metadata voor het beheren van een bron, referentie metadata voor inhoud en kwaliteit van statistisch data, statistische metadata voor het beschrijven van processen van statistische data. 1, 36
- metric** is een kwantitatieve meting van gegevens, deze kunnen bedragen of ratio's zijn. Het zijn individuele elementen van een [dimension](#) die kunnen worden gemeten als een som of een verhouding. 1, 21, 37, 39, 57, 59, 65, 66, 70, 71, 73, 74, 80
- Microsoft Azure Application Insights** is een functie van Azure Monitor, en een uitbreidbare APM [service](#) voor ontwikkelaars en DevOps professionals. Het kan gebruikt worden om live applicaties te volgen, detecteert automatisch prestatie afwijkingen en bevat krachtige analyse tools om te helpen bij het diagnosticeren van problemen en om te begrijpen wat gebruikers werkelijk met de applicatie doen. 1, 38, 39, 82

N

- native applicatie** is een applicatie dat ontwikkeld is voor het gebruik van een specifiek platform of toestel. Zodat deze toegang heeft tot het toestel of platform specifieke hardware en software.. 1, 18, 30, 32
- native component** is een platform specifiek software [component](#) waarbij platform specifieke eigenschappen gebruikt kunnen worden. 1, 19, 31
- native container** is een houder voor webpagina's waarbij [native component](#) gebruikt kunnen worden zoals de camera, GPS, enz. 1, 30
- native platform** is een platform die gericht is op specifieke mobiele platformen. Wanneer men in [React Native](#) een applicatie maakt dan kan men kiezen de native platformen Android, iOS, website en [UWP](#) te gebruiken bij de ontwikkeling wanneer men de applicatie voor deze platformen wenst te bouwen. 1, 16, 17, 20, 31, 33
- native rendering engine** is een kern software [component](#) die gebruikt wordt door mobiele applicaties op een specifiek platform dat voor het renderen van de UI zorgt. 1, 31
- netwerk applicatie** is een applicatie die op een server werkt en gebruik maakt van internet of andere netwerk hardware infrastructuur om nuttige functies uit te voeren, zoals bestandsoverdracht binnen een netwerk. Ze helpen om gegevens van het ene punt naar het andere binnen het netwerk over te dragen. Zoals data van een databank naar een website te sturen. 1, 49
- Node.js** is een open-source, [cross-platform](#) JavaScript runtime omgeving that JavaScript code uitvoert buiten de web browser. Hierdoor kunnen ontwikkelaars JavaScript gebruiken om command line tools en server side scripts te kunnen schrijven en server side scripts uit te voeren om dynamisch web pagina inhoud te produceren voordat

de pagina naar de web browser wordt gestuurd. 1, 20, 43, 49

npm is een pakket beheerder voor de JavaScript programmeertaal. Het is de standaard pakketbeheerder voor het JavaScript runtime omgeving [Node.js](#). 1, 43, 76

O

Objective-C is een objectgeoriënteerde programmeertaal voor algemene doeleinden die Smalltalk-achtige berichten toevoegt aan de C programmeertaal, en die door Apple werd ondersteund. 1, 30

operationele kost is de kost die een onderneming maakt om de operatie draaiende te houden. Deze kost komt door herhaalde betalingen zoals personeelskosten, hardware onderhouden en ook elektriciteit.. 1, 42, 70, 73, 74

P

page hit andere benaming voor [page view](#). 1, 57–60, 63, 65–67, 71, 73, 80

page view ook wel pagina weergave, [analytics tools](#) voeren hierop metingen uit waarbij het aantal weergaven wordt gemeten voor een specifieke pagina op de website. Pagina's komen vaak overeen met een volledig HTML document, maar ze kunnen ook dynamisch geladen inhoud vertegenwoordigen, dit staat bekend als virtuele page views. 1, 20, 21

Polymer is een open-source JavaScript [library](#) voor het maken van [web applicaties](#) met behulp van web [componenten](#). Het wordt ontwikkeld door Google en Github contributors. Het biedt dezelfde functionaliteiten aan als [Angular](#) maar dan door middel van web [componenten](#), hierdoor biedt het ook de mogelijkheid om zelfgemaakte elementen te maken. 1, 30

property is een verzamelpunt voor [analytics](#) data. Het is zo dat [analytics](#) een property aanziet als een bron die is gekoppeld aan de [tracking code](#). Wanneer een bron wordt opgevolgd met [analytics](#) dan wordt de property ID in de [tracking code](#) opgenomen die op de web pagina's of website plaatst. Op deze manier kan data verzameld worden en toegewezen worden aan de juiste property doordat deze data is [getag](#)d met die bepaalde ID. 1, 16, 22, 36, 37, 39, 58, 61, 62

PWA ook wel progressieve [web applicaties](#) is een soort applicatie software die via het web wordt geleverd en is gebouwd met veel gebruikte [web technologieën](#) zoals HTML, CSS en JavaScript. 1, 30

R

Rails wordt ook Ruby on Rails genoemd, het is een [web applicatie framework](#) aan de serverzijde dat is geschreven in Ruby. Het is een MVC [framework](#) dat standaard structuren aanbiedt voor een database, een [webservice](#) en webpagina's. 1, 34

React is een open-source JavaScript [library](#) voor het bouwen van UIs dat gemaakt is en onderhouden wordt door Facebook. Daarom is React dan ook geen volwaardig MVC [framework](#) aangezien het enkel functionaliteit aanbied voor het View gedeelte, het biedt wel de mogelijkheid om zelf [libraries](#) toe te voegen om het een volwaardig MVC [framework](#) te maken. 1, 17, 20, 30–33, 49

React Native is een open-source [framework](#) voor mobiele applicaties dat gemaakt is door Facebook. Het wordt gebruikt om applicaties voor Android, iOS, [UWP](#) en websites te ontwikkelen door de ontwikkelaars in staat te stellen [React](#) te gebruiken te samen met de [native platform](#) mogelijkheden. 1, 3, 5, 17, 19, 25, 29, 31–34, 37, 39, 43, 72, 74, 76, 77, 79, 81

release cyclus ook wel software release life cycle is de som van de ontwikkelingsstadia en volwassenheid van een stuk computer software. Zo variëren cycli van de initiële ontwikkeling tot de uiteindelijke release, en bevatten bijgewerkte versies van de gereleased versie om te helpen met het verbeteren van software of het oplossen van software fouten die nog in de software aanwezig zijn. 1, 15, 37

S

screen hit ook wel screen view, screenview [hit](#) of screen tracking [hit](#), is een instantie van een scherm wanneer een scherm transitie plaatsvindt en het nieuwe actieve scherm doorgeeft. Zo is het dus hetzelfde als een [page view](#), aangezien het ook een [metric](#) is maar die wordt gedefinieerd als het totaal aantal bekeken schermen. 1, 57, 58, 71, 73

SDK ook wel software development kit is een collectie van software ontwikkelings tools in één installeerbaar pakket. Deze vergemakkelijken het maken van applicaties door een compiler, debugger en eventueel een software [framework](#) te gebruiken. 1, 17, 18, 34, 36–40, 49, 71–73, 77, 81

segment is een subset van [analytics](#) data. Segmenten kunnen subverzamelingen van gegevens isoleren en analyseren zodat de trends kunnen onderzocht worden en hierop kan gereageerd worden. 1, 61, 69, 73

SEO ofwel Search Engine Optimization is het proces van het verhogen van de kwaliteit en de kwantiteit van [website verkeer](#) door de zichtbaarheid van een website of web pagina te vergroten voor de gebruikers die een web zoekmachine gebruiken. Het verwijst naar de verbetering van onbetaalde resultaten en sluit direct verkeer en de aankoop van betaalde plaatsing uit. 1, 35

service verwijst naar een software functionaliteit of een set software functionaliteiten met als doel om dit te kunnen hergebruiken voor andere doeleinden. 1, 16, 17, 19, 20, 22, 33–35, 37–39, 62, 77

SKU ook wel stock keeping unit of voorraad beheer eenheid, is een uniek nummer die onder andere in de detailhandel gebruikt wordt om een product te identificeren. 1, 67

Swift is een [gecompileerde](#) programmeertaal voor algemene doeleinden en meerdere paradigma's (programmeer concepten of technieken) ontwikkeld door Apple Inc. voor het gebruik op iOS, iPadOS, macOS, watchOS, tvOS, linux en z/OS. 1, 30

T

tag is een stuk code die wordt geleverd door [analytics](#), marketing en ondersteuning leveranciers om te helpen met het integreren van hun producten in een website of mobiele applicatie. Met [Google Tag Manager](#) is het niet meer nodig om deze tags rechtstreeks aan een project toe te voegen. Met behulp van de Tag Manager UI kunnen de tags geconfigureerd en gepubliceerd worden en hoe deze tags worden geactiveerd. 1, 17, 18, 20, 37

third-party met betrekking tot een onafhankelijk persoon, groep of organisatie naast de 2 die hoofdzakelijk zijn betrokken in een situatie. 1, 25

tool is een applicatie of programma dat gebruikt kunnen worden om te helpen met bepaalde taken. In dit onderzoek worden er [analytics](#) tools gebruikt om te helpen met het analyseren van het gebruik van websites of applicaties. 1, 3, 5, 6, 17, 20, 25–27, 29–31, 33–37, 39–43, 45, 48, 50, 57, 60, 67, 71, 72, 74, 77, 79, 81, 82, 84

tracking code een klein stukje code dat meestal is geïmplementeerd als JavaScript in de HTML bron code van een website. Deze code zorgt ervoor dat adverteerders, web

beheerders en marketeers de stroom van bezoekers naar websites en de activiteiten van gebruikers analyseren. 1, 20, 22, 25, 36, 37, 39

tracking ID is een gewone string dat moet worden opgenomen in de [tracking code](#) om [analytics](#) te laten weten naar welk account en welk [property](#) het data moet sturen. 1, 39, 61–63, 73, 82

transactie is een financiële verrichting zoals de aankoop van een product of dienst. 1, 39, 57–60, 66, 67, 71, 73, 80

TypeScript is een open-source programmeer taal ontwikkeld door Microsoft. Het is een strikte syntactische superset van JavaScript en voegt optioneel statisch typen toe aan de taal. TypeScript is ontworpen voor de ontwikkeling van grote applicaties en [transcompileert](#) naar JavaScript. 1, 15–17, 33, 34

U

UWP ook wel Universieel Windows Platform, is een computer platform en helpt bij het ontwikkelen van universele applicaties die draaien op Windows 10, Windows 10 Mobile, Xbox One en de HoloLens zonder dat het moet herschreven worden voor elk apart. 1, 19, 20

V

VDOM ook wel virtual document object model, is verschillend van de gewone dom. Het is namelijk zo dat de gewone dom de volledige structuur van de website aanpast terwijl de virtuele dom kijkt naar de verschillen tussen de vorige en de huidige HTML en verandert het onderdeel dat moet worden bijgewerkt. 1, 33

view of rapportage view is een subgroep van een [analytics](#) account [property](#) dat zijn eigen unieke configuratie instellingen kan hebben. Zo kunnen er meerdere views voor één [property](#) aangemaakt worden en elke view configureren om een andere subgroep van gegevens voor de [property](#) weer te geven. 1, 37, 61

Visual Studio is een [IDE](#) van Microsoft om computer programma's, websites, [web applicaties](#), [web services](#) en mobiele applicaties te maken. Het is een code editor die opnieuw is gedefinieerd en geoptimaliseerd voor het bouwen van moderne web en cloud applicaties. 1, 15, 43

W

web analytics is het meten, verzamelen, analyseren en rapporteren van web data om het web gebruik te begrijpen en te optimaliseren. Het is echter niet alleen een proces voor het meten van [website verkeer](#), maar het kan ook gebruikt worden als een hulpmiddel voor zakelijk en markt onderzoek, en om de effectiviteit van een website te beoordelen en te verbeteren. 1, 35, 38

web applicatie is een term die wordt gebruikt om een computer software programma te beschrijven dat op een webserver wordt uitgevoerd, in tegenstelling tot computer gebaseerde softwareprogramma's die lokaal op het besturingssysteem van een apparaat zijn opgeslagen. 1, 15, 17, 18, 20, 22, 30, 33

web technologie is een technologie die nodig is om een website te kunnen maken en tonen om ervoor te zorgen dat computers met elkaar kunnen communiceren. 1, 20, 30

WebRTC is een gratis, open-source project dat webbrowsers en mobiele applicaties realtime communicatie aanbiedt via eenvoudige API's. 1, 34

website verkeer personen die een website bezoeken en verlaten behoren tot het website verkeer van die bepaalde website. 1, 21, 22, 35, 38

WebView toont web inhoud als een deel van de mobiele applicatie. 1, 30

X

Xamarin is een gratis en open-source mobiel applicatie platform voor het bouwen van native en krachtige iOS, Android, tvOS, watchOS, macOS en Windows applicaties in C# met .NET. 1, 31

Xcode is een [IDE](#) voor macOS met een set software ontwikkelings tools voor het ontwikkelen van software voor macOS, iOS, iPadOS, watchOS en tvOS. 1, 34

1. Inleiding

[Analytics tools](#) is een veel gebruikt onderwerp binnen bedrijven de afgelopen jaren. (Trends, 2020) [Analytics tools](#) hebben dan ook een brede waaier van mogelijkheden tot (bijna) eindeloos, vooral indien deze geïntegreerd kunnen worden met [third-party tools](#).

[Analytics tools](#) zorgen voor het ophalen van gebruikersinformatie door de [tool](#) te koppelen aan de website of mobiele applicatie, door een [tracking code](#), [library](#) of plugin te gaan toevoegen. Deze informatie is redelijk uiteenlopend zoals http requests van de gebruiker, browser of systeem informatie en [first-party](#) cookies bijvoorbeeld.

Vele bedrijven zagen het potentieel in [analytics](#) zoals Google en Microsoft en hebben na verloop van tijd hier dan ook in geïnvesteerd waardoor producten zoals Google analytics en App Center Analytics ontstaan zijn. Met de opkomst van [big data](#) is [analytics](#) alleen nog maar belangrijker geworden, hierdoor is [big data analytics](#) ontstaan. Het enige nadeel dat [analytics](#) hier heeft is dat de [tools](#) er wel zijn maar het gebruikt moet worden door mensen die hier kennis van hebben zoals data specialisten. Hierdoor moeten vele bedrijven investeren in hun mensen om deze [tools](#) te kunnen gebruiken. Google investeert dan ook meer in Google Analytics zodat mensen met bijna geen kennis van [analytics](#), ook gewoon aan de slag kunnen gaan met de [analytics tool](#), zodanig dat mensen geen opleiding nodig hebben om de tool te gebruiken.

Het implementeren van [analytics tools](#) is redelijk eenvoudig, maar dit hangt af van de omgeving waarin het geïmplementeerd wordt. In dit onderzoek zullen [analytics tools](#) geïmplementeerd worden binnen [React Native](#) met [Managed Expo](#).

[React Native](#) is een [framework](#) dat gemaakt is door Facebook, waarbij er een applicatie gemaakt kan worden voor meerdere besturingssystemen (OS) dit wil zeggen [cross-platform](#), men hoeft dus geen applicatie te maken per besturingssysteem. [Expo](#) daarentegen is een

open-source platform (omgeving) dat ervoor zorgt dat de development, het inzetten en de iteratie, over de verschillende besturingssystemen van die applicatie eenvoudiger kan verlopen.

1.1 Probleemstelling

Tegenwoordig gebruikt iedereen applicaties op hun gsm waarbij de meeste applicaties dan ook gebruik maken van **analytics tools**. Ook al worden **tools** gelinkt met de applicatie er moeten nog een aantal reeksen configuraties gebeuren alvorens alle informatie kan worden opgehaald zoals **events** die gebeuren en opgevangen moeten worden. Als een bedrijf **analytics** wilt gaan gebruiken dan is de hoofdvraag dan ook welke **tool** ze moeten gebruiken.

Dit onderzoek dient dan ook om op deze vraag te kunnen beantwoorden, zoals de vraag van Cogn-IT zelf. Aangezien ze al een mobiele applicatie hadden, wouden ze hebben dat er een **analytics tool** werd geïmplementeerd die meer informatie kon bieden over de applicatie. Ze vroegen zich dan ook af welke **tool** de meest geschikte/efficiënte is zonder dat hun code moet worden aangepast.

1.2 Onderzoeksvraag

In deze scriptie zullen een aantal onderzoeksvragen behandeld worden:

- Welke **tool** biedt de meeste mogelijkheden (collecteren en verwerking van data)?
- Welke **tool** werkt het beste samen met **Managed Expo**?
- Welke **tool** is het meest eenvoudige/gebruiksvriendelijke te implementeren zonder bijkomende aanpassingen?
- Wat zijn de voor- en nadelen bij het gebruik van de **tool**?
- Welke **tool** is integreerbaar met bestaande **analytics** (momenteel **Google Analytics**)?
- Wie heeft de ownership van de data en wat met privacy (**GDPR**)?

1.3 Onderzoeksdoelstelling

De onderzoeksdoelstelling van deze scriptie is een geschikte **analytics tool** vinden die toekomstgericht voldoet aan de eisen van Cogn-IT. Het is namelijk zo dat Cogn-IT gespecialiseerd is in het bedenken, ontwikkelen en implementeren van Microsoft SharePoint en Office 365 oplossingen om medewerkers in organisaties efficiënter samen te laten werken. Ze hebben dan ook een office 365 intranet oplossing genaamd Involv, hiervan is er ook een mobiele applicatie voor ontwikkeld en Cogn-IT wenst in deze applicatie dan ook **analytics** te gaan gebruiken aangezien dit al gebeurd in de website. Men wilt door **analytics** te gaan gebruiken meer informatie verkrijgen over hoe de applicatie van Involv wordt gebruikt

door de klanten, en om zo problemen op te sporen. De onderzoeksvragen van hierboven zullen alvast de criteria vormen om de geschikte [analytics tool](#) te vinden.

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 onderzoeksdomein, op basis van een literatuurstudie.

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

In Hoofdstuk 4 wordt het aanmaken van de demo applicatie toegelicht te samen met de werking van de applicatie.

In Hoofdstuk 5 wordt de [analytics tool Expo Analytics](#) onderzocht en toegelicht.

In Hoofdstuk 6 wordt de [analytics tool Google Analytics](#) onderzocht en toegelicht.

In Hoofdstuk 7 wordt de [analytics tool App Center Analytics](#) onderzocht en toegelicht.

In Hoofdstuk 8, 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

2.1 Inleiding

In dit hoofdstuk zal de lezer meer informatie verkrijgen over [analytics](#) in [React Native](#) met [Managed Expo](#). Er zal dieper ingegaan worden op de soorten mobiele applicaties, [Expo](#) en de verschillende [analytics tools](#). Men zal na dit hoofdstuk dan ook meer inzicht verkregen hebben in de huidige stand van zaken wat betreft de mobiele applicaties, de omgeving en de [analytics tools](#).

2.2 Mobiele Applicaties

Vandaag de dag kunnen applicaties gemaakt worden op verschillende besturingssystemen, de meest gekende zijn onder andere Android en iOS. In het verleden werden applicaties gemaakt per besturingssysteem zodat het de look en feel had van dat bepaald besturingssysteem. Indien ze die applicatie voor een ander besturingssysteem wou, dan maakte ze deze volledig opnieuw met als resultaat dat deze applicatie er anders uitzag en dus de look and feel telkens verschildte afhankelijk van het besturingssysteem. Tegenwoordig kan men ook 1 applicatie maken voor meerdere besturingssystemen dit noemt men [cross-platform](#) applicaties. Dit zorgt ervoor dat de code maar 1 maal moet geschreven worden en direct op alle platformen kan werken. We zullen over de verschillende manieren om een applicatie te maken dieper op ingaan.

2.2.1 Native App

Native applicatie zijn mobiele applicaties die gemaakt zijn voor een enkel platform zoals Android of iOS, de applicaties worden gemaakt in de taal die gepromoot wordt door de makers van de platformen. Zoals Java of **Kotlin** voor Android, en **Objective-C** of **Swift** voor iOS.

Wanneer **native applicaties** gemaakt worden dan gebeurt dit in een omgeving dat de technische en gebruikerservaring richtlijnen volgt van het besturingssysteem waarvoor het gemaakt wordt. De interactie met de applicaties zullen hetzelfde aanvoelen als de andere **native applicaties** op het apparaat, zodat de gebruiker eenvoudig de applicatie kan leren gebruiken.

Ook al is dit een zeer gewilde manier om applicaties te maken vele startups durven hier niet mee aan de slag te gaan doordat deze ook veel duurder is, wanneer men dit op meerdere platformen wil hebben. (Korolev, 2019)

2.2.2 Hybrid App

Hybrid applicatie combineren **web applicatie** en **native applicaties**. Waarbij het project gemaakt is met **web technologie** zoals HTML, CSS en JavaScript. Het project wordt in een **native container** geplaatst genaamd - de **WebView**.

De **WebView** zorgt ervoor dat inhoud als een standaard website eruitziet en zorgt voor de gebruikerservaring (UX) en toegang tot de hardware functionaliteit zoals onder andere camera en GPS.

Mogelijke **tools** om **hybrid applicatie** te maken zijn onder andere **Ionic** en **Apache Cordova**. (Korolev, 2019)

2.2.3 Progressieve Web App

Progressieve Web Applicaties of **PWA's** is een relatief nieuwe term op de markt. Op een bepaalde manier zijn ze gelijkaardig aan **hybrid applicatie**, aangezien beide gebaseerd zijn op **web technologie** en in een browser worden uitgevoerd. Bij **PWA** is er geen nood tot installatie, toch is er nog steeds toegang tot de applicatie op het startscherm.

Het verschil tussen beide is dat bij **PWA** de UX aanzienlijk geoptimaliseerd is maar enkel bij Android, het werkt offline, biedt GPS-toegang, pushmeldingen en nog veel meer. Bij iOS versie 1.12.2 is dit niet toegankelijk en zijn de voordelen beperkt.

Mogelijke **tools** om **PWA** te maken zijn onder andere **Angular**, **React** en **Polymer**. (Korolev, 2019)

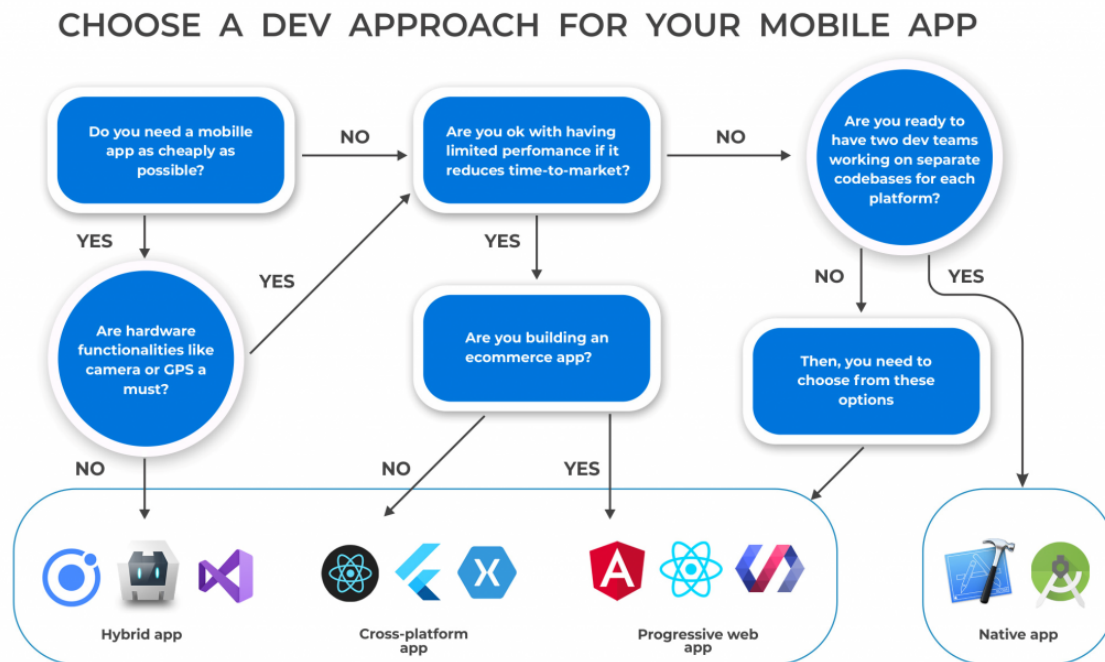
2.2.4 Cross-Platform App

Cross-platform applicaties hebben 1 gemeenschappelijk kenmerk met **hybrid applicatie**, namelijk de deelbaarheid van code, voor de rest zijn deze verschillend. **Cross-platform** applicaties gebruiken een **native rendering engine**, waarbij de code geschreven in JavaScript verbinding maakt met **native componenten**. Op deze manier wordt een bijna-Native UX aangeboden.

Mogelijke **tools** om **cross-platform** applicaties te maken zijn **React Native**, **Xamarin** en **Flutter**.

Gedurende het onderzoek wordt er gewerkt met een **cross-platform** applicatie door middel van de **tool React Native**. (Korolev, 2019)

Om de juiste keuze te maken van welke soort applicatie en omgeving je moet gebruiken, kan je onderstaand schema voor gebruiken.

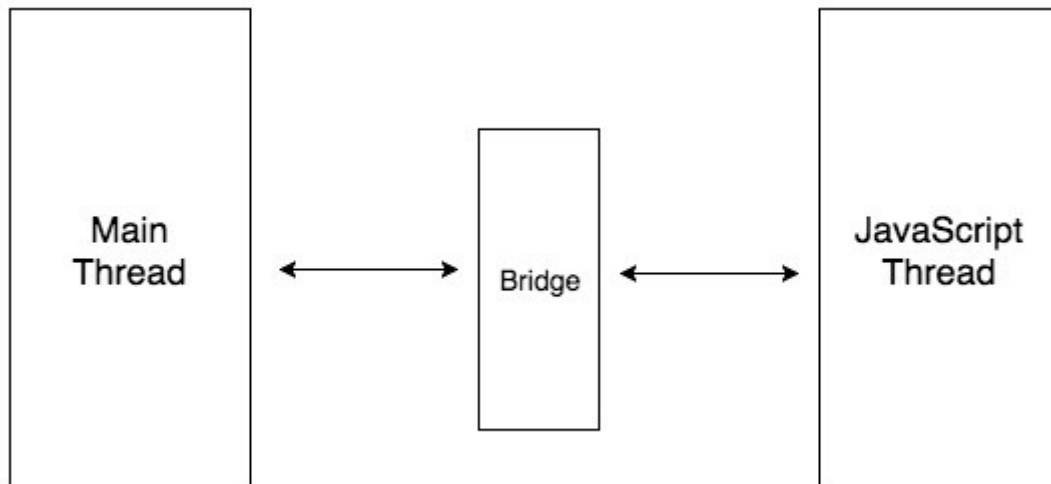


Figuur 2.1: Hoe kies je de juiste applicatie voor jouw project? (Korolev, 2019)

2.3 React Native

React Native is een **framework** om het **React framework** te kunnen gebruiken op iOS en Android, waarbij de Native UI (User Interface) gebruikt wordt en waarbij er dus volledige toegang is tot het **native platform**. **React Native** is een **cross-platform framework** en dus zo gemaakt dat de applicatie voor zowel Android als iOS gebruikt kan worden. (Facebook, 2019)

Het komt er dus op neer dat [React Native](#) een uitgebreide versie is van het [React framework](#), dat voornamelijk is gericht op het gebruik van het [React framework](#) voor mobiele platformen zoals iOS en Android.



Figuur 2.2: De architectuur van React Native (Grover, 2018)

Op deze afbeelding zie je de architectuur van [React Native](#). Om te beginnen heb je links de “Main thread” en deze draait in elke standaard [native applicatie](#). De Main thread behandelt het weergeven van de elementen van de UI en verwerkt de gebruikers gebaren, vb. zoals het swipen in de Tinder applicatie.

Het rechtse deel op de afbeelding is de JavaScript thread dit is specifiek voor [React](#) en de [React Native](#) UI Controls. Deze heeft namelijk als taak de [React](#) code uitvoeren in een aparte JavaScript engine. De JavaScript houdt zich bezig met de bedrijfslogica van de applicatie. Het definieert ook de structuur en de functionaliteiten van de UI.

De Main thread en de JavaScript thread die gaan nooit direct met elkaar communiceren of elkaar blokkeren. Om ervoor te zorgen dat er toch communicatie kan verlopen wordt er een bridge gebruikt dit is de taak van [React Native](#).

Deze bridge zorgt er namelijk voor dat beide threads [asynchroon](#) met elkaar kunnen communiceren, zodat we zeker kunnen zijn dat ze elkaar nooit gaan blokkeren. Deze communicatie verloopt dan ook gebatcht wat wil zeggen dat het communiceert van de ene thread naar de andere in een geoptimaliseerde wijze. De bridge zorgt er ook voor dat de 2 threads nooit delen of werken met dezelfde gegevens, door ze geserialiseerde berichten te laten uitwisselen.

[React Native](#) zal dus de gecompileerde [React](#) code van de JavaScript thread [asynchroon](#), geserialiseerd en gebatcht uitwisselen met de Main thread. De Main thread zal op zijn beurt de UI tonen gebruikersgebaren verwerken en terug doorsturen, zodat de JavaScript thread op de juiste manier kan antwoorden en de functionaliteiten uitvoeren.

2.3.1 React

Aangezien [React Native](#) de omgeving is waarin gewerkt zal worden en deze gebaseerd is op [React](#) is het ook belangrijk om te weten wat [React](#) zelf is wat het doet en hoe het werkt.

[React](#) zelf is een JavaScript [library](#) dat is ontwikkeld door Facebook. Dit wil zeggen dat [React](#) geen volwaardig MVC (Model View Controller) [framework](#) is aangezien het enkel de view meegeeft en de rest moet hierbij dan zelf bepaald worden afhankelijk van de noden van het project. Hierbij moet er weliswaar zelf gezorgd worden voor de nodige updates en migraties van de [libraries](#) waardoor de kans groter is dat het fout loopt.

[React](#) is een [library](#) voor het bouwen van een gebruikersomgeving of UI, hierbij kan een complexe UI opgebouwd worden in kleine en geïsoleerde stukjes code. Dit noemt men “components”, deze [componenten](#) zorgen ervoor dat er interactieve elementen opgebouwd kunnen worden zoals menu's, zoekbalken, lijsten en meer.

Voor [React](#) gebruikten ontwikkelaars voornamelijk gewoon JavaScript en gelijkaardige [libraries](#). Gelukkig heeft Facebook in 2011 [React JS](#) gemaakt om UI ontwikkelingen sneller te laten verlopen. [React](#) wordt geleverd met twee belangrijke functies namelijk [JSX](#) (JavaScript XML) en de virtuele Document Object Model ook wel [VDOM](#) genoemd, deze functies is wat [React](#) uniek maakt.

[JSX](#) is redelijk gelijkaardig aan HyperText Markup Language of HTML, alleen kan JavaScript gebruikt worden om hier dieper op in te gaan. [JSX](#) voert dus updates uit aan de DOM en doet dit met een hoge performantie en ontwikkelings efficiëntie, dit komt namelijk door de [VDOM](#). Standaard wordt HTML gebruikt om de DOM aan te passen wat goed werkt bij statische pagina's waarbij de inhoud niet verandert, bij dynamische websites met vele gebruikers interacties kan dit problematisch zijn.

Dit is waarbij de [VDOM](#) een rol speelt. Bij [React](#) wordt [JSX](#) gebruikt om de DOM te updaten, hierbij maakt [React](#) de [VDOM](#) aan. Dit is eigenlijk het maken van een copy van de DOM van de website, [React](#) gebruikt dit om te bepalen wat er van de werkelijke DOM moet aangepast worden bij een interactie. Dit wil zeggen dat bij HTML de volledige website moet vernieuwd worden bij een interactie, dit is wat [React](#) vermijdt, het vernieuwt enkel hetgene dat nodig is.

2.4 Expo

[Expo](#) is een [framework](#) en platform voor universele [React](#) applicaties en een set van [tools](#) en [services](#) dat is gebouwd voor [React Native](#) en [native platformen](#). [Expo](#) helpt daarmee applicaties ontwikkelen, bouwen, implementeren en dit snel te herhalen voor iOS, Android en [web applicatie](#) met 1 gemeenschappelijke code dat geschreven is in JavaScript of [TypeScript](#). (Expo, 2019)

Om de [Expo tools](#) te gebruiken zijn er 2 manieren namelijk “Managed” en “Bare” werkwijze. Dit wordt nader uitgelegd in volgende delen.

2.4.1 Bare Expo

Wanneer men de Bare methode gebruikt of ook wel [Bare Expo](#) dan heeft de ontwikkelaar de volledige controle over elk aspect van het Native project, waarbij de [Expo tools](#) niet echt kunnen helpen.

Dit zorgt dan ook voor de nodige complexiteit, ook al worden de meeste APIs van de [Expo SDK](#) gebruikt, niet alles wordt momenteel dan ook ondersteund.

Onderdelen zoals de [build service](#), meldingen, draadloze updates en makkelijke configuraties met app.json worden nog niet ondersteund, maar zullen mogelijk gemaakt worden in toekomstige updates. (Expo, 2019)

2.4.2 Managed Expo

De tweede mogelijkheid dat [Expo](#) aanbiedt is [Managed Expo](#), bij deze methode schrijft de ontwikkelaar enkel JavaScript of [TypeScript](#) en zorgen de [Expo tools](#) en [services](#) voor de rest. Deze methode lijkt op “[Rails](#)” en de “Create React App” voor [React Native](#).

Wanneer applicaties gemaakt worden met deze werkwijze dan gebruikt dit de expo-cli, de [Expo client](#) op uw mobiel apparaat en de verschillende [services](#) die ze aanbieden.

De bedoelingen hiervan is zoveel mogelijk van de complexiteit van het bouwen van applicaties te beheren, daarom noemt het dan ook “Managed” Expo. Het voordeel hiervan is dat programma’s zoals [Xcode](#) en [Android Studio](#) niet nodig zijn. (Expo, 2019)

Bij het gebruik van [Expo](#) wordt standaard [Managed Expo](#) gebruikt en aangeraden doordat dit de meest eenvoudige wijze is om een mobiele applicatie te ontwikkelen binnen Expo. Wanneer [Managed Expo](#) is gekozen kan men nog steeds het project “[ejecten](#)” door “expo eject” in de terminal van het project uit te voeren. Hierdoor wordt de geavanceerde configuraties en Native code getoond / openbaar gemaakt. Een mogelijke reden om toch naar [Bare Expo](#) over te stappen is indien er voornamelijk configuraties moeten gebeuren op het Native niveau, of die niet mogelijk zijn in de Managed workflow.

Er zijn dan ook verschillende limitaties in het gebruik van [Managed Expo](#). Zo worden niet alle API’s van Android en iOS ondersteund, zo als bluetooth, [WebRTC](#), In-App aankopen en Apple en Google Play integratie. Ook is de Managed workflow niet voordelig wanneer men de applicatie zo klein mogelijk wenst te houden doordat dit verschillende API’s toevoegt ook al worden deze niet gebruikt. Zo zijn er nog een aantal limitaties die het gebruik van [Managed Expo](#) kunnen beperken. Deze limitaties zijn weliswaar allemaal punten waaraan gewerkt zal worden.

2.5 Analytics Tools

Vandaag de dag zijn er enorm veel [analytics tools](#) beschikbaar afhankelijk van wat er geanalyseerd moet worden. (TrustRadius, 2020) Gedurende dit onderzoek gebruiken we [web analytics](#), dit is het bezoekersgedrag meten en analyseren op websites of mobiele applicaties. Het doel hiervan is om meer mensen aan te trekken, nieuwe klanten voor goederen of diensten te behouden of aan te trekken, of om het financieel volume dat elke klant uitgeeft te vergroten.

Door gebruik te maken van deze [analytics](#) kan er ook bepaald worden wat de gebruikers zoeken, kopen of bekijken en op basis van deze informatie geschikte advertenties en informatie tonen. Zo kan er ook gezien worden op welke manier een klant op een bepaalde pagina is terechtgekomen, of waar de gebruiker de website verlaten heeft.

Op deze manier kan er beter ingezet worden op de kwaliteit van de website en [SEO](#) (Search Engine Optimization of Zoekmachineoptimalisatie) beter afstellen op wat de gebruikers zoeken. (Hughes, 2019) [SEO](#) is dan ook het proces om de kwaliteit en kwantiteit van het [website verkeer](#) te verbeteren zodat deze hoger scoort in de zoekresultaten van een zoekmachine.

Gedurende dit onderzoek zullen we vergelijken met 3 [tools](#) met name [Google Analytics](#), [App Center Analytics](#) en [Expo Analytics](#).

[Expo Analytics](#) is weliswaar een plugin voor [Google Analytics](#), maar dit wordt onderzocht omwille van de omgeving waarin er gewerkt wordt.

2.5.1 Google Analytics

Gedurende dit hoofdstuk wordt er uitgelegd wat [Google Analytics](#) is, hoe het gebruikt wordt en hoe deze data geanalyseerd wordt.

Wat is Google Analytics?

[Google Analytics](#) is een gratis [web analytics service](#) dat wordt aangeboden door Google om mensen te helpen met het [website verkeer](#) te analyseren. Voor veel mensen lijkt het alsof [web analytics](#) maar een klein aandeel heeft in de hedendaagse digitale aanwezigheid, maar [Google Analytics](#) zijn aanwezigheid is groter dan de meeste denken, doordat deze voornamelijk op de achtergrond van websites werken en hierdoor ook niet opgemerkt worden.

Voor de meeste bedrijven dient hun website als een centraal ontmoetingspunt voor een verscheidenheid aan gebruikers gaande van klanten, leveranciers, enz. Deze gebruikers maken deel uit van het dagelijkse digitale verkeer dat zich op een website plaatsvindt. Wanneer hier [marketing activiteiten](#) worden uitgevoerd, zoals zoekadvertisenties of sociale media-advertisenties, dan zullen de gebruikers hoogstwaarschijnlijk de website ergens bezoeken tijdens het surfen op het internet.

Wanneer de website de centrale ontmoetingsplaats is van de digitale aanwezigheid, dan is de website de beste manier om een algemeen beeld te geven van de effectiviteit van de [campagnes](#) die uitgevoerd worden om de producten of diensten online te promoten. Dit wil zeggen dat [Google Analytics](#) een gratis [tool](#) is waarbij de digitale [marketing effectiviteit](#) bijgehouden kan worden. (Su, 2017)

Hoe werkt Google Analytics?

Om [Google Analytics](#) te gebruiken is het nodig om een aantal regels [tracking code](#) in de code van de website te plaatsen. Deze code registreert hiermee de verschillende activiteiten van de gebruikers wanneer ze de website bezoeken, samen met de [metadata](#) van die gebruikers zoals de leeftijd, geslacht en interesses. Waarna het deze gegevens stuurt naar de [Google Analytics](#) server zodra de gebruiker de website verlaat.

Deze gegevens worden door [Google Analytics](#) verder verzamelt op meerdere manieren, Google (2019) heeft deze ingedeeld in vier niveaus:

- Gebruikersniveau: dit is gerelateerd aan acties van elke gebruiker
- Sessieniveau: elk individueel bezoek
- Pageview-niveau: elke bezochte individuele pagina
- Evenementniveau: dit is gerelateerd met het klikken op knoppen, videoweergaven, enz.

Wanneer een bedrijf [Google Analytics](#) wilt gebruiken dan kan er op [Google Analytics Academy](#) tutorials gevolgd worden om een beter zicht te hebben op hoe [Google Analytics](#) werkt. De persoon die deze tutorials volgt krijgt op het einde ervan een certificaat als bewijs dat de persoon dit volledig heeft afgerond en dus een goede kennis heeft hoe dit werkt.

[Google Analytics](#) is een platform dat data verzamelt en [compileert](#) naar rapporten.

Om een website op te volgen, moet er eerst een [Google Analytics](#) account aangemaakt worden. Dit account heeft meestal een algemene naam waarvoor het gebruikt wordt, er moet dan ook geselecteerd worden voor welke platformen de tracking zal werken, ofwel enkel website, ofwel enkel app, ofwel de combinatie hiervan. Bij het aanmaken van een account wordt er gevraagd om een [property](#) aan te maken dat de naam heeft van de website of app, de categorie waarvoor het dient, de tijdzone waarin de rapporten gemaakt/ingedeeld worden en de valuta waarin gewerkt. (Google, 2019)

Wanneer er is gekozen om websites en applicaties of enkel applicaties te gaan analyseren, dan is het nodig om eerst een [data stream](#) aan te maken voor het gewenste platform.

Er zijn 3 soorten [data streams](#) voor iOS app, Android app en website.

Deze [data streams](#) zorgen ervoor dat data verzameld kan worden voor een specifieke applicatie of website. Voor de applicaties is het nodig dat deze via de [Firebase SDK](#)

gekoppeld worden omdat deze de naam van de applicatie kent te samen met de gekoppelde identiteit.

Hetgene dat gekozen moet worden bij het tracken van een [React Native](#) applicatie is website. Hierbij moet er een url worden meegegeven zodat [Google Analytics](#) weet van waar er data verwacht moet worden. Daarna moet er een klein stukje JavaScript [tracking code](#) gebruikt worden op elke pagina van de website, deze is te vinden waar de data stream wordt aangemaakt.

Bij [Google Analytics](#) zijn er twee manieren om deze code toe te voegen ofwel wordt de [Global Site Tag](#) gebruikt ofwel de [Google Tag Manager tool](#) om de [tags](#) te kunnen beheren. Dit is namelijk 1 van de voordelen van [Google Analytics](#), het integreren van andere [Google tools](#) om betere en meer gedetailleerde informatie over de website te hebben.

Wanneer de code is toegevoegd dan zal deze code, telkens dat een gebruiker de pagina bezoekt, anonieme informatie van de gebruiker collecteren over hoe de gebruiker met die pagina is omgegaan. Deze [tracking code](#) gaat informatie over de pagina doorgeven gaande van hoeveel gebruikers een bepaalde pagina hebben bezocht tegenover een andere pagina of hoeveel gebruikers die een aankoop deden werkelijk een bevestigingspagina zagen. Daarnaast geeft deze code ook informatie over de taal, het soort browser, het toestel, het besturingssysteem (OS), en de bron van waar de gebruiker is gekomen. (Google, 2019)

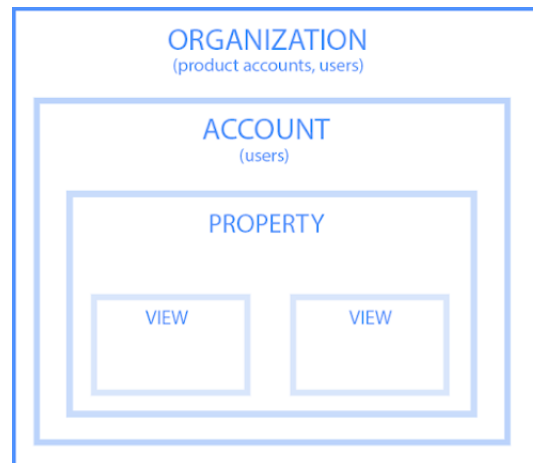
[Google Analytics](#) gebruikers moeten wel rekening houden met het feit dat als data verwerkt wordt, deze wordt opgeslagen in een database en kan niet meer worden gewijzigd.

Wanneer [Google Analytics](#) data verzamelt dan kan die data bekeken worden in verschillende rapporten binnen 1 [property](#) en deze kan verder onderverdeeld worden in verschillende [views](#) waarbij een filter toegepast kan worden. Wanneer een [property](#) is aangemaakt wordt er standaard ook een [view](#) aangemaakt om alles makkelijk en overzichtelijk te houden is het beter om die [view](#) te hernoemen naar Raw Data aangezien het alle data bevat waarbij geen aanpassingen hebben plaatsgevonden. [Google Analytics](#) raad dan ook aan om 2 extra [views](#) bij te maken, 1 Test [view](#) waarbij filters uitgetest kunnen worden en 1 Master [view](#) om de dagelijkse rapportering te kunnen maken. Hoe deze hiërarchie is opgebouwd kan je zien in onderstaande afbeelding. Het “Organization” gedeelte is weliswaar optioneel.

Om [Google Analytics](#) te gebruiken voor websites volstaat het dus om een aantal regels op elke pagina te plaatsen, maar er kan ook code aan toegevoegd worden om custom [events](#), [dimensions](#) en [metrics](#) etc. te gebruiken. Hiermee kan er geavanceerde rapporten gemaakt worden zoals het bepalen van de gemiddelde laad tijd.

2.5.2 Microsoft Visual Studio App Center Analytics

[App Center Analytics](#) is een [service](#) dat [App Center](#) aanbiedt door gebruik te maken van een [library](#) dat deel uitmaakt van de [App Center SDK](#). [App Center](#) is een continue integratie, levering en leer oplossing voor iOS en Android applicaties, het zorgt voor snellere [release](#)



Figuur 2.3: De hierarchie van Google Analytics (Google, 2020)

[cycli](#), hogere kwaliteit en inzichten om te maken wat de gebruikers willen. (Microsoft, 2019)

[App Center Analytics](#) is hetgene dat de inzichten geeft in wat de gebruikers willen en wat ze doen. Het is zoals [Google Analytics](#) een [web analytics service](#) dat helpt met het [website verkeer](#) te analyseren en om gebruikersgedrag en klantbetrokkenheid te begrijpen om de website of app hierdoor te verbeteren.

Doordat [App Center Analytics](#) een [library](#) is, dat deel uitmaakt van de [App Center SDK](#), hoeven gebruikers enkel deze te installeren in het project en kunnen ze door een aantal regels code deze gebruiken. Wanneer gebruikers een beter inzicht willen verkrijgen met de data van [App Center Analytics](#) is het mogelijk deze te exporteren naar [Microsoft Azure Application Insights](#). (Elkin, 2018)

Alvorens [App Center Analytics](#) te gebruiken in de applicatie moet er eerst via [App Center](#) een applicatie worden aangemaakt door een naam, besturingssysteem en platform op te geven. De [app secret](#) kan gevonden worden door naar de settings te gaan en dan rechts boven op het menu (drievoudige verticale stippen) te klikken en “Copy app secret” aan te klikken. (Elkin, 2018; D. Microsoft, 2019)

Om [App Center Analytics](#) in de app te kunnen gebruiken kan je in de code deze opstarten aan de hand van de [app secret](#), die online te vinden is, en door te geven welke soorten van de [App Center SDK](#) er gebruikt zullen worden. (Microsoft, 2019; Oladapo, 2019)

Om gebeurtenissen ([events](#)) op te volgen hoeft men enkel [analytics](#) op te roepen en te zeggen een [event](#) op te volgen waaraan een tekst wordt doorgegeven als de applicatie op dat punt komt. (Elkin, 2018)

2.5.3 Expo Analytics

[Expo Analytics](#) is een [library](#) dat afkomstig is van derden en dus niet van [Expo](#) of [Google Analytics](#) is. Deze plugin zorgt ervoor dat [Google Analytics](#) gebruikt kan worden binnen

een [React Native](#) omgeving met [Managed Expo](#) zonder enige vorm van [linking](#) aangezien [Managed Expo](#) dit ook niet ondersteunt.

[Expo Analytics](#) kan gebruikt worden wanneer in [Google Analytics](#) een [property](#) is aangemaakt voor websites. Hierdoor heeft de [property](#) een id dat begint met UA wat herkend wordt door deze plugin, wat niet gebeurt indien er gekozen wordt voor applicaties of de combinatie van websites en applicaties. Wanneer er gekozen wordt voor website dan moet er een url meegegeven worden en deze is vrij te kiezen want het maakt niet uit welke url er gebruikt wordt, doordat de [property](#) id voldoende is om informatie door te sturen.

Indien de [property](#) afgewerkt is dan kan er in de applicatie expo-analytics geïnstalleerd worden en nadien ook aangemaakt worden in de code door de [tracking ID](#) mee te sturen. De [tracking ID](#) kan gevonden worden wanneer de [property](#) geselecteerd is, er geklikt is op Tracking info en vervolgens op [tracking code](#).

Standaard zal [Expo Analytics](#) ervoor zorgen dat informatie zoals scherm resolutie, app naam, ID, versie en andere parameters automatisch wordt doorgestuurd met elke [hit](#) of [event](#).

Daarna kan de [analytics](#) gebruikt worden om data door te sturen naar [Google Analytics](#) zoals een [hit](#), custom (zelfgemaakte) [events](#), custom [dimensions](#) (attributen van data) en [metrics](#) (kwantitatieve metingen), maar ook [e-commerce transacties](#). (ryanvanderpoel, 2019)

2.6 Google Analytics vs. App Center Analytics vs. Expo Analytics

Dit hoofdstuk worden enkele verschillen tonen tussen de 3 [tools](#).

In deze tabel kan je zien dat zowel [Google Analytics](#) als [App Center Analytics](#) enkel gebruikt kunnen worden als de applicatie gelinkt ([linking](#)) wordt. Dit kan dus een probleem vormen wanneer we dit willen implementeren aangezien dit niet mogelijk is met [Managed Expo](#). Bij het bekijken van de [Expo Analytics library](#) is er opgemerkt dat deze [Google Analytics](#) gebruikt en hier rond een eigen implementatie heeft gemaakt, dit wil zeggen dat we mogelijk een eigen implementatie kunnen maken van [Google Analytics](#).

Verder kunnen we ook zien dat er een nadeel is bij [App Center Analytics](#) wanneer deze uitgebreid gebruikt zou worden aangezien er maar 20 custom [events](#) aangemaakt kunnen worden en [properties](#), maar 200 unieke namen kunnen hebben. Bovendien is er geen mogelijkheid om custom [metrics](#) en [dimensions](#) te maken.

Het nadeel bij het gebruik van [Expo Analytics](#) is dat er geen mogelijkheid is voor integraties terwijl [Google Analytics](#) deze mogelijkheid aanbiedt. Zo kan [App Center Analytics](#) ook geen andere [analytics](#) integreren, enkel [services](#) die deel uitmaken van de [App Center SDK](#), behalve het exporteren naar [Microsoft Azure Application Insights](#).

	Google Analytics	App Center Analytics	Expo Analytics
Nodig voor gebruik	Firebase SDK	App Center SDK	expo-analytics library
Linking	ja	ja	nee
Account dat word gebruikt	Google Analytics	App Center	Google Analytics
Hits	ja	ja	ja
Custom Events	ja	ja	ja
Custom dimensions	ja	nee	ja
Custom metrics	ja	nee	ja
Limitaties	nee	custom events 20 properties, 200 unieke namen	nee
Integraties	ja, Google producten zoals Tag Manager	ja, andere services van de SDK en export naar Azure Application Insights	nee

Tabel 2.1: Verschillen tussen Google Analytics, App Center Analytics en Expo Analytics

Als we op basis van deze tabel de [tools](#) een rang geven dan staat [App Center Analytics](#) als laatste omwille van de limitaties en [linking](#), [Google Analytics](#) met [SDK](#) (dus met [linking](#)) op de tweede plaats en is [Expo Analytics](#) de beste keuze. Indien er een eigen implementatie van [Google Analytics](#) wordt gebruikt te samen met integraties dan staat deze op de eerste plaats en staat [Expo Analytics](#) op de tweede plaats. Als een eigen implementatie gebruikt wordt, maar zonder integraties dan staat deze gelijk met [Expo Analytics](#), omdat een eigen implementatie zonder integraties hetzelfde is als [Expo Analytics](#).

Nu zullen we dan ook deze [tools](#) onderzoeken en kijken of deze voorspelling werkelijk klopt.

3. Methodologie

Alvorens dit onderzoek te bespreken zullen we eerst werkwijze bekijken. Dit onderzoek is opgedeeld in een aantal grote punten:

1. Literatuurstudie
2. Demo Applicatie
3. Onderzoek per tool:
 - (a) Data en functies
 - (b) Implementatie tool en Architectuur
 - (c) Documentatie
 - (d) Prijs / Kwaliteit
 - (e) Case studies / Getuigenissen

Dit onderzoek is net als alle andere onderzoeken begonnen met een literatuurstudie over het onderzoeksdomein. Deze literatuurstudie is te vinden in Hoofdstuk 2 Hierbij wordt de omgeving en [analytics tools](#) grondig besproken en uitgelegd zodat iedereen kan begrijpen wat het is en waarvoor het dient.

Na deze literatuurstudie wordt er een demo applicatie gebouwd, deze applicatie zal de basis vormen om de 3 [tools](#) te implementeren en te testen. Daarom zal de structuur eerst besproken worden te samen met de architectuur. Daarna zal het opzetten van de omgeving en het maken van de applicatie uitgelegd worden. Deze applicatie is een kleine, simpele en werkende applicatie genaamd Recipe Chef. Dit zal een lijst van recepten tonen waarbij men een recept kan bekijken en liken, maar ook 'aankopen' er is ook een navigatie structuur aangemaakt zodat de belangrijkste scenario's aanwezig zijn om [analytics](#) op toe te passen. Zoals in de inleiding (zie Hoofdstuk 1) is vermeld, is er gevraagd om [Google Analytics](#), [App Center Analytics](#) en [Expo Analytics](#) te onderzoeken.

Wanneer de demo applicatie is afgewerkt dan wordt deze 3 maal gekopieerd zodat elke **tool** apart geïmplementeerd kan worden. Bij de implementatie wordt ervoor gezorgd dat de code van de demo applicatie niet aangepast hoeft te worden of toch niet veel. Verder wordt er gekeken naar de architectuur zodat de **analytics tool** zo veel mogelijk los van de applicatie staat en dus overal op eenvoudige wijze gebruikt kan worden.

Na de implementatie wordt er bekeken of alle data goed wordt aangegeven in de dashboards op het internet, indien dit niet gebeurt wordt er onderzocht welke aanpassingen doorgevoerd moeten worden om deze werkende te krijgen. Zodat we nadien kunnen bespreken welke functies de **tool** heeft, welke functies werken, welke data in de dashboards getoond worden en welke niet. Daarnaast wordt er ook onderzocht naar de integratie mogelijkheden dat deze **tool** heeft.

Wanneer de data en functies zijn onderzocht zal er gekeken worden naar de documentatie. Dit gedeelte is het belangrijkste, want indien het gebruik van de **tool** niet voldoende gedocumenteerd is dan kan dit ervoor zorgen dat er veel onduidelijkheden ontstaan en dus regelmatig problemen opduiken die niet eenvoudig opgelost kunnen worden. De reden waarom dit na de implementatie, data en functie onderzoek komt is omdat het eenvoudiger is om te oordelen of deze documentatie voldoende was om de **tool** werkende te krijgen, het te onderzoeken en/of er problemen opdoken dat met de documentatie opgelost kon worden.

Daarna volgt de prijs bepaling wat voornamelijk een **operationele kost** is aangezien de **analytics tools** gratis zijn. Er wordt gekeken naar wat het product verder aanbiedt ondersteuning platformen, cloud, enz. en dit te gaan vergelijken met de prijs om te bepalen welke het voordeligst is.

Wanneer de prijs / kwaliteit is onderzocht dan wordt er een aantal case studies / getuigenissen bekeken en besproken. Hieruit kunnen we zien of dit wordt aangeraden of afgeraden en waarom, dit kunnen we met de implementatie gaan vergelijken en kijken of dit werkelijk zo is of niet.

Wanneer de 3 **tools** zijn onderzocht dan zullen we deze met elkaar vergelijken en bepalen welke het meest geschikt is over de volledige lijn. Uit deze vergelijking wordt dan een conclusie gevormd dat alle onderzoeksvragen beantwoorden.

4. Demo Applicatie

4.1 Inleiding

Om te weten welke [tool](#) het meest geschikt is binnen een [React Native](#) met [Managed Expo](#) applicatie, moeten we eerst een demo applicatie bouwen om de [analytics tools](#) hierin te gaan onderzoeken. Daarom is er ook beslist om een kleine applicatie te bouwen die een aantal functionaliteiten heeft zodat de [analytics tools](#) volledig getest kunnen worden met die functionaliteiten. Om deze applicatie op te bouwen zijn er een aantal dingen nodig:

- Android of iOS toestel of emulator
- [Visual Studio](#) code of een andere text editor
- [Node.js](#), [npm](#), [Expo](#) en andere [libraries](#) (deze zullen geïnstalleerd worden bij het Opzetten Van De Omgeving)

Deze applicatie is voor het bekijken en liken van recepten genaamd Recipe Chef. Het is een kleine applicatie met niet veel functionaliteit. Er is een Tabblad Navigatie structuur met 3 tabbladen namelijk een lijst van recepten, persoonlijke recepten, en het opzoeken van recepten op basis van overblijvende ingrediënten enkel de lijst van recepten is werkende de andere 2 hebben geen functionaliteit.

Wanneer de gebruikers de applicatie starten zullen ze eerst een lijst van recepten zien en krijgen ze keuze om ofwel te navigeren naar een ander tabblad ofwel een recept te liken ofwel een recept aan te klikken en de details te bekijken. Bij het bekijken van de details kan een gebruiker terugkeren naar de lijst van recepten of kan men het recept zagezegd aankopen. Wanneer er een recept wordt aangekocht dan wordt deze doorgestuurd naar het tabblad met de persoonlijke recepten. Hoe de applicatie eruit ziet kan je bekijken na het deel “Maken van de applicatie”.

In het volgende deel zal het verloop van de applicatie dan ook worden aangetoond, daarnaast wordt er beschreven hoe de demo is opgebouwd en wordt er aangetoond waar **analytics** op toegepast wordt.

4.2 Architectuur

Voor we de omgeving en demo applicatie gaan opbouwen, moeten we eerst eens kijken naar de architectuur van de applicatie zodat we weten hoe alles met elkaar zal werken en communiceren.

Om deze architectuur te bespreken zullen we gebruikmaken van het C4 model¹⁾ dat wordt gebruikt om software architectuur te gaan visualiseren. Het bestaat eigenlijk uit 4 verschillende lagen namelijk:

- Context: Een systeem context diagram toont hoe het software systeem past in de wereld eromheen. Zodat het de connectie met de mensen en de al bestaande software systemen kan aantonen.
- Container: Een container diagram zoomt in op het software systeem en toont de hoogwaardige technische bouwstenen. Zodat het de connectie met de andere bouwstenen, de mensen en de al bestaande software systemen kan aantonen.
- Component: Een component diagram zoomt in op een individuele container en toont de **componenten** erin. Op deze manier toont het de interne werking van deze container, de connectie met andere containers, met de mensen en al bestaande software systemen aan.
- Code: Een code diagram zoomt in op een afzonderlijk **component** en toont hoe deze is geïmplementeerd. (Dit diagram wordt niet aangeraden zelf te maken aangezien dit gegenereert kan worden).

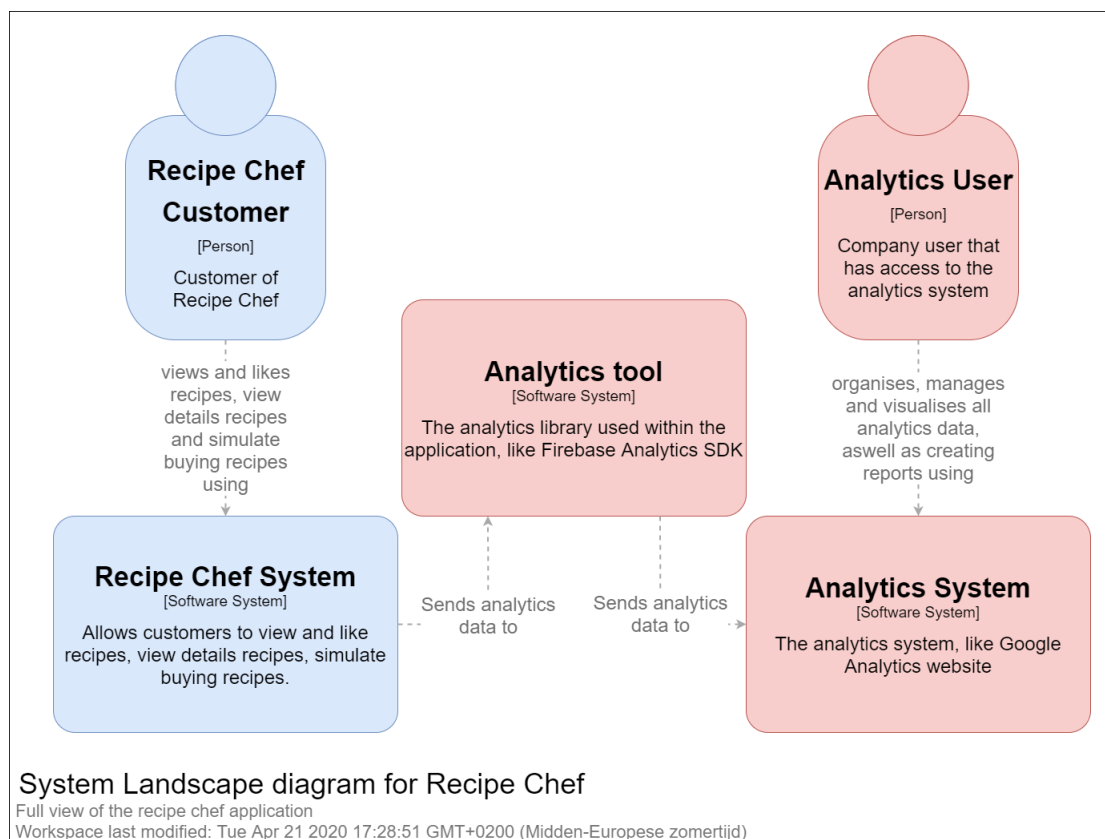
Laten we beginnen met een volledig overzicht van het systeem.

¹Meer info over het C4 model is te vinden op <https://c4model.com>

4.2.1 Systeem Context Diagram

Op het volgende diagram kan je zien hoe flow van het systeem verloopt. De Recipe chef klanten maken gebruik van de applicatie waarbij **analytics** data wordt gegenereerd en bekeken kan worden door het bedrijf in het **analytics** systeem.

Hetgene dat in het blauw staat is alles betreffende de applicatie en de personen die het gebruiken, het rode gedeelte is over het **analytics** gedeelte. Het rode gedeelte behoort dus niet tot de omvang van het project en is puur om de communicatie met deze **analytics tools** aan te duiden. Het is mogelijk dat wanneer we de specifieke **analytics tools** bespreken dat de communicatie tussen de applicatie en de **tool** verschilt met de volgende diagrammen. Dit zal afhangen van de mogelijkheden dat de **tool** zal aanbieden. In volgende diagrammen zal daarom dan ook getoond worden waar we wensen **analytics** te gaan gebruiken

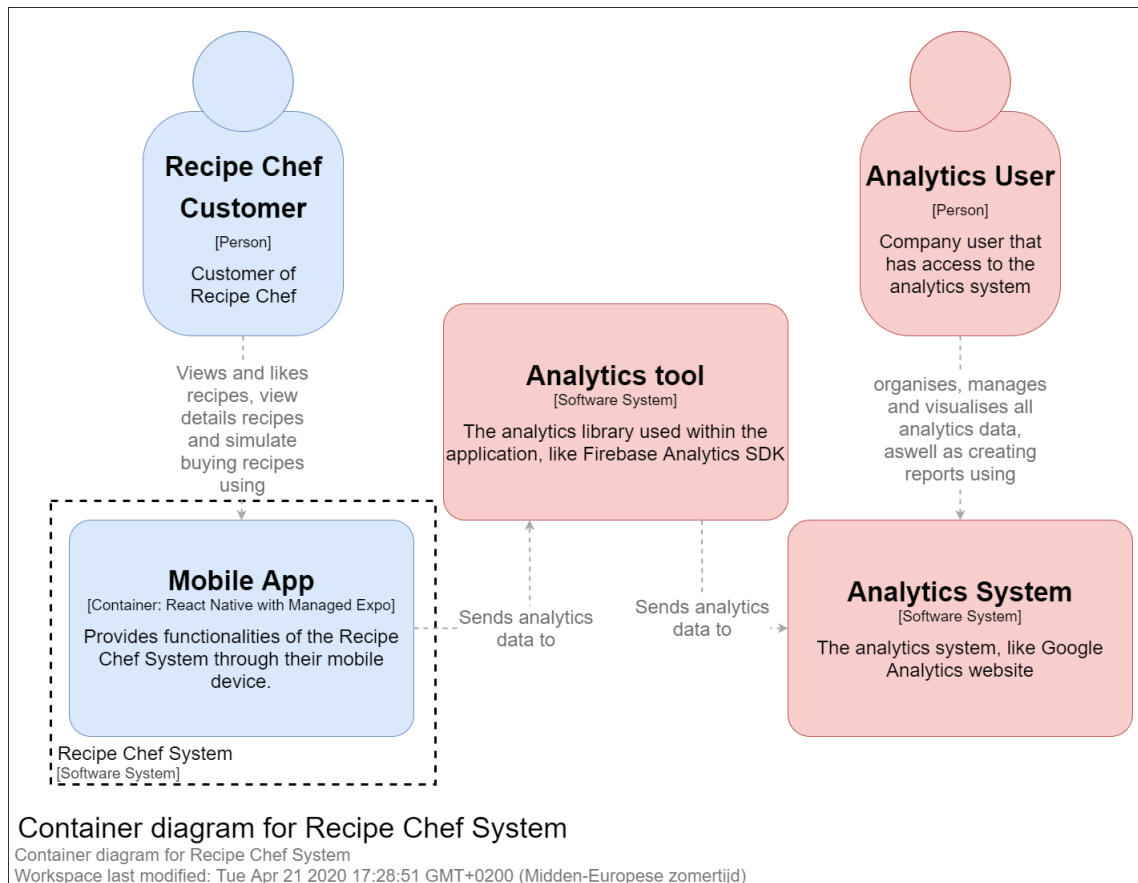


Figuur 4.1: Systeem Context diagram

Laten we het Recipe Chef Systeem van dichterbij bekijken.

4.2.2 Container Diagram

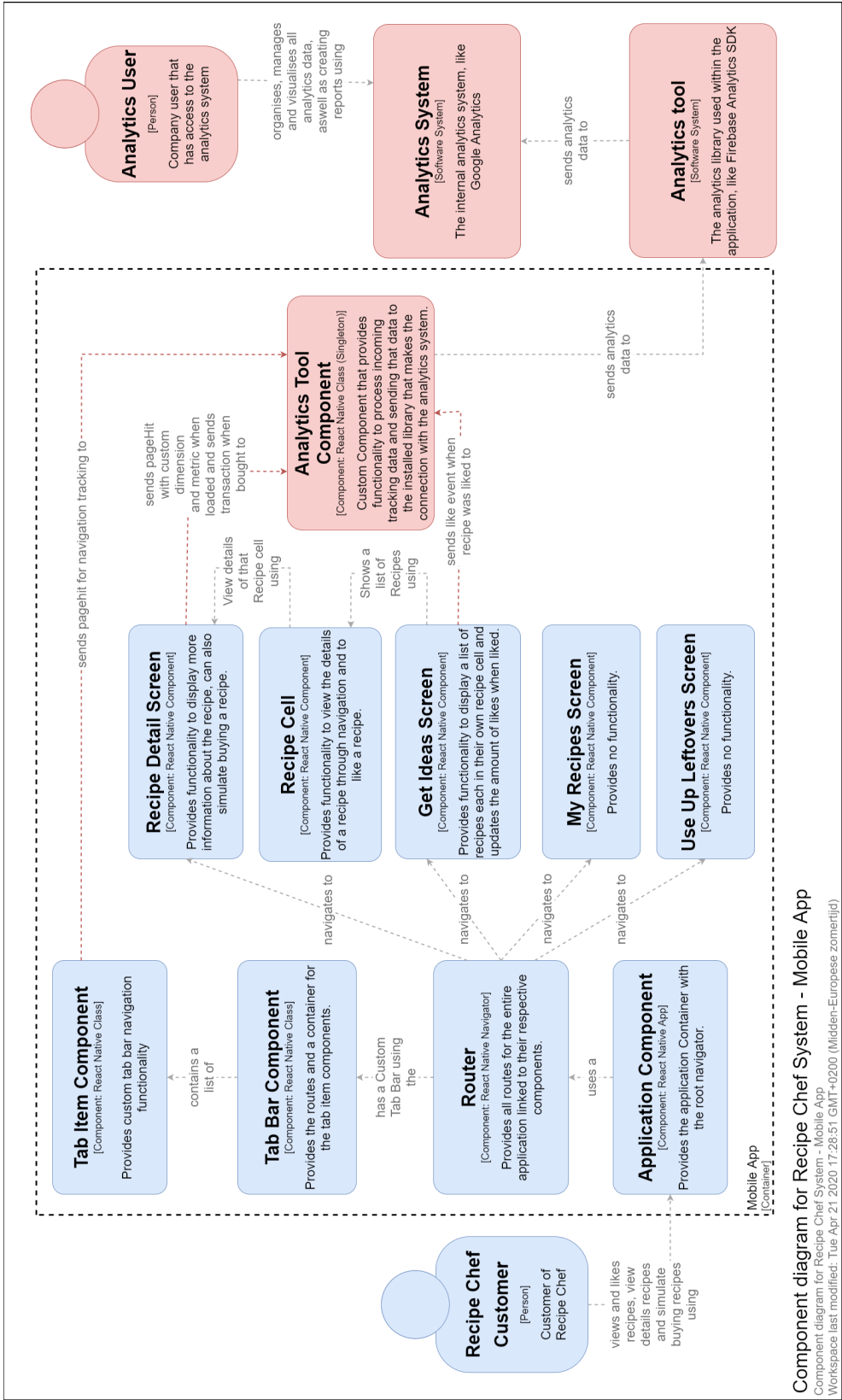
Op het volgende diagram kan je zien waaruit het Recipe Chef Systeem bestaat. Doordat het een simpele demo applicatie is bestaat het maar uit een container namelijk de mobiele applicatie. Als dit een normale applicatie is dan zou er hierbij ook de API applicatie en de database bijkomen.



Figuur 4.2: Container diagram

Laten we nu eens de mobiele applicatie van dichterbij bekijken. Aangezien de demo applicatie redelijk klein is en er dus geen indeling is in verschillende groepen, waarbij er per groep een bepaalde functionaliteit wordt uitgevoerd, zullen de verschillende onderdelen van de applicatie rechtstreeks in dit diagram komen. Waardoor de Code diagram ook niet weergegeven zal worden aangezien alle onderdelen al in het Component diagram aanwezig zullen zijn.

4.2.3 Component Diagram



Figuur 4.3: Component diagram

Op deze afbeelding kunnen we de volledige structuur van de applicatie zien.

Laten we beginnen bij het begin, de Recipe Chef Customer is een klant dat de mobiele applicatie gebruikt om recepten te bekijken, te liken, maar ook om details van de recepten te bekijken en de aankoop van recepten te kunnen simuleren.

Bij het gebruik van deze mobiele applicatie begint men met het applicatie [component](#). Dit bevat de applicatie container waarin de volledige applicatie zal weergegeven worden, hierin zal dan ook de hoofd navigatie ingestoken worden door gebruik te maken van de Router.

Deze Router geeft de hoofd navigatie terug waarin de Tabblad navigatie terug te vinden is.

De Tabblad navigatie bevat 4 schermen waarvan 1 niet getoond wordt, My Recipes, Use Up Leftovers en Get Ideas deze 3 schermen zijn de 3 tabbladen die te zien zijn op het scherm. My Recipes en Use Up Leftovers bevatten geen functionaliteit en zijn enkel toegevoegd om navigatie mogelijkheden te testen. Zoals vermeld is er 1 scherm dat gekoppeld ligt met de tabbladen, maar niet getoond wordt en dat is Recipe Detail dat is om ervoor te zorgen dat de tabbladen zichtbaar blijven wanneer de details bekeken worden.

De Tabblad navigatie bevat een zelfgemaakt Tab Bar [component](#) dat de routes en een container aanbiedt voor het zelfgemaakt Tab Item [component](#) om de tabbladen juist terug te geven. Dit [component](#) zorgt ervoor dat het juiste icoon en naam wordt weergegeven in de Tab Bar en dat de navigatie correct verloopt dit is later belangrijk om de navigatie te meten voor de [analytics tools](#).

Het Get Ideas scherm biedt 2 functionaliteiten aan, namelijk een lijst van recepten tonen en updaten nadat er 1 is geliket en dit dan ook door te geven aan de [analytics tools](#). De lijst van recepten is een lijst dat de zelfgemaakte Recipe Cell bevat voor elk recept. Deze Recipe Cell heeft als functie het weergeven van informatie over het recept, het liken van recepten en dit terug te geven aan het Get Ideas scherm om het te updaten, maar ook om te navigeren naar het Recipe Detail scherm voor meer informatie.

Het Recipe Detail scherm zorgt ervoor dat er meer informatie over het recept bekeken kan worden, maar ook dat de aankoop van een recept gesimuleerd kan worden voor de [analytics tools](#). Door het aanklikken van de afbeelding zal er naar My Recipes genavigeerd worden bij deze navigatie zal er een zogenaamde aankoop worden verstuurd naar de [analytics tools](#). Wanneer Recipe Detail wordt geladen dan zal er ook [analytics](#) data worden verstuurd om aan te tonen dat een bepaald recept is bekeken en van welke groep deze afkomstig is om zelfgemaakte data verzameling te kunnen uittesten.

Wanneer er data wordt verzameld voor de [analytics tools](#) dan zal dit gestuurd worden naar het [analytics tool component](#) dit zal gedurende het gebruik van de applicatie informatie doorsturen naar de [analytics tool \(library\)](#) die is geïnstalleerd. Deze zal de data naar het [analytics](#) Systeem waarmee het verbonden is doorsturen. Zo zal [Expo Analytics](#) de [tool](#) zijn in de applicatie die geconnecteerd is met [Google Analytics](#) waar de mensen van het bedrijf de data kunnen bekijken en beheren om de applicatie te verbeteren.

4.3 Opzetten Van De Omgeving

Nu dat we weten hoe de applicatie er zal zien, kunnen we beginnen met de omgeving op te bouwen waarin er gewerkt zal worden.

Om deze omgeving op te zetten hebben we een aantal **libraries** nodig, als eerste hebben we **Node.js** nodig. **Node.js** is een omgeving dat **asynchroon** en op **events** gericht is, het dient voornamelijk om schaalbare **netwerk applicaties** te bouwen. De meeste gekende **frameworks** die op deze omgeving draaien zijn **Angular** en **React**. **Node.js** kan gedownload worden via <https://nodejs.org/en/>, daar is er de mogelijkheid om ofwel de laatste nieuwe versie te gebruiken die niet aangeraden wordt voor niet ervaren gebruikers ofwel de **LTS** versie te gebruiken wat eerder aangeraden wordt voor de meeste gebruikers.

Wanneer **Node.js** is geïnstalleerd dan kunnen we beginnen om **Expo** te installeren. Het installeren van **Expo** is eenvoudig het volstaat om een commando te typen in een gewone opdrachtprompt of terminal. Het commando is als volgt:

```
'npm install expo-cli --global'
```

dit zorgt ervoor dat **Expo** globaal wordt geïnstalleerd en dus overal gebruikt kan worden. Om **Expo** te kunnen gebruiken is het wel nodig om een account aan te maken, dit kan gewoon in de terminal wanneer **Expo** voor de eerste keer gebruikt wordt (in dit geval bij het aanmaken van het project).

Wanneer **Node.js** en **Expo** klaar staan om te gebruiken kunnen we beginnen met het opzetten van het project. We nemen opnieuw een terminal en gaan naar de locatie navigeren waar het project moet komen. Dan kunnen we **Expo** gebruiken om het project aan te maken met volgend commando: `'C:\'pad waar het project komt'> expo init recipechef'` daarna zal deze vragen welk sjabloon er gebruikt moet worden. Voor dit project wordt er gekozen voor blank zodat het eenvoudiger is om het project van niets op te bouwen. Eens dit gekozen zullen alle nodige **libraries** geïnstalleerd worden om de basis applicatie werkende te krijgen.

Alvorens de applicatie opgestart wordt moet we eerst de **SDK** versie van **Expo** en **React** aanpassen aangezien er voor dit onderzoek met **SDK** versie 35 gewerkt wordt en de **SDK** versie 36 niet werkte wanneer de applicatie werd opgestart. De aanpassingen die moeten gebeuren hiervoor is in `app.json` daar staat op regel 6 `'sdkVersion'` wat versie 35.0.0 moet worden. Ook in de `package.json` zal er aanpassingen moeten gebeuren, onder de `dependencies` (**libraries**) staat `'expo'` dit moet `'~35.0.0'` worden, er staat ook `'react-native'` daar staat er op het einde van die link `'sdk-36.0.0.tar.gz'` wat uiteraard ook terug 35.0.0 moet worden.

Nu om de applicatie te kunnen starten moet het commando

`C:\'pad naar project'> npm start` uitgevoerd worden in de map van de applicatie (in dit geval de map `'/recipechef'`). Wanneer dit volledig is opgestart zal een website opengaan in de standaardbrowser, dan is er de optie om de applicatie te openen via de website of de terminal. Op de website kan de applicatie opgestart worden, de link sturen via email of de QR code inscannen. Via de terminal kan dit ook maar dan door bepaalde knoppen in te drukken behalve de QR code dan. Door `'a'` te drukken start de applicatie op de emulator of het gekoppelde Android toestel, `'c'` zal een nieuw toestel connecteren, `'e'` zal een email sturen met de link en `'w'` zal de applicatie starten in de website.

Bij dit onderzoek zal er een emulator gebruikt worden van [Android Studio](#) en zal er dus gewoon in de terminal `'a'` gedrukt worden. Dit zorgt ervoor dat de [Expo](#) applicatie geïnstalleerd wordt op het toestel, wanneer het klaar is met de installatie dan zal het vragen toe te stemmen met het tonen van de applicatie over andere applicaties anders zal [Expo](#) niet werken tot dit goedgekeurd is.

Wanneer het is goedgekeurd volstaat het om op de terug knop (de meeste linkse knop) van het toestel te drukken (niet van de applicatie) dan zal deze terug naar de applicatie gaan en deze vervolgens opstarten. Normaal zou deze volledig moeten werken en geen problemen mogen geven en kunnen we beginnen met het toevoegen van de [libraries](#) die nodig zijn voor de demo applicatie werkende te krijgen.

Alvorens de [libraries](#) toegevoegd worden is het best om eerst de applicatie uit te schakelen in de terminal door de knop combinatie `'ctrl + c'` te drukken, en om dan het volgende commando uit te voeren:

```
C:\'pad naar project'> npm install
react-native-design-utility
react-native-elements react-native-gesture-handler
react-native-reanimated react-navigation
react-navigation-stack react-navigation-tabs
```

Wanneer dit is geïnstalleerd start de applicatie opnieuw op door het commando `'npm start'` nu staat de omgeving klaar om de applicatie te maken.

4.4 Maken van de applicatie

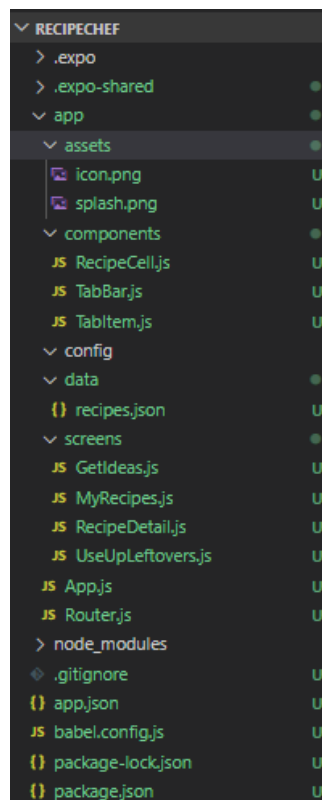
Het volledige project, code voor de demo applicatie en de [analytics tools](#) zijn te vinden op <https://github.com/BillyCottrell/bachelorproef>. Weliswaar wanneer de demo applicatie wordt gebruikt en de [libraries](#) zijn geïnstalleerd dan moet er eerst nog een aanpassing gebeuren aan de [library](#) anders zal deze dan ook niet werken. Zo moet er een aanpassing gebeuren in het bestand `AppEntry.js` dit is te vinden in de map `node_modules/expo/` daar staat op lijn 5 `import App from '.././App'`; dit moet aangepast worden naar `import App from '.././app/App'`;

Om de applicatie te maken beginnen we met het maken van de folder structuur zodat alles goed is ingedeeld en makkelijk terug te vinden is. We zullen daarom dan ook beginnen met de map app aan te maken waaronder alle bestanden komen die deeluitmaken van de applicatie of deze opbouwen. Zo komt de map assets hieronder waarin alle afbeeldingen komen van de applicatie net zoals het bestand App.js. Eerst moeten we app.json aanpassen, daar moeten we bij icon het pad aanpassen naar `./app/assets/icon.png`, dit is hetzelfde voor de splash afbeelding maar deze staat dan wel onder 'splash' en dan 'image'.

Nu gaan we eerst de mappen components, config, data en screens maken om alle bestanden makkelijker in te delen. In de map components komen de onderdelen die de schermen opbouwen in de map config komen bestanden die configuraties bevatten voor onder andere databases of [analytics](#) (in dit geval) of andere externe verbindingen die geconfigureerd moeten worden voor gebruik. In de map data zal er een JSON bestand komen dat de data van de recepten omvat zodat het lokaal ontwikkelen eenvoudiger gaat, verder komen de verschillende schermen in de map screens.

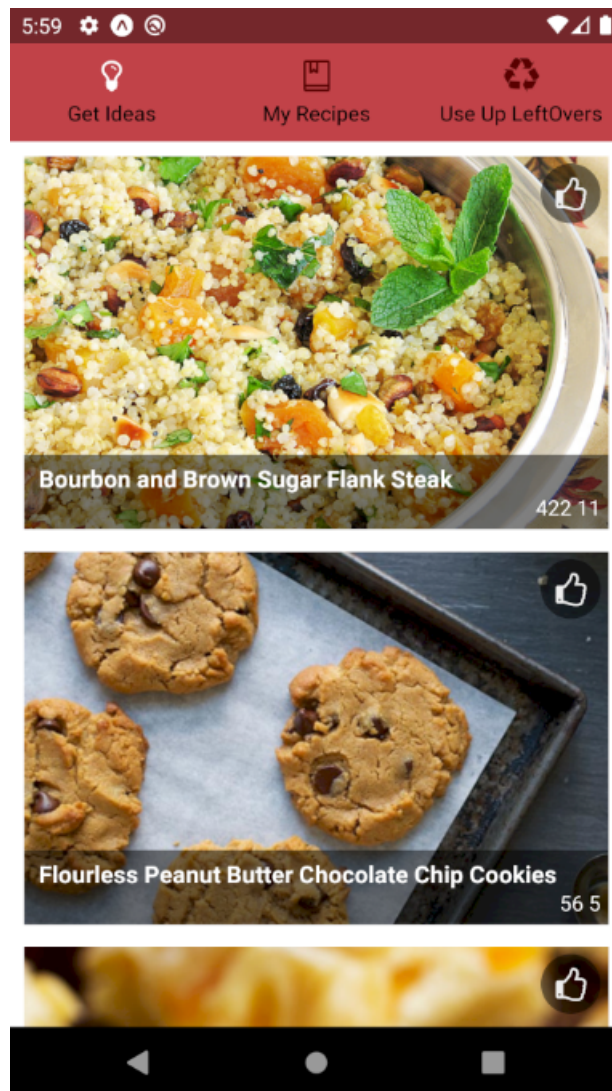
In de map components zullen er 3 [componenten](#) komen de TabBar, TabItem en RecipeCell. In de map config zal later de [analytics](#) configuratie komen. Bij de map screens zullen er dan 4 schermen komen het hoofdscherm GetIdeas, dan MyRecipes, UseUpLeftovers en RecipeDetail.

Wanneer bovenstaande stappen zijn uitgevoerd dan bekomt men volgende folder structuur:



Figuur 4.4: Folder structuur demo applicatie

Wanneer de applicatie opstart zal volgend scherm getoond worden:



Figuur 4.5: Startscherm Get Ideas

Wanneer er naar het tabblad My Recipes wordt genavigeerd dan bekomt men volgend scherm:



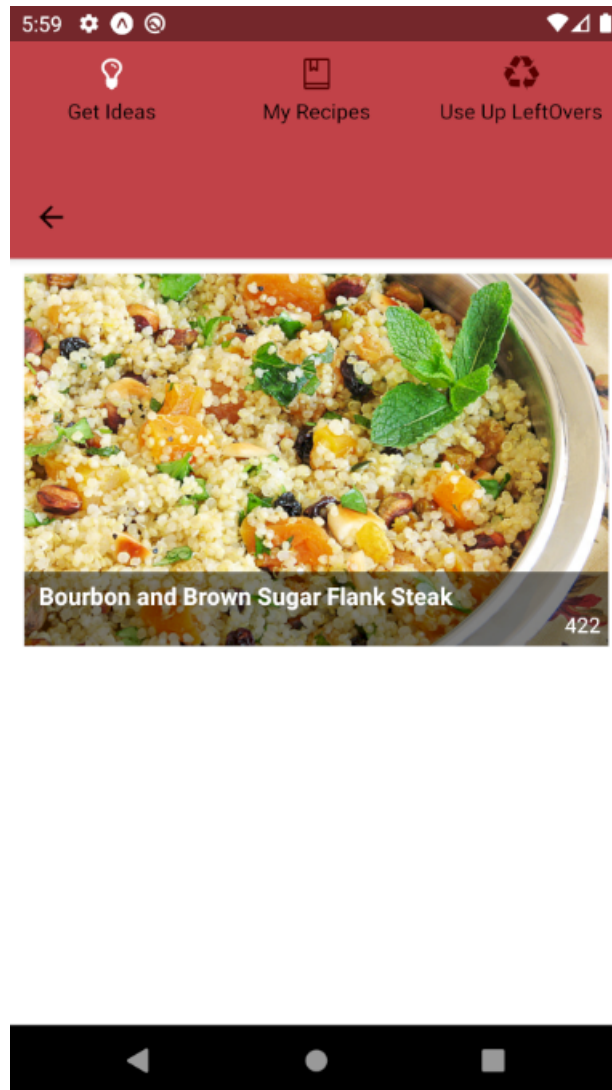
Figuur 4.6: My Recipes scherm

Leftovers is hier gelijkaardig aan doordat de functionaliteit in Get Ideas allemaal zit:



Figuur 4.7: Leftovers scherm

Wanneer er in Get Ideas een recept wordt aangeklikt dan ziet men de details van dat recept:



Figuur 4.8: Recipe Detail scherm

5. Expo Analytics

Gedurende deze vergelijkende studie wordt er eerst dieper ingegaan op [Expo Analytics](#). Zo zullen de mogelijkheden die deze [tool](#) aanbied afgetoetst worden door op 5 verschillende gebieden te gaan onderzoeken zoals vermeld in Hoofdstuk 3.

5.1 Data en Functies

[Google Analytics](#) en [Expo Analytics](#) bieden een verscheidenheid aan data en functies aan. Deze functionaliteiten zijn namelijk dezelfde als [Google Analytics](#) alleen kunnen er geen integraties gebeuren hiermee.

Er is de mogelijkheid om [page hits](#), [screen hit](#), [events](#), custom [dimensions](#), custom [metrics](#) en [transacties](#) te versturen. Daarnaast is er een lange lijst aan parameters die mee verstuurd kunnen worden. Het heeft alle mogelijkheden van [Google Analytics](#) behalve integraties met andere [tools](#).

De data die [Google Analytics](#) heeft naast de data die worden verstuurd met de verschillende functies, meer informatie zoals de scherm resolutie, naam, ID en versie van de applicatie, maar ook informatie over de gebruiker of deze uniek is of niet, de sessies, enz.

Door middel van parameters kan informatie over goals en [campagnes](#) ook meegestuurd worden zodat deze volledig gebruikt kunnen worden.

Laten we deze functionaliteiten iets dieper bekijken met een aantal voorbeelden en resultaten:

Volgende voorbeelden kunnen enkel gebruikt worden door het implementeren van de ExpoAnalyticsManager klasse deze is te vinden in het project op github onder app/config.

Het verzenden van een [page hit](#) binnen [Expo Analytics](#) gebeurt op deze manier:

```
ExpoAnalyticsManager.getAnalytics().pageHit('Get Ideas');
```

Hierbij zal er naar [Google Analytics](#) worden gestuurd dat een gebruiker naar de pagina Get Ideas is gebrowsed. Wanneer deze [page hit](#) juist heeft plaatsgevonden dan kunnen we dit bekijken binnen [Google Analytics](#) bij de Real-Time rapporten. We kunnen dit bij het Overview rapport zien, maar ook bij Content. Wanneer er wat tijd voorbij is gegaan (tot wanneer de tijd is overgegaan naar een ander uur) dan kunnen we deze ook bekijken bij Behaviour onder Overview.

Wanneer er meerdere [page hits](#) actief zijn kunnen we hier een flow van zien bij Behaviour onder Behaviour Flow, daar kunnen we onder andere zien welke pagina's er worden bezocht en welke eerst en vooral bekeken worden. Daarnaast kunnen we ook zien waar de meeste gebruikers de applicatie verlaten.

Bij Behaviour onder Site Content zal er meer informatie staan over het aantal keer de pagina's zijn bekeken voor hoelang het aantal sessies hoeveel elke pagina heeft opgeleverd bij [transactions](#), maar ook waar gebruikers de applicatie opstarten of verlaten.

Bij Site Speed kan er gezien worden hoelang het geduurd heeft alvorens de pagina was geladen deze meting moet wel manueel worden doorgegeven om dit op te kunnen volgen. Dit is alles wat opgevraagd kan worden in verband met [page hits](#).

Naast [page hits](#) kunnen er ook [screen hits](#) worden gestuurd dit gebeurt op volgende wijze:

```
ExpoAnalyticsManager.getAnalytics().screenHit('Get Ideas');
```

Het verschil met een [page hit](#) is dat [page hits](#) enkel gemeten worden wanneer er in [Google Analytics](#) bij de [property](#) instellingen is gekozen om een website op te volgen. Wat wil zeggen dat in dit geval [screen hits](#) wel doorgestuurd worden maar niet in rapporten worden getoond doordat er enkel voor websites gemeten wordt. Dit is dus een functie dat geen nut heeft en dus niet gebruikt moet worden.

Daarnaast kunnen we ook nog [events](#) doorsturen dit gebeurde op volgende manier:

```
ExpoAnalyticsManager.getAnalytics({dp: '/Get Ideas', dt: 'Get Ideas'})  
.event('Like', 'Like-Bourbon-and-Brown-Sugar-Flank-Steak',  
'Bourbon and Brown Sugar Flank Steak', 1);
```

Hierbij is het nodig om extra parameters mee te geven over de pagina waarop het **event** heeft plaatsgevonden. Wanneer een **event** juist is doorgestuurd dan kunnen we deze ook bij de Real-Time rapporten bekijken onder Events. Daarnaast kunnen we deze informatie ook bekijken in Behaviour onder Events, bij Overview en Top Events kunnen we algemeen informatie zien van de **events** die hebben plaatsgevonden, Bij Pages kunnen we dan zien op welke pagina's deze **events** hebben plaatsgevonden. Bovendien kunnen we de volledige flow bekijken bij Events Flow hier zien we welke **events** er gebeurd zijn, welke daarop volgen en wanneer de gebruiker de applicatie verlaat.

Daarnaast kunnen we ook zelfgemaakte **dimensions** en **metrics** doorsturen naar **Google Analytics** dit gebeurt als volgt: Toevoegen zelfgemaakte **dimension**:

```
ExpoAnalyticsManager.getAnalytics()  
.addCustomDimension(1, 'Main Dish');
```

Toevoegen zelfgemaakte **metric** (vb als het een Main Dish gerecht is doe dan +1):

```
ExpoAnalyticsManager.getAnalytics().addCustomMetric(1,1);
```

Wanneer deze zijn aangemaakt dan worden deze achteraf toegevoegd aan de **page hit** die wordt verstuurd. We kunnen deze waarden overal bekijken waar het toegevoegd kan worden bij de bestaande rapporten. Daarbovenop kan dit ook bekeken worden in zelfgemaakte rapporten dit staat boven de gewone rapporten bij Customization, daar kunnen er nieuwe rapporten aangemaakt worden met op voorhand gedefinieerde en zelfgemaakte **dimensions** en **metrics**.

Daarnaast kunnen we ook **transacties** sturen als volgt:

```
ExpoAnalyticsManager.getAnalytics()  
.sendTransaction('Recipe Chef', 35.42, 0, 2);  
ExpoAnalyticsManager.getAnalytics()  
.addItem('Bourbon and Brown Sugar Flank Steak',  
35.42, 1, 'SKU21', 'Main Dish');
```

Deze informatie kunnen we voornamelijk bekijken onder Conversions bij **e-commerce** daar kunnen we het aantal **transacties** bekijken hoeveel deze hebben opgeleverd en wat het gemiddeld heeft opgebracht. Daarnaast kunnen we onder Product Performance bekijken hoe goed een bepaald product heeft gedaan, en onder Sales Performance kunnen we dan zien hoeveel per maand er is verdiend. Bij het rapport **Transacties** kunnen we zien hoeveel elke **transactie** heeft opgeleverd en bij Time to Purchase zien we hoe lang het geduurd heeft alvorens er een aankoop is gebeurd.

Naast al deze informatie kunnen we ook informatie bekijken over de gebruiker en het toestel, bij Audience kunnen we ook nog informatie bekijken over specifieke gebruikers zoals hun geschiedenis van acties, gaande van [page hits](#), [events](#) en [transacties](#). Natuurlijk zijn de gegevens van deze gebruikers anoniem en weten we niet wie dit is aangezien er geen naam, email, telefoonnummer of dergelijke is aangekoppeld. Informatie dat de gebruiker zou kunnen identificeren wordt dan ook niet aanvaard door Google omwille van de [GDPR](#) regels.

Verder kunnen er ook nog doelstellingen en [campagnes](#) opgezet worden en door middel van parameters deze gaan gebruiken waar dit nodig is. Deze rapporten zijn dan ook te vinden bij Acquisition en Conversions.

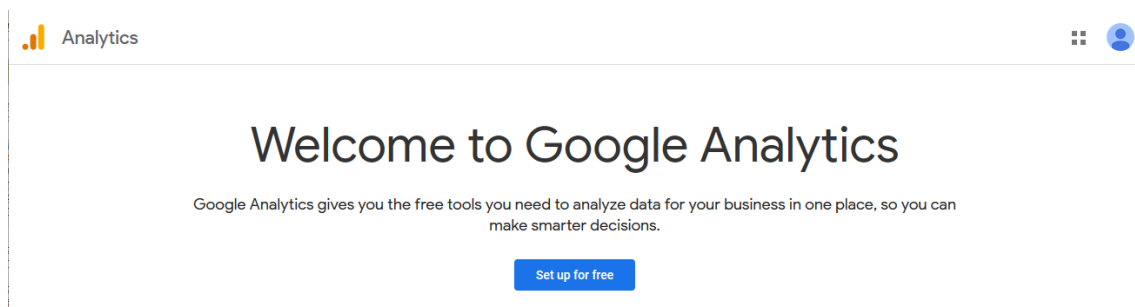
De data van bij [Google Analytics](#) kan niet geëxporteerd worden tenzij er gebruikgemaakt kan worden van externe [tools](#) die kan inpikken op deze data en gegevens.

5.2 Implementatie en Architectuur

Nu dat we de mogelijkheden van de [tool](#) hebben besproken kunnen we beginnen met deze functionaliteiten te implementeren.

5.2.1 Implementatie

Om [Expo Analytics](#) te gebruiken moet er eerst een account aangemaakt worden in [Google Analytics](#). Hierbij wordt er wel verondersteld dat er al een account is aangemaakt bij Google zelf en de gebruiker al is ingelogd. Het aanmaken van een account bij [Google Analytics](#) kan gebeuren door naar de website <https://analytics.google.com> te browsen daar kan je een account opzetten, wanneer er op de blauwe knop is gedrukt.



Figuur 5.1: Aanmaken Google Analytics Account

Nadien zal er doorverwezen worden naar [Google Analytics](#) om een account aan te maken en deze te gaan instellen.

Create account

Account setup

Account details

Account name (Required)
Accounts can contain more than one tracking ID.
RecipeChef

Account Data Sharing Settings

The data sharing options give you more control over sharing your Google Analytics data. [Learn more.](#)

- ☒ **Google products & services** RECOMMENDED
If you have enabled Google signals, this setting will also apply to authenticated visitation data which is associated with Google user accounts. This setting is required for **Enhanced Demographics & Interests reporting**. If you disable this option, data can still flow to other Google products explicitly linked to your property. Visit the product linking section in each property to view or change your settings. [Show Example](#)
- ☒ **Benchmarking** RECOMMENDED
Contribute anonymous data to an aggregate data set to enable features like benchmarking and publications that can help you understand data trends. All identifiable information about your website is removed and combined with other anonymous data before it is shared with others. [Show Example](#)
- ☒ **Technical support** RECOMMENDED
Let Google technical support representatives access your Google Analytics data and account when necessary to provide service and find solutions to technical issues.
- ☒ **Account specialists** RECOMMENDED
Give Google marketing specialists and your Google sales specialists access to your Google Analytics data and account so they can find ways to improve your configuration and analysis, and share optimization tips with you. If you don't have dedicated sales specialists, give this access to authorized Google representatives.

[Learn how Google Analytics safeguards your data.](#)

[Next](#) [Previous](#)

What do you want to measure? [Edit](#)

Property setup [Edit](#)

Figuur 5.2: Opzetten Google Analytics Account

Hiervoor kan je een naam opgeven dit kan de naam van het bedrijf of de organisatie zijn. Daarna wordt er gevraagd wat er gemeten moet worden enkel website, apps of beide, hierbij zal er gekozen worden voor website doordat **Expo Analytics** een **tracking ID** nodig heeft wat enkel beschikbaar is bij websites.

Create account

What do you want to measure?

Web
Measure your website

- Understand where your users are coming from and turn data into insights
- Analyze user behavior and optimize for your business
- Discover trends with performance and conversion analysis

Apps
Measure your iOS or Android apps

- Understand user growth and get insights into app behavior
- Automatically capture key events or define your own
- Add web stream later to measure cross-platform behavior BETA

Apps and web BETA
Measure your users across app and web

- Explore cross-platform user-centric analytics
- Get started quickly with codeless event configuration and out-of-the-box reports
- Implement without retagging if you already use gtag.js or Tag Manager on your site

[Next](#) [Previous](#)

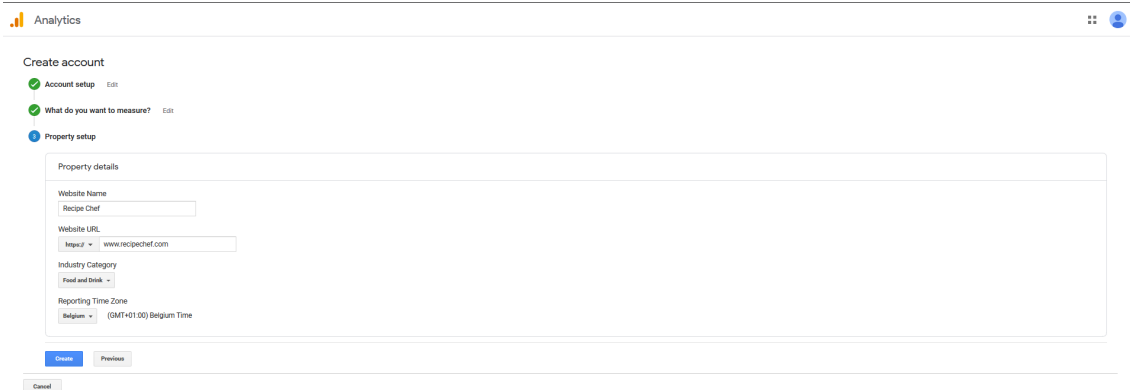
Property setup [Cancel](#)

© 2019 Google | [Analytics home](#) | [Terms of Service](#) | [Privacy Policy](#) | [Send feedback](#)

Figuur 5.3: Keuze van platform voor specifieke tracking

Wanneer men dit gedaan heeft wordt er nog gevraagd om een **property** aan te maken dit is om data makkelijker in te delen. Bij 1 **property** kunnen er verschillende websites aangekoppeld worden zodat alle data bij elkaar komt, dit kan achteraf terug opgedeeld worden met **views** en filters om dit te organiseren, er kunnen ook **segments** gebruikt worden om data op rapport niveau verder op te delen in subgroepen. Natuurlijk hangt dit af van het plan dat er gekozen en kan er evengoed meerdere **properties** gekozen zijn dat elk met 1 website verbonden is.

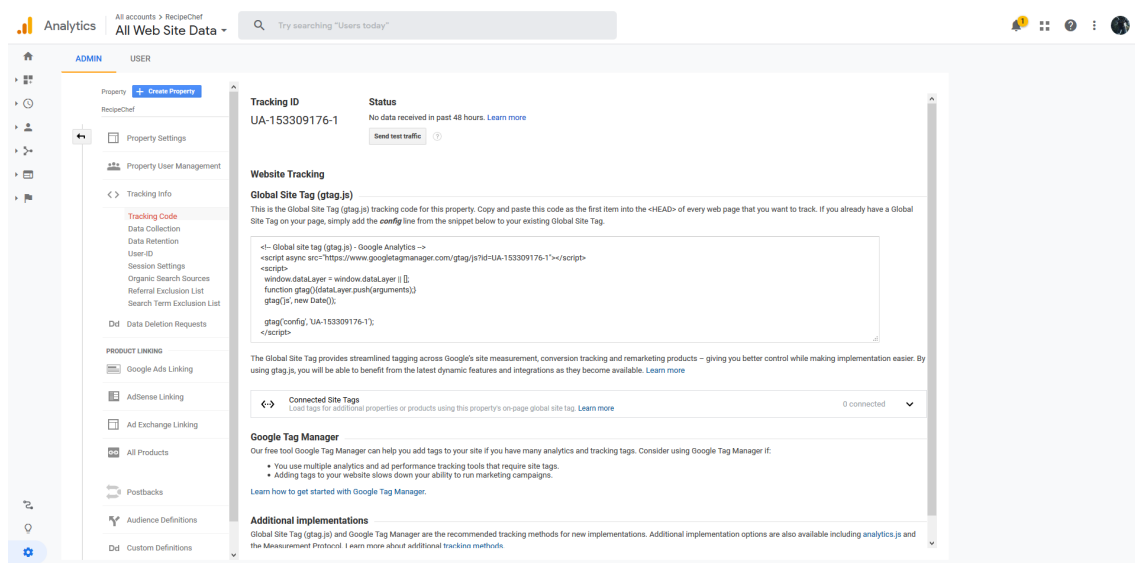
Om een **property** aan te maken wordt de naam van de website opgegeven, de url van de website, de industriële categorie en de tijdzone waarin gerapporteerd wordt.



The screenshot shows the 'Property setup' form in the Expo Analytics interface. It includes fields for 'Website Name' (Recipe Chef), 'Website URL' (https://www.recipechef.com), 'Industry Category' (Food and Drink), and 'Reporting Time Zone' (Belgium). There are 'Create', 'Previous', and 'Cancel' buttons at the bottom.

Figuur 5.4: Opzetten van een property

Nadat alle informatie is ingegeven zijn zowel het account als de **property** aangemaakt en wordt de **tracking ID** getoond. Nu we de **tracking ID** hebben kunnen we beginnen met de implementatie binnen de applicatie.



The screenshot shows the 'Tracking ID' page in the Expo Analytics interface. It displays the 'Tracking ID' (UA-153309176-1) and the 'Status' (No data received in past 48 hours). Below this, there is a 'Website Tracking' section with a 'Global Site Tag (gtag.js)' code snippet. The code is as follows:

```
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-153309176-1"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments)};
  gtag('js', new Date());

  gtag('config', 'UA-153309176-1');
</script>
```

Below the code, there is a 'Connected Site Tags' section showing '0 connected' tags. At the bottom, there is a 'Google Tag Manager' section with a brief description and a link to learn more.

Figuur 5.5: Verkrijgen van de tracking id

Om **Expo Analytics** binnen de demo applicatie te gebruiken is het nodig om deze te installeren door middel van `npm install expo-analytics --save`

Expo Analytics kan gebruikt worden op 2 verschillende mogelijkheden, het **analytics** object aanmaken op elke locatie waar we **Expo Analytics** willen implementeren of het **analytics** object in een aparte klasse voorzien. Bij de tweede mogelijkheid kunnen we dit doen als een **service**, een klasse dat geïnjecteerd en enkel aangemaakt wordt wanneer nodig, ofwel maken we hiervan een singleton. Dit is een klasse die maar 1 keer wordt aangemaakt tijdens het gebruik van de applicatie.

Hierbij is er gekozen om dit in een aparte klasse te plaatsen en deze als een singleton te gebruiken.

Bij het gebruik hiervan was er tevens gekozen om het `analytics` object ook maar 1 maal aan te maken. Dit was eenvoudig en geen probleem tot wanneer er opgemerkt werd dat bij het versturen van een `event` er niet geweten was van op welke pagina het werd verzonden.

Na verder onderzoek is er beslist om de structuur te behouden en het `analytics` object telkens opnieuw aan te maken met de parameters die werden doorgegeven, waaronder de pagina waarop het `event` plaatsvond. Dit resulteerde in de volledige correcte werking van `Expo Analytics`. Het resultaat hiervan is te vinden op <https://github.com/BillyCottrell/bachelorproef/blob/master/Expo%20Analytics%20project/app/config/ExpoAnalyticsManager.js>.

In deze code ziet u dat in de `getAnalytics()` methode het `analytics` object wordt aangemaakt aan de hand van de `tracking ID` die terug te vinden is in `Google Analytics`. Hierdoor kan deze klasse overal aangesproken worden om `analytics` te kunnen gebruiken volstaat het om bovenaan in een bestand

```
'import ExpoAnalyticsManager from '../config/ExpoAnalyticsManager';' te plaatsen.
```

Om ervoor te zorgen dat er telkens een `page hit` wordt gestuurd wanneer 1 van de Tabs wordt geselecteerd, wordt er voor dat de navigatie gebeurd een `page hit` verstuurd. Deze code bevindt zich in `TabItem.js`:

```
/**
 * Deze methode behandelt het klikken van een tabItem en zorgt
 * ervoor dat er een pageHit verstuurd wordt naar Google Analytics
 */
handlePress = () => {
  ExpoAnalyticsManager.getAnalytics()
    .pageHit(this.props.routeName);
  this.props.navigation.navigate(this.props.routeName);
}
```

Wanneer de `GetIdeas` pagina wordt geladen bij het opstarten van de applicatie dan wordt bovenstaande code niet uitgevoerd om ervoor te zorgen dat dit wel het geval is moet er een `page hit` verstuurd worden wanneer de pagina wordt ingeladen. Deze code bevindt zich in `GetIdeas.js`:

```
/**
 * Deze methode zal aangeroepen worden nadat het scherm geladen is
 */
componentDidMount(){
  ExpoAnalyticsManager.getAnalytics().pageHit('Get Ideas');
}
```

Wanneer er op de GetIdeas pagina op de like knop wordt gedrukt dan moet er een [event](#) worden doorgestuurd. Deze code bevindt zich in `GetIdeas.js` maar wordt aangesproken vanuit `RecipeCell.js`:

```
/**
 * Deze methode updates het recept dat geliked is en updates de
 * volledige lijst.
 * @param {number} recipeIndex - is de positie van het recept in
 * de lijst.
 */
updateRecipes = (recipeIndex) => {
  const {recipes} = this.state;
  recipes[recipeIndex].likes++;
  ExpoAnalyticsManager.getAnalytics().event('Like', 'Like-' +
    recipes[recipeIndex].name.replace(/\s+/g, '-'),
    recipes[recipeIndex].name, 1);
  this.setState({recipes});
}
```

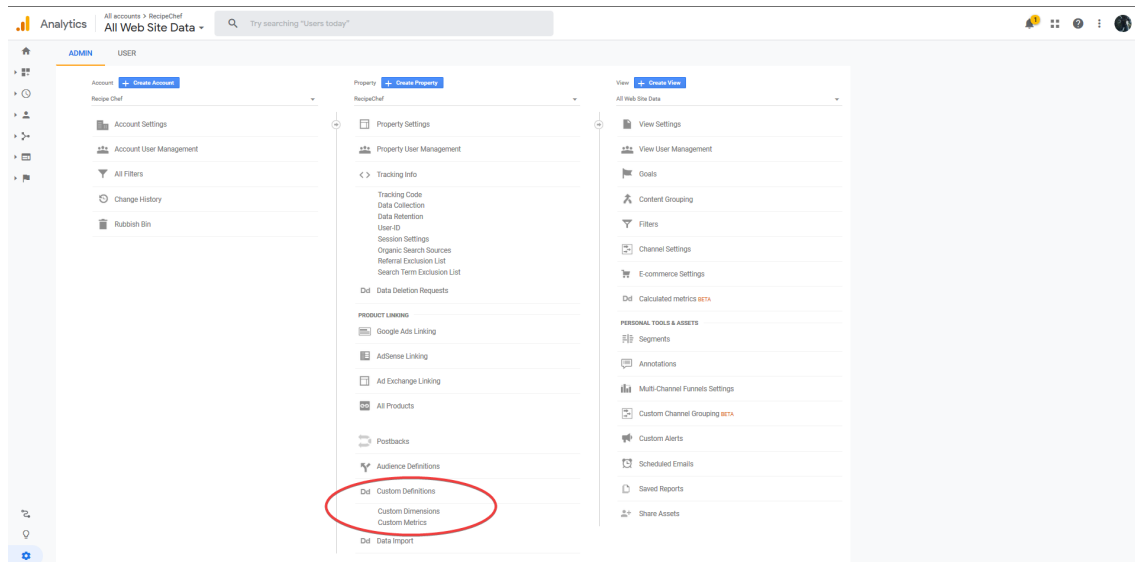
Nu deze code zal werken maar zal geen informatie over de pagina hebben om dit op te lossen volstaat het om bij `getAnalytics()` een aantal parameters toe te voegen `getAnalytics({dp: '/Get Ideas', dt: 'Get Ideas'})`. Bij `dp` wordt het pad van de pagina meegegeven en bij `dt` de pagina titel.

Wanneer een recept wordt aangeklikt dan zal volgende functie aangeroepen worden in `RecipeDetail.js`:

```
/**
 * Deze methode wordt aangeroepen voordat het scherm zal laden
 */
componentWillMount(){
  const {navigation} = this.props;
  const recipe = navigation.getParam('recipe');
  ExpoAnalyticsManager.getAnalytics()
    .addCustomDimension(1, recipe.category);
  if(recipe.category==='Main Dish'){
    ExpoAnalyticsManager.getAnalytics().addCustomMetric(1,1);
  }
  ExpoAnalyticsManager.getAnalytics()
    .pageHit('Recipe ' + recipe.name);
  ExpoAnalyticsManager.getAnalytics().removeCustomDimension(1);
  ExpoAnalyticsManager.getAnalytics().removeCustomMetric(1);
}
```

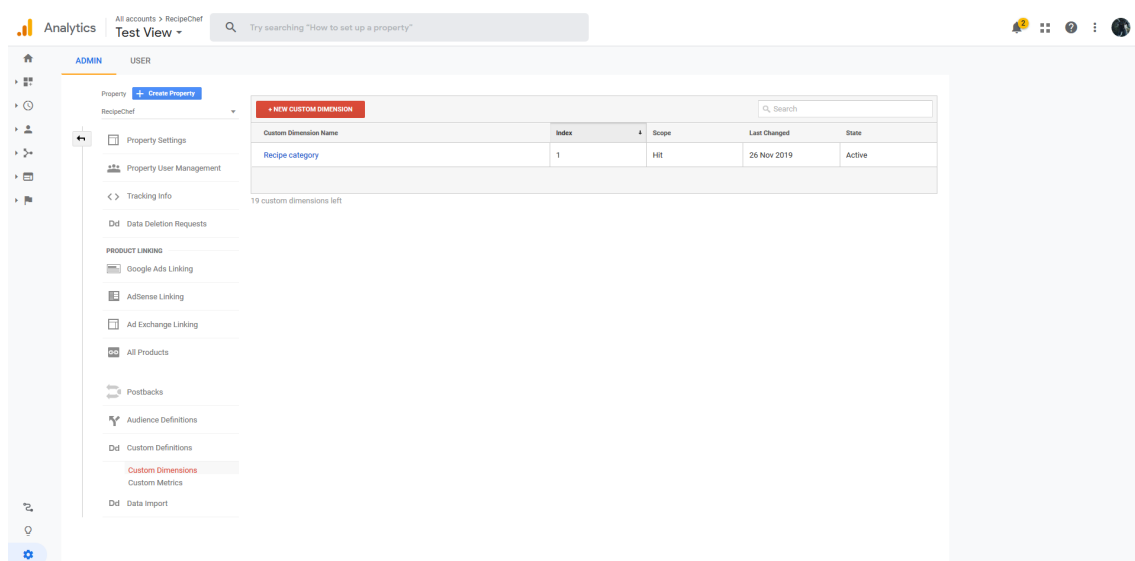
Naast het versturen van de normale **page hit** zal er ook een zelfgemaakte **dimension** en **metric** meegestuurd worden. De **dimension** zal telkens de categorie van het recept meesturen zodat we later in rapporten kunnen bekijken welke categorieën het meest worden bekeken. De **metric** zal telkens meten of het recept een Main Dish recept is of niet.

De eerste waarde die meegestuurd wordt bij zowel de **metric** als de **dimension** is de id die terug te vinden is in **Google Analytics** onder Admin -> [select property] -> Custom Definitions -> Custom **Metrics** zoals te zien in onderstaande afbeelding.



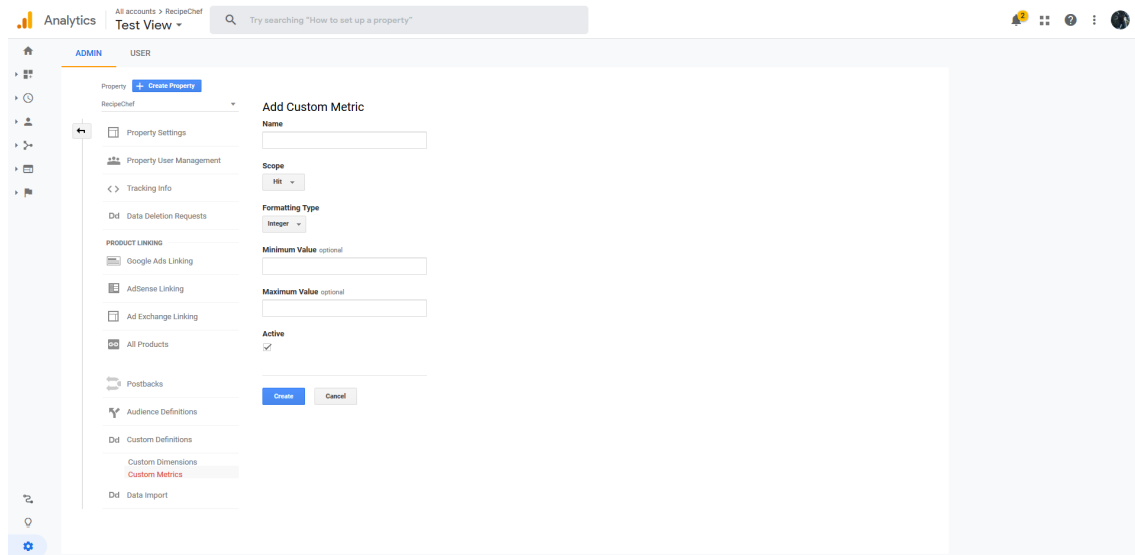
Figuur 5.6: Custom Dimensions en Metrics

Om een **dimension** of **metric** aan te maken volstaat het om 1 van beide te selecteren daar kan je al de **dimensions** en **metrics** bekijken en door op de rode knop te klikken kan je er 1 aanmaken.



Figuur 5.7: Pagina voor het aanmaken van Dimensies en Metrics

Bij het aanmaken van een **dimension** volstaat het om een naam in te geven en aan te duiden op welk niveau deze zal gelden **Hit**, Sessie, Gebruiker of Product, maar er kan ook aangeduid worden of deze actief moet zijn of niet. Bij het aanmaken van een **metric** kan er daar bovenop de formattering meegegeven worden waar de keuze bestaat uit een getal, valuta of tijd en kan de minimum en maximum waarde daarvoor opgegeven worden.



Figuur 5.8: Metric aanmaken

De **metric** en **dimension** worden nadien voor het zekerste verwijderd zodat deze toch niet opnieuw zouden meegestuurd worden met andere **page hits**.

Nu wanneer de gebruiker verder klikt op de afbeelding dan koopt deze zeggend het recept aan en wordt een **transactie** verzonden waarbij het recept aangekoppeld wordt.

```
/**
 * Deze methode wordt aangeroepen als men het recept wilt ‘aankopen’
 * en updates de transactie ID zodat deze klaar is voor de volgende
 * ‘aankoop’
 */
_buyRecipe = () => {
  const {navigation} = this.props;
  const recipe = navigation.getParam('recipe');
  ExpoAnalyticsManager.getAnalytics()
    .sendTransaction('Recipe Chef', 35.42, 0, 2);
  ExpoAnalyticsManager.getAnalytics()
    .addItem(recipe.name, 35.42, 1, 'SKU21', recipe.category);
  this.state.transid++;
  this.props.navigation.navigate('My Recipes',
    { id: this.props.id });
}
```

Om de [transactie](#) volledig te kunnen sturen moet deze als eerste worden aangemaakt, hierbij wordt de naam van de applicatie meegestuurd, de totale prijs, transportatie kosten, en belastingen. Aan de [transactie](#) wordt een recept toegevoegd met de naam van het aangekocht recept, de prijs van dat recept, hoeveel recepten (in dit geval 1), de [SKU](#) of Stock Keeping Unit (soort van id voor het magazijn) van het recept en de categorie van het recept.

Om ervoor te zorgen dat wanneer men terugkeert naar de lijst van recepten er terug een [page hit](#) wordt verstuurd dan moet er in `RecipeDetail.js` volgende code komen:

```
/**
 * Deze methode wordt aangeroepen voor het scherm alvorens
 * er naar een ander scherm wordt genavigeerd
 */
componentWillUnmount(){
  ExpoAnalyticsManager.getAnalytics().pageHit('Get Ideas');
}
```

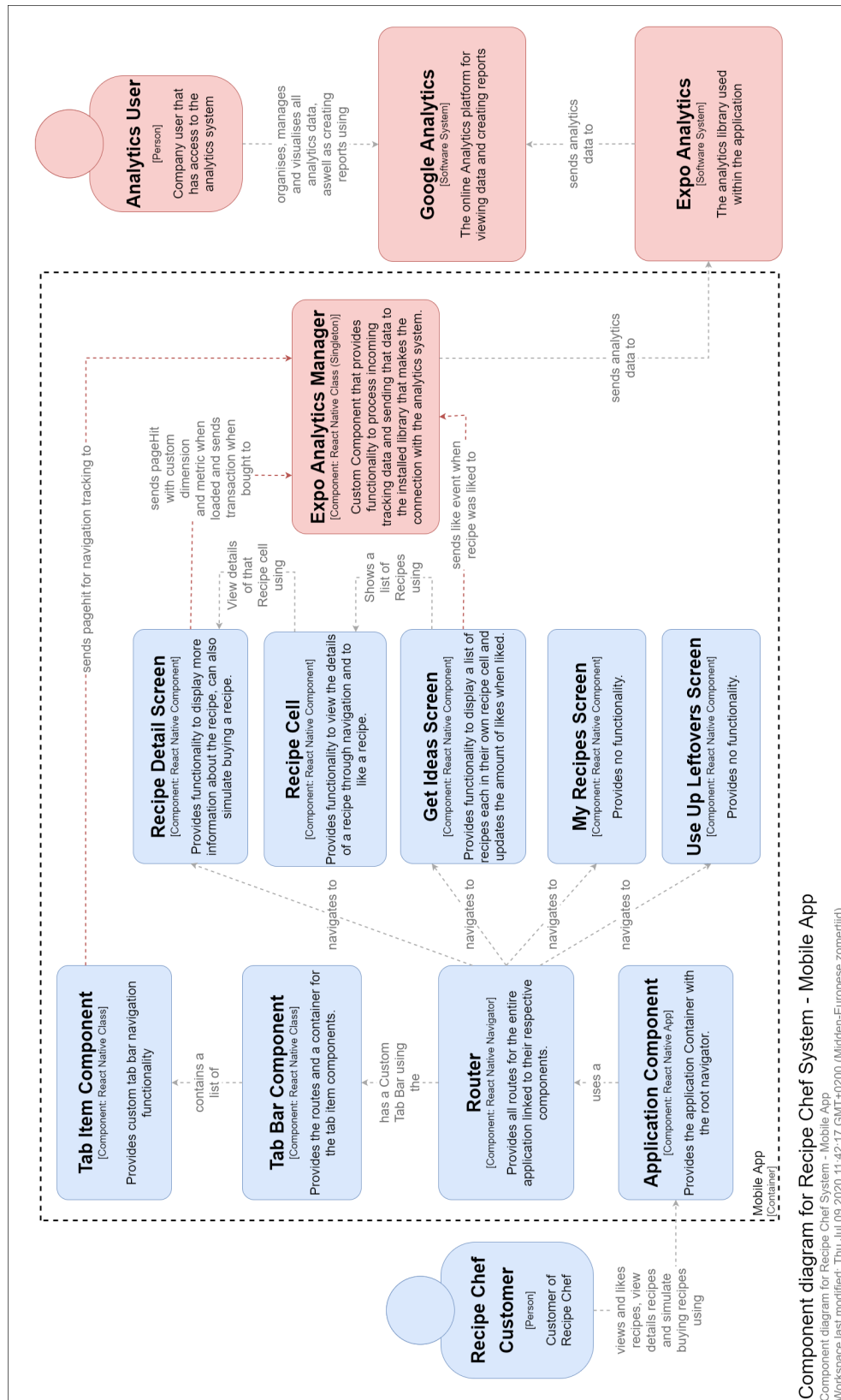
Er zou gezegd kunnen worden dat dit niet nodig is doordat deze code al in `GetIdeas.js` staat wanneer het [component](#) geladen wordt. De reden hiervoor is dat het `GetIdeas` [component](#) niet uitgeladen wordt en dus wordt deze methode ook niet uitgevoerd.

Hierbij is de implementatie klaar en kunnen we bekijken hoe de architectuur van onze applicatie eruit ziet

Let wel op dat deze demo applicatie niet conform is aan de [GDPR](#) regels, om hieraan te voldoen moet de gebruiker een melding krijgen wanneer hij deze applicatie voor het eerst gebruikt. Deze melding houdt in dat het de gebruiker laat weten dat er [analytics](#) gebruikt wordt voor deze applicatie en deze gebruiker mag dan ook [analytics](#) uitschakelen indien men dit niet wilt. Er mag ook de mogelijkheid zijn om [analytics](#) deels te activeren vb. dat er geen [e-commerce](#) tracking zou mogen plaatsvinden, maar dat alle andere [analytics](#) wel mag plaats vinden. De mogelijkheid om [analytics](#) uit te schakelen moet aanwezig zijn aangezien [analytics](#) nooit noodzakelijk is bij het gebruik van een applicatie.

5.2.2 Architectuur

Aangezien we de algemene architectuur hebben bekeken van de demo applicatie en op welke manier de [analytics tools](#) zullen gebruikt worden hoeven we enkel het laatste architectuur te bespreken namelijk het Component diagram. Dit is omdat de andere diagram bijna hetzelfde is en de componenten hieronder ook voorkomen.



Figuur 5.9: Component diagram van Recipe Chef met Expo Analytics

Hier kunnen we zien dat de schermen eerst de data versturen naar de zelfgemaakte [Expo Analytics Manager](#). Deze zal op zijn beurt de informatie verwerken en doorgeven aan de [Expo Analytics library](#) zodat alle informatie verzonden kan worden naar [Google Analytics](#).

5.3 Documentatie

Over [Google Analytics](#) is er een variëteit aan informatie te vinden:

- <https://developers.google.com/analytics/> dit is de documentatie van Google.
- <https://analytics.google.com/analytics/academy/> waar tutorials te vinden zijn en certificaten behaald kunnen worden. Dit is aangeraden voor mensen die geen kennis hebben van [Google Analytics](#) en die hiermee aan de slag willen gaan.
- <https://support.google.com/analytics> dit is een forum van Google wordt ook vaak naar gerefereerd vanuit Google Academy.
- <https://support.google.com/analytics/community> dit is de community van Google waarbij men vragen kan stellen en antwoorden vinden.
- <https://analytics.google.com/analytics/gallery> hier kan men dashboards, zelfgemaakte rapporten en [segments](#) vinden.
- <https://www.youtube.com/user/googleanalytics> youtube kanaal met videos over [Google Analytics](#).

Bij [Expo Analytics](#) is de informatie beperkt doordat er maar 1 bron is en dat is op <https://github.com/ryanvanderpol/expo-analytics>, naast deze bron is er geen informatie over te vinden. Indien er vragen of problemen optreden dan zal men zelf naar de oplossingen moeten zoeken of een issue hierover moeten openen op deze github pagina. Het is wel zo dat de community niet zo actief is en dus enkele dagen kan duren alvorens er antwoord zal komen.

Hoe je een issue aanmaakt vindt je op

<https://help.github.com/en/github/managing-your-work-on-github/creating-an-issue>, om een voorbeeld te hebben over hoe een issue opgebouwd moet worden kan je dit vinden op <https://github.com/stevemao/github-issue-templates>, hoe een issue opgebouwd wordt hangt namelijk af van de soort issue dat het is.

De documentatie van [Expo Analytics](#) is niet zo uitgebreid, maar er zijn wel voorbeelden over het gebruik van de verschillende functionaliteiten en parameters. Het is wel zo dat er maar enkele parameters aanbod komen en voor meer informatie er verwezen wordt naar de documentatie bij Google zelf op <https://developers.google.com/analytics/devguides/collection/protocol/v1/parameters>.

5.4 Prijs / Kwaliteit

Het gebruik van [Google Analytics](#) en [Expo Analytics](#) is volledig gratis. Dit wil zeggen dat er enkel een [operationele kost](#) zal zijn, deze kost zal niet zo hoog liggen enkel in het begin bij de implementatie en het opzetten van [Google Analytics](#). Nadien zal deze kost redelijk zijn aangezien men enkel de rapporten hoeft te bekijken en eventueel als er iets extra bijkomt van implementatie of verwachtingen.

Om de [operationele kost](#) te bepalen zullen we dit gaan opsplitsen zodat alles overzichtelijker is.

In het eerste deel zal er voornamelijk het opzetten van [Google Analytics](#) bekeken worden te samen met de implementatie van [Expo Analytics](#). In het tweede deel zal er voornamelijk het onderhoud van [analytics](#) worden besproken.

Voor het opzetten van [Google Analytics](#) en de implementatie van [Expo Analytics](#) zal er een gemiddelde prijs gehanteerd worden van € 400 per dag en dit voor 5 werkdagen. Dit wil dus zeggen dat het opzetten en implementeren om en bij de € 2000 kost.

Nu hier zijn dan ook onderhoudskosten aan wat vooral het bekijken van rapporten is en het aanpassen van zelfgemaakt [dimensions](#), [metrics](#), [events](#) en rapporten en het verbeteren hiervan. Hiervoor wordt er ongeveer 1 dag per kwartaal aangerekend wat dus € 400 per kwartaal is of € 1600 per jaar.

Dit wil zeggen dat bij het eerste jaar de totale kost ongeveer € 3600 is, en nadien jaarlijks € 1600 hierbij komt.

Dit voor alle functionaliteiten van [Google Analytics](#), rekening houdend met het nadeel dat integraties niet mogelijk zijn. Bij een eigen implementatie is het mogelijk om extra parameters te gebruiken voor externe tools.

5.5 Case studies / Getuigenissen

Over [Expo Analytics](#) zijn er geen getuigenissen of case studies te vinden. Een mogelijke reden waarom er geen getuigenissen en case studies zijn is omdat het niet zo bekend is, de community niet zo groot is en er ook niet veel documentatie/informatie hierover te vinden is. Er zijn wel verschillende getuigenissen en case studies over [Google Analytics](#) en deze worden over het algemeen ervaren als een zeer goede [analytics](#) tool en dit door zowel kleine als grote bedrijven.

6. Google Analytics

De eerste [analytics tool](#) waarbij er in deze vergelijkende studie dieper op ingegaan wordt is [Google Analytics](#). Zo zal ook deze [tool](#) afgetoets worden op de verschillende mogelijkheden dat deze aanbiedt door op de 5 verschillende gebieden te onderzoeken zoals besproken in Hoofdstuk 3.

6.1 Data en Functies

[Google Analytics](#) biedt een verscheidenheid aan data en functies aan.

Doordat we de [Firebase SDK](#) niet hebben gebruikt (zie 6.2) kunnen we ook niet bespreken hoe de functies ervan gebruikt kunnen worden en welke data we hieruit zullen verkrijgen. Daarom zullen we dit bespreken aan de hand van een eigen implementatie deze is weliswaar hetzelfde alleen zal de code om de data te verkrijgen er anders uitzien en zijn er integraties mogelijk.

Zoals gezegd heeft [Google Analytics](#) dezelfde functionaliteiten als [Expo Analytics](#) zo is er de mogelijkheid om [page hits](#), [screen hits](#), [events](#), custom [dimensions](#), custom [metrics](#) en [transacties](#) te versturen. Daarnaast is er een lange lijst aan parameters die mee verstuurd kunnen worden.

De data die [Google Analytics](#) heeft naast de data die worden verstuurd met de verschillende functies, meer informatie zoals de scherm resolutie, naam, ID en versie van de applicatie, maar ook informatie over de gebruiker of deze uniek is of niet, de sessies, enz.

Door middel van parameters kan informatie over goals en [campagnes](#) ook meegestuurd worden zodat deze volledig gebruikt kunnen worden.

6.2 Implementatie en Architectuur

De implementatie van [Google Analytics](#) is niet gelukt bij het gebruik van de [Firebase SDK](#). Dit komt doordat de [Firebase SDK](#) vraagt voor welk platform de applicatie dient en aangezien dit [cross-platform](#) is kunnen we geen specifiek platform aanduiden. Er is weliswaar de mogelijkheid om web aan te duiden, maar dit vereist dat we in de HTML een scriptje toevoegen en dit is iets dat niet gebruikt kan worden binnen [React Native](#).

Doordat we [Expo Analytics](#) onderzoeken en dit gemaakt is om [Google Analytics](#) te gebruiken binnen een [React Native](#) applicatie met [Managed Expo](#) is er eens gekeken hoe deze [tool](#) met [Google Analytics](#) juist werkt. Nadat [Expo Analytics](#) is onderzocht is er opgemerkt dat deze door middel van een url de data doorstuurt.

Er is dus een mogelijkheid om toch [Google Analytics](#) te gaan implementeren maar dan niet door het gebruik van de [Firebase SDK](#). Het is wel mogelijk om een eigen implementatie te maken net zoals [Expo Analytics](#). Er zijn een aantal voordelen aan het maken van een eigen implementatie. Zo kan men deze implementatie zelf beheren en bijwerken wanneer er problemen zouden optreden en bij het gebruik van andere [tools](#) zoals [Expo Analytics](#) moet er niet gewacht worden op updates om deze problemen op te lossen. Het grootste voordeel is dat wanneer bepaalde [tools](#) geïntegreerd worden en met deze URL werken dat deze zelf geïmplementeerd kunnen worden wat natuurlijk niet mogelijk is bij het gebruik van een [library](#) tenzij er aan de eigenaar van die [library](#) wordt gevraagd of dit geïmplementeerd zou kunnen worden.

We zouden dit dus volledig zelf kunnen maken, zonder dat we moeten afhangen van een [library](#). We gaan weliswaar hetzelfde moeten doen als de [libraries](#) zoals [Expo Analytics](#). Wat dan ook het grootste nadeel is aangezien er meer tijd ingestoken moet worden om eigenlijk hetzelfde te bekomen met [Expo Analytics](#).

Om dit te kunnen opbouwen zoals [Expo Analytics](#) zouden we dus een klasse moeten hebben dat de URL zal opbouwen om de data naar [Google Analytics](#) te sturen. Deze url zal op deze wijze opgebouwd moeten worden:

```
'https://www.google-analytics.com/collect?tid=${this.propertyId}&
v=1&cid=${this.clientId}&${hit.toQueryString()}&${params}&
${customDimensions}&${customMetrics}&
z=${Math.round(Math.random() * 1e8)}';
```

Hierbij is `propertyId` de [tracking ID](#) die te vinden is in [Google Analytics](#), `v=1` is de protocol versie van [Google Analytics](#) en is altijd 1, de `clientId` is de id van het toestel, `hit.toQueryString()` bevat de informatie over de [page hit](#), [screen hit](#), [event](#), [transactie](#) en het toevoegen van een item aan de [transactie](#). Daarnaast is er nog `params`, dit bevat toegevoegde parameters zoals de naam van een pagina die bij een [event](#) wordt meegestuurd, `customDimensions` zijn de zelfgemaakte [dimensions](#), `customMetrics` de zelfgemaakte [metrics](#). De laatste parameter in deze url is de cache buster om ervoor te zorgen dat deze request niet door browsers worden gecached.

6.3 Documentatie

Zoals vermeld bij [Expo Analytics](#) is er een variëteit aan informatie te vinden over [Google Analytics](#):

- <https://developers.google.com/analytics/> dit is de documentatie van Google.
- <https://analytics.google.com/analytics/academy/> waar tutorials te vinden zijn en certificaten behaald kunnen worden. Dit is aangeraden voor mensen die geen kennis hebben van [Google Analytics](#) en die hiermee aan de slag willen gaan.
- <https://support.google.com/analytics> dit is een forum van Google wordt ook vaak naar gerefereerd vanuit Google Academy.
- <https://support.google.com/analytics/community> dit is de community van Google waarbij men vragen kan stellen en antwoorden vinden.
- <https://analytics.google.com/analytics/gallery> hier kan men dashboards, zelfgemaakte rapporten en [segments](#) vinden.
- <https://www.youtube.com/user/googleanalytics> youtube kanaal met videos over [Google Analytics](#).

Voor degene die geïnteresseerd zijn is er ook informatie te vinden over het gebruik van de [Firebase SDK](#) om [Firebase Analytics](#) te gebruiken: <https://firebase.google.com/docs/analytics>.

Dankzij deze variëteit aan informatiebronnen is alle informatie die hierover nodig is voorhanden. Indien er toch vragen zouden zijn waarop geen antwoord te vinden is bij bovenstaande bronnen, dan zijn er ook bronnen voorhanden zoals Stack Overflow waar iedereen vragen kan stellen en beantwoorden.

6.4 Prijs / Kwaliteit

Het gebruik van [Google Analytics](#) is volledig gratis. Dit wil zeggen dat er enkel een [operationele kost](#) zal zijn. Voor deze casus zal dit dus minder dan € 10.000 per jaar kosten weliswaar is dit afhankelijk van hoeveel er moet geconfigureerd worden.

Zo zal de prijs enkel in het begin hoog liggen, tijdens de implementatie en het opzetten van [Google Analytics](#). Weliswaar zal dit nog steeds onder de € 10.000 liggen. Nadien zal deze kost lager liggen aangezien men enkel de rapporten hoeft te bekijken en de implementie te onderhouden wanneer er problemen, updates of nieuwe verwachtingen zouden optreden.

Laten we nu eens bekijken wat de mogelijke kostprijs is hiervan. Om deze [operationele kost](#) te bepalen zullen we dit gaan opsplitsen zodat alles overzichtelijker is.

In het eerste deel zal er voornamelijk het opzetten van [Google Analytics](#) bekeken worden te samen met de implementatie van de [Google Analytics](#) URL. In het tweede deel zal er voornamelijk het onderhoud van [analytics](#) worden besproken.

Na bespreking met Cognit is het volgende resultaat gekomen:

Voor het opzetten van [Google Analytics](#) en de implementatie van de [Google Analytics](#) URL zal er een gemiddelde prijs gehanteerd worden van € 400 per dag en dit voor 10 werkdagen. Dit wil dus zeggen dat het opzetten en het implementeren van [Google Analytics](#) om en bij de € 4000 kost. Dit zijn dus 5 dagen meer gerekend dan bij [Expo Analytics](#).

Nu hier zijn dan ook onderhoudskosten aan wat vooral het bekijken van rapporten is en het aanpassen van zelfgemaakt [dimensions](#), [metrics](#), [events](#) en rapporten en het verbeteren hiervan. Daarbovenop moet ook de implementatie aangepast worden om dit te laten overeenstemmen, of wanneer er problemen, updates of nieuwe verwachtingen zijn. Hiervoor wordt er ongeveer 1 dag per kwartaal aangerekend wat dus € 400 per kwartaal is of € 1600 per jaar.

Dit wil zeggen dat bij het eerste jaar de totale kost ongeveer € 5600 is, en nadien jaarlijks € 1600 hierbij komt.

Dit voor alle functionaliteiten van [Google Analytics](#), integraties met andere [tools](#) worden hier niet bijgerekend aangezien hier extra parameters gebruikt kunnen worden.

Deze prijs is weliswaar afgestemd met Cognit voor andere bedrijven kan deze prijs natuurlijk hoger of lager liggen afhankelijk van hoeveel het kost per dag, de complexiteit van het project, het aantal dagen er aan gewerkt zal worden voor het opzetten en het aantal dagen er gespendeerd zullen worden voor opvolging en onderhoud van [Google Analytics](#).

6.5 Case Studies / Getuigenissen

Over [Google Analytics](#) zijn er veel Case Studies en Getuigenissen te vinden, op de website Garnter (2020) alleen zijn er al meer dan 1600 reviews te vinden waarbij meer dan 60% zeer positief is. Weliswaar zijn deze reviews over het implementeren op een website of een andere soort mobiele applicatie. Over het gebruik van [Google Analytics](#) met [React Native](#) en [Managed Expo](#) zijn er geen case studies of getuigenissen te vinden.

7. Microsoft App Center Analytics

7.1 Data en Functies

[App Center Analytics](#) biedt maar 2 functionaliteiten en een beperkte reeks data, het biedt informatie van gebruikers en [events](#) aan te samen met een logboek over het gebruik van de applicatie in real time. De 2 functionaliteiten die [App Center Analytics](#) aanbied is het standaard analyseren van de applicatie en het meten van [events](#). Bij het bekijken van de data kan er volgende informatie verkregen worden:

- Het aantal gebruikers die de applicatie gebruiken en gebruikt hebben
- Het gebruikte toestel en besturingssysteem
- De locatie
- Welke taal er werd gebruikt
- Het aantal sessies per gebruiker
- De duurtijd van een sessie
- Het aantal actieve gebruikers per versie

Ook al wordt al deze informatie verkregen van de gebruiker, de gebruiker blijft anoniem zodat alles verloopt volgens de [GDPR](#) wetgeving. Naast het standaard analyseren van het gebruik van de applicatie kan er ook [events](#) gemeten worden. Bij het gebruik van [events](#) kan er informatie verkregen worden over het aantal keer dat een [event](#) werd getriggerd, het aantal gebruikers die het [event](#) hebben getriggerd en het gemiddeld aantal getriggerde [events](#) per gebruiker en sessie. Zo kan er ook een verandering gemeten worden in het triggeren van een [event](#), deze veranderingen uitten zich op basis van het aantal triggers van een [event](#) of het aantal gebruikers die het [event](#) hebben getriggerd.

7.2 Implementatie en Architectuur

Nu dat er is bekeken welke informatie er uit een applicatie kan worden gehaald is het tijd om te kijken hoe dit in de applicatie gebruikt moet worden om deze data te kunnen verkrijgen. Om te beginnen moet er een account worden aangemaakt op <https://appcenter.ms>, hiervoor is er een Microsoft account gebruikt er is natuurlijk ook de mogelijkheid om een ander soort account te gebruiken zoals Github, Facebook of zelfs Google.

Eens een account is aangemaakt kan er een applicatie aangemaakt worden. Hiervoor moet er eerst een naam gegeven worden, in welk stadium de applicatie zich bevind zoals alpha, beta, productie, etc. en kan er ook aangeduid worden voor welk besturingssysteem en omgeving de applicatie wordt gebouwd. Dit laatste is weliswaar een probleem aangezien de applicatie voor zowel Android als iOS is.

Wanneer de applicatie wordt aangemaakt voor iOS specifiek dan wordt er een stappenplan weergegeven hoe de applicatie moet geconfigureerd worden om [analytics](#) te kunnen gebruiken. Bij de eerste stap zou appcenter en appcenter-analytics geïnstalleerd moeten worden door middel van [npm](#). Bij de tweede stap wordt er vermeld dat er een installatie zou moeten gebeuren in de iOS map, dit is waar de implementatie wordt stop gezet.

Doordat er gedurende dit onderzoek wordt gewerkt met [React Native](#) in [Managed Expo](#), is er geen opslitsing in iOS en Android en wordt alles in één applicatie gemaakt. Dit zorgt ervoor dat de tweede stap in het configuratie process van [App Center Analytics](#) niet kunnen uitvoeren aangezien dit iOS specifieke configuraties zijn waartoe er geen toegang is, hetzelfde geldt voor het Android gedeelte.

Aangezien het onderzoek naar [App Center Analytics](#) is stopgezet kan er ook geen informatie gegeven worden over hoe de [analytics](#) geïmplementeerd moet worden en hoe de architectuur er dan zal uitzien. Het is dus zo dat [App Center Analytics](#) niet geïmplementeerd kan worden wanneer er binnen [React Native](#) wordt gewerkt met [Managed Expo](#), er moet hiervoor uit [Managed Expo](#) worden gestapt om toegang te verkrijgen tot de onderverdeling in iOS en Android.

7.3 Documentatie

Doordat er niet veel functionaliteiten zijn binnen [App Center Analytics](#), is er ook niet zo heel veel documentatie. Zo is er bijvoorbeeld wel documentatie op <https://docs.microsoft.com/en-us/appcenter/analytics/> en <https://docs.microsoft.com/en-us/appcenter/sdk/>, maar heel veel documentatie is er niet over maar het bevat wel alle nodige documentatie. Op het [App Center](#) Portaal, nadat een applicatie is aangemaakt, wordt er ook uitgelegd welke stappen er moeten gebeuren om het te kunnen gebruiken van hieruit wordt er ook verwezen naar de documentatie die hierboven werd vermeld. Ook kan informatie verkregen worden via het youtube kanaal https://www.youtube.com/channel/UC8a2fLwePFv_ywawI_IeGOA ook hier zijn er videos die meer informatie geven over [App Center Analytics](#).

Indien er toch nog vragen onbeantwoord zijn kunnen deze altijd gesteld worden via het [App Center](#) portaal door rechts boven op het vraagteken te klikken en dan te kiezen voor Contact Support. Via deze weg kan je het formulier invullen waarna er contact zal opgenomen worden via email.

7.4 Prijs / Kwaliteit

Het gebruik van [App Center](#) is gratis, maar kan uitgebreid worden indien nodig zie: <https://visualstudio.microsoft.com/app-center/pricing/>.

Nu deze prijs is weliswaar wanneer de applicatie al is opgesplitst in iOS en Android. Wanneer dit nog niet het geval is moet dit eerst nog gebeuren en moet de applicatie correct worden opgebouwd binnen elke omgeving. Nadien moet er 2 maal [App Center Analytics](#) opgezet worden. De prijs zal dan veel hoger liggen aangezien het veel tijd in neemt om de applicatie op te splitsen en op te bouwen zoals men het wilt.

Maar aangezien het niet mogelijk is volgens de documentatie op Github om dit te gaan gebruiken binnen [Managed Expo](#) is er dan ook geen prijs aan verbonden.

7.5 Case Studies / Getuigenissen

Over [App Center Analytics](#) zijn er zeer weinig reviews te vinden. Op Gartner (2020) zijn er 16 reviews van [App Center](#) waarbij er een aantal zijn die het [analytics](#) gedeelte hebben gebruikt en hierop zeer positief scoorde. De reviews die zijn gegeven gaan dan ook meestal over het algemeen gebruik van [App Center](#), wat wil zeggen dat alle andere [services](#) van de [SDK](#) ook gebruikt worden. Over het algemeen zijn er zeer weinig reviews te vinden over [App Center Analytics](#) zelf, maar deze zijn wel over het algemeen positief.

Nu zijn er weliswaar geen reviews over het gebruik van deze [analytics tool](#) binnen [React Native](#), en al zeker niet met [Managed Expo](#) aangezien de [analytics tool](#) niet gebruikt kan worden hiermee.

8. Conclusie

In deze scriptie werden de [tools Expo Analytics](#), [Google Analytics](#) en [App Center Analytics](#) met elkaar vergeleken binnen [React Native](#) met [Managed Expo](#), zie Methodologie. Hierbij is onderzocht welke [tool](#) het meest geschikt is binnen de omgeving [React Native](#) met [Managed Expo](#) en welke voor- en nadelen er aan zijn. Met deze informatie kan een antwoord gevormd worden op de onderzoeksvragen.

8.1 Conclusie onderzoek

Het onderzoek dat is uitgevoerd is per [tool](#) opgedeeld in 5 delen. Eerst hebben we bij elke [tool](#) gekeken welke functionaliteiten deze aanbieden en wat voor data we hiervan verkrijgen.

Daarna hebben we bekeken of deze [tools](#) implementeerbaar zijn of niet en wat de verkregen architectuur was.

Daarnaast hebben we ook bekeken waar er documentatie verkrijgbaar is van de [tools](#) zelf en waar er mogelijkheden zijn tot forums, support of andere kanalen waar mensen terecht kunnen met hun vragen.

Zo hebben we ook onderzocht hoeveel het kost om een [tool](#) te gebruiken en dit ook in overeenstemming is met de kwaliteit.

Als laatste hebben we ook gekeken naar verschillende getuigenissen en case studies zodat we een beeld hebben van wat anderen van deze [tools](#) vinden en of het wordt aangeraden of niet en of dit in overeenstemming is met dit onderzoek.

Laten we eens een samenvatting bekijken van dit onderzoek:

	Expo Analytics	Google Analytics	App Center Analytics
Meeste functies en data	ja	ja (enkel via URL)	enkel gebruikers en events (Bare Expo)
Beste implementatie mogelijkheid	ja	ja (enkel via URL)	Niet mogelijk
Beste documentatie en support	ja (vooral via Google)	ja	ja
Beste prijs/kwaliteit	ja (voor snelle implementatie)	ja (enkel via URL)	nee
Wordt aangeraden door case studies	geen case studies	ja (niet in deze omgeving specifiek)	ja (niet in deze omgeving specifiek)

Tabel 8.1: De verschillen tussen Google Analytics, Microsoft Visual Studio App Center Analytics en Expo Analytics volgens het onderzoek.

Wanneer we bovenstaande tabel bekijken dan kunnen we concluderen dat [Expo Analytics](#) het meest geschikt is wanneer een snelle implementatie nodig is. Daarnaast is [Google Analytics](#) enkel geschikt door het te implementeren op basis van de URL zodat we hetzelfde resultaat bekomen als [Expo Analytics](#). Alleen bij [Google Analytics](#) moet men de [Expo Analytics library](#) zelf aanmaken en beheren. Verder kan [App Center Analytics](#) niet geïmplementeerd worden tenzij men overstapt van [Managed Expo](#) naar [Bare Expo](#). Met deze informatie kunnen we een antwoord vormen op de verschillende onderzoeksvragen.

8.2 Antwoorden op de onderzoeksvragen

8.2.1 Welke tool biedt de meeste mogelijkheden (collecteren en verwerking van data)?

Zowel [Google Analytics](#) als [Expo Analytics](#) bieden veel mogelijkheden aan met betrekking tot het collecteren en verwerken van data. Dit komt doordat [Expo Analytics](#) ook gebruik maakt van [Google Analytics](#), alleen biedt het een [library](#) aan voor het gebruik binnen [Managed Expo](#), iets dat [Google Analytics](#) niet aanbiedt. Hierdoor hebben beide ondersteuning voor anonieme gebruiksinformatie, [page hits](#), [events](#), [transactions](#), custom [dimensions](#) en custom [metrics](#). [App Center Analytics](#) is daarentegen gelimiteerd tot gebruiksinformatie en [events](#) (weliswaar anoniem) en is bovendien niet implementeerbaar binnen [Managed Expo](#).

8.2.2 Welke tool werkt het beste samen met Managed Expo?

Van deze 3 tools werkt enkel [Expo Analytics](#) binnen [Managed Expo](#), omdat deze gericht is op het gebruik van [Google Analytics](#) binnen [Managed Expo](#). Natuurlijk kan [Expo Analytics](#) ook zelf opgebouwd worden aangezien het gewoon de url van [Google Analytics](#) gebruikt om data te versturen. Hierdoor kan ook [Google Analytics](#) zelf gebruikt worden binnen [Managed Expo](#), weliswaar enkel door de url te gebruiken om data te verzamelen. Dit wil zeggen dat de [SDK](#) of [Firebase Analytics](#) niet geïmplementeerd kan worden, dit is omdat deze platform specifiek zijn terwijl [React Native](#) met [Managed Expo](#) voor meerdere platformen zijn. Dit is ook de reden bij [App Center Analytics](#) waardoor deze niet implementeerbaar is.

8.2.3 Welke tool is het meest eenvoudige/gebruiksvriendelijke te implementeren zonder bijkomende aanpassingen?

De meest eenvoudige tool om te gebruiken is [Expo Analytics](#), aangezien dit een library is gericht op [Managed Expo](#). Weliswaar is er niet veel documentatie, maar alles is wel voorhanden. Ook kan er informatie verkregen worden van [Google Analytics](#) zelf. Dit is dus ideaal voor een snelle implementatie, weliswaar als er problemen opduiken zullen deze wel niet onmiddellijk opgelost worden aangezien dit wordt onderhouden door een beperkt team.

[Google Analytics](#) zelf daarentegen kan niet geïmplementeerd worden behalve op basis van de URL zoals [Expo Analytics](#) is opgebouwd, ook indien er problemen opduiken kunnen die gemeld worden en zal deze sneller opgelost kunnen worden aangezien een groot team hieraan werkt. Deze oplossing is dus het meest ideale op langere termijn aangezien het niet afhankelijk is van een library en makkelijker te onderhouden is om problemen sneller op te lossen.

[App Center Analytics](#) daarentegen kan niet geïmplementeerd worden aangezien dit gericht is op applicaties die platform specifiek zijn wat enkel mogelijk is met [Bare Expo](#). Hierdoor is deze tool dan ook totaal niet eenvoudig of gebruiksvriendelijk aangezien er heel veel bijkomende aanpassingen moeten gebeuren omdat de applicatie in 2 wordt gesplitst en deze apart moeten geconfigureerd worden.

8.2.4 Wat zijn de voor- en nadelen bij het gebruik van de tool?

De tabel 8.1: 'Samenvatting onderzoeksvragen alleen voor Managed Expo' geeft een goed overzicht van de verschillen tussen de 3 tools.

Zo heeft [Expo Analytics](#) als voordelen dat het een kant en klare library is om analytics te gebruiken en het makkelijk is om de tool te configureren. Het nadeel is dat er beperkte ondersteuning is, waardoor het langer duurt alvorens problemen opgelost zijn.

[Google Analytics](#) met URL heeft als voordeel dat het niet afhangt van een [library](#) waardoor het eenvoudiger te beheren is en er niet moet gewacht worden tot wanneer een probleem is opgelost. Het nadeel is dat er zelf een implementatie nodig is om het te kunnen gebruiken waardoor het geen ideale oplossing is voor snelle implementaties.

[App Center Analytics](#) heeft geen voordelen aangezien het niet gebruikt kan worden. Het heeft wel veel nadelen het is dus zo dat er geen ondersteuning is voor [Managed Expo](#) en het enkel gebruikt kan worden met [Bare Expo](#). Bovendien duurt het langer om deze dan ook te implementeren aangezien het per applicatie geïmplementeerd moet worden. Als laatste nadeel [App Center Analytics](#) verzamelt enkel informatie over de gebruikers en de [events](#) wat dus veel minder is dan wat mogelijk is met [Google Analytics](#).

8.2.5 Welke tool is integreerbaar met bestaande Analytics (momenteel Google Analytics)?

Zowel [Expo Analytics](#) als [Google Analytics](#) kan gebruikt worden om aan te sluiten aan de bestaande [Google Analytics](#) door gewoon de [tracking ID](#) door te geven. [App Center Analytics](#) daarentegen biedt hiervoor geen ondersteuning behalve met [Microsoft Azure Application Insights](#) wat dient om applicaties en websites te monitoren.

Integraties met andere [tools](#) is weliswaar niet mogelijk, behalve bij [Google Analytics](#) en [Expo Analytics](#) indien parameters met de url mee verzonden kunnen worden, of als een eigen implementatie van deze integratie mogelijk is. Dit zal dan verder onderzocht moeten worden om te kunnen bepalen of dit mogelijk is.

8.2.6 Wie heeft de ownership van de data en wat met privacy (GDPR)?

Bij zowel [Expo Analytics](#) als [Google Analytics](#) is [Google Analytics](#) de eigenaar van de data, maar de gebruiker kan wel de data van gebruikers verwijderen indien nodig. Bij [App Center Analytics](#) is dit eerder van Microsoft zelf, of de data ook verwijderd kan worden is onbekend aangezien de implementatie niet was gelukt.

Alle 3 de [tools](#) volgen de [GDPR](#) regels en alle data die verzamelt wordt is anoniem. Verder moet de applicatie wel een melding geven en de toestemming vragen of [analytics](#) gebruikt mag worden om in orde te zijn met de [GDPR](#) regels.

	Google Analytics	App Center Analytics	Expo Analytics
Meest gedetailleerde rapporten	ja (enkel via URL)	/	ja
Meeste functies voor data verzameling	ja (enkel via Url)	/	ja
Werkt het best met Managed Expo	nee	nee	ja
Meest gebruiksvriendelijke qua implementatie	ja(enkel via Url)	nee	ja
Is integreerbaar met bestaande web analytics	ja (enkel via Url)	/	ja
Wie heeft ownership van de data	Google Analytics	/	Google Analytics
Wat met privacy (GDPR)?	Data is anoniem, verbiedt data die gelinkt kan worden aan de gebruikers.	/	Data is anoniem, verbiedt data die gelinkt kan worden aan de gebruikers.
Meeste voordelen?	ja(enkel via Url)	nee	ja
Minste nadelen?	ja(enkel via Url)	nee	ja(op Google Analytics met URL na)

Tabel 8.2: Samenvatting onderzoeksvragen alleen voor Managed Expo

8.3 Conclusie van deze bachelorproef

We kunnen concluderen dat enkel [Expo Analytics](#) binnen deze omgeving werkt, maar ook dat er een eigen implementatie mogelijk is voor [Google Analytics](#) door het opbouwen van een url en deze data daarnaar te verzenden. Het enige nadeel is dat [Expo Analytics](#) met een versie beheer zit en dus niet enkel kan werken met de geschikte versies. Wanneer deze [library](#) niet wordt geüpdatet dan zit men vast op die versie. Dit is iets wat beheerd kan worden bij een eigen implementatie.

[Expo Analytics](#) is hierbij gebruiksvriendelijker om dat het grootste deel van de implementatie al gedaan is en deze makkelijk te gebruiken is. Ook al is er niet veel documentatie hierover er is ook de mogelijkheid om deze informatie via [Google Analytics](#) te verkrijgen of via de community van [Expo Analytics](#).

Doordat [Expo Analytics](#) [Google Analytics](#) gebruikt, hebben ze beide hetzelfde op basis van het collecteren en verwerken van data.

Het nadeel bij [Expo Analytics](#) is de beperking om enkel [Google Analytics](#) te gebruiken en minder mogelijkheden heeft tot integraties met andere [tools](#). Het grootste voordeel is dat deze te gebruiken is met een bestaand [Google Analytics](#) account, en er minder werk is om deze te implementeren. [Google Analytics](#) heeft als voordeel dat via de URL er meer mogelijkheden zijn tot integraties met andere [tools](#) waarbij parameters naar [Google Analytics](#) gestuurd kunnen worden.

Beide zijn conform aan de [GDPR](#) regels aangezien er geen data wordt doorgestuurd waarmee een gebruiker geïdentificeerd kan worden.

Met al deze verschillende punten kunnen we concluderen dat [Expo Analytics](#) de meeste voordelen heeft en [Google Analytics](#) de minste nadelen heeft. Wanneer men vooral gericht is op een [tool](#) met integraties, onafhankelijkheid en vooral toekomstgericht is dan is [Google Analytics](#) door middel van url de beste keuze. Wanneer het vooral gericht is op minimaal gebruik en implementatie is [Expo Analytics](#) de beste keuze.

De andere opties zijn niet aangeraden doordat deze niet met [Managed Expo](#) werken en eerder voor [Bare Expo](#) geschikt zijn. Dit zal als resultaat een langere implementatie en configuratie periode opleveren te samen met het optreden van verschillende problemen. Hierbij zal de kost dan ook enorm verhogen voor een applicatie dat als resultaat hetzelfde zal opleveren.

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

Dit onderzoek zal de organisatie inzicht en informatie verschaffen over het gebruik van de mobiele applicatie door de klanten van Cogn-IT. Hierdoor kan er gericht ingezet worden op verdere ontwikkelingen en kan deze data gebruikt worden voor promotionele doeleinden. Waardoor er gericht ingezet kan worden op verdere ontwikkelingen en kan deze data gebruikt worden voor promotionele doeleinden. Het bepalen van de geschikte Analytics tool zal gebeuren op basis van een aantal onderzoeksvragen.

A.2 Literatuurstudie

De dag van vandaag worden Analytics tools vaak gebruikt, maar de vraag is meestal welke tool de geschikte is. Er is gekozen om Google Analytics, Microsoft App Center Analytics en Expo Analytics te vergelijken met elkaar, doordat deze het dichtste aanleunen bij de omgeving. Aangezien Google Analytics al binnen Cogn-IT wordt gebruikt dan zou het wenselijk zijn om dit ook te gebruiken voor mobiele applicaties. Cogn-IT is een office 365 intranet oplossing en dus zou ook het gebruik van App Center Analytics mogelijk zijn aangezien dit ook van Microsoft is. De derde tool waarmee vergeleken wordt is Expo Analytics, een plugin om Google Analytics te gaan gebruiken. Dit werd gekozen aangezien de mobiele applicatie binnen een React Native met Managed Expo omgeving werkt.

Er is weliswaar geen bestaande of vergelijkbare studie over dit onderwerp in deze omgeving (voor zover dat er gezocht is). Alvorens dieper op de Analytics tools in te gaan, zullen we eerst meer informatie geven over de omgeving.

A.2.1 React Native

Het onderzoek zal binnen React Native plaatsvinden doordat er een applicatie van de organisatie in deze omgeving werkt, maar wat is React Native en wat doet het precies?

React native is een framework om het React framework te kunnen gebruiken op iOS en Android, waarbij de native UI (User Interface) gebruikt wordt en waarbij er dus volledige toegang is tot het native platform. React Native is zo gemaakt dat er 1 app gemaakt kan worden dat zowel voor Android als iOS gebruikt kan worden. (Facebook, 2019)

React zelf is een javascript library dat is ontwikkeld door Facebook. Dit wil zeggen dat React geen volwaardig MVC framework is aangezien het enkel de view meegeeft en de rest moet hierbij dan zelf bepaald worden afhankelijk van de noden van het project. Hierbij moet er weliswaar zelf gezorgd worden voor de nodige updates en migraties van de libraries waardoor er meer fout kan gaan.

A.2.2 Managed Expo

Expo is een framework en een platform voor universele React Applicaties, het is een set van tools en services dat gebouwd is rond React Native en native platformen die helpen om een applicatie te ontwikkelen voor zowel iOS, Android als Web apps op basis van 1 gemeenschappelijke code.

Wanneer Expo gebruikt wordt zijn er 2 werkwijzen om dit te gaan gebruiken. Er is de “Bare” en de “Managed” workflow. Managed Expo wilt zeggen dat er enkel Javascript/Typescript wordt geschreven en Expo tools en services eerder zorgen voor de rest, terwijl bij Bare Expo de ontwikkelaar de volledige controle heeft hierover waardoor Expo tools niet veel helpen. (Expo, 2019)

Wanneer React Native gebruikt wordt zonder Expo dan is dit de Bare workflow. Gedurende dit onderzoek zal er gewerkt worden met Managed Expo aangezien dit aanwezig is in de huidige applicatie en de organisatie wenst zo min mogelijk tot geen aanpassingen te doen aan hun applicatie wanneer ze analytics zouden implementeren.

A.2.3 Analytics

Gedurende dit onderzoek zullen 3 tools met elkaar vergeleken worden, Google Analytics, Azure Analytics en Expo Analytics.

Google Analytics is een tool dat toelaat om websites, blogs en sociale netwerken op te volgen.

Daarnaast biedt het vooraf bepaalde en aanpasbare rapporten aan en voert het Return On Investment (ROI, geeft het rendement op de investering aan) metingen uit. Google Analytics is dan ook de meest gekende en gebruikte Analytics tool, doordat ze de mogelijkheid hebben om te gaan combineren met andere tools van Google. (Editorial, 2018)

Microsoft App Center Analytics is een tool dat helpt om gebruikersgedrag en klantbetrokkenheid te begrijpen om uw app te verbeteren. Het is een Software development kit (SDK) dat automatisch het aantal sessies en apparaat eigenschappen vastlegt, zoals model, OS versie, enz. (Microsoft+contributors, 2019)

Expo Analytics is een plugin dat Google Analytics integreert voor het gebruik met React Native applicaties die gemaakt zijn op Expo. Wanneer Google Analytics gebruikt wordt in deze omgeving dan zal bij de meeste libraries linking nodig zijn met Google Analytics wat niet ondersteund wordt door Expo. Deze plugin zorgt ervoor dat Google Analytics geïntegreerd kan worden zonder dat deze gelinkt hoeft te worden. (ryanvanderpoel, 2019)

In de bachelorproef zal hier dieper op ingegaan worden waarbij er grondig zal vergeleken worden tussen de verschillende tools.

A.3 Methodologie

Gedurende dit onderzoek zal er een literatuurstudie opgesteld worden met alle informatie omtrent mobiele applicaties, React Native, Expo, Managed Expo en de verschillende Analytics tools. In de literatuurstudie zal er een vergelijkende studie gedaan worden op basis van informatie die online gevonden wordt.

Na de literatuurstudie zal de uitwerking van de 3 mobiele applicaties komen. Hiervoor zal er gebruik gemaakt worden van een basis demo applicatie die gebruikt zal worden als basis voor de 3 tools. Dit wil zeggen dat per tool dezelfde basis gebruikt zal worden en dat elke tool apart zal onderzocht worden.

Wanneer de 3 experimenten uitgewerkt zijn dan zal deze vergeleken worden en zal er onderzocht worden op volgende punten:

- Welke tool biedt de meeste mogelijkheden (collecteren en verwerking van data)?
- Welke tool werkt het beste samen met Managed Expo?
- Welke tool is het meest eenvoudige/gebruiksvriendelijke te implementeren zonder bijkomende aanpassingen?
- Wat zijn de voor- en nadelen bij het gebruik van de tool?
- Welke tool is integreerbaar met bestaande Analytics (momenteel Google Analytics)?
- Wie heeft de ownership van de data en wat met privacy (GDPR)?

Wanneer het onderzoek voltooid is dan zullen de resultaten verder besproken worden waarvan een conclusie zal worden getrokken.

A.4 Verwachte resultaten

Als resultaat is het wenselijk dat Google Analytics de meest geschikte is aangezien deze al gebruikt wordt door de organisatie. Het is mogelijk dat Expo Analytics voordeliger is doordat deze de implementatie van Google Analytics vergemakkelijkt en er geen linking nodig is. Microsoft App Center Analytics zal dan het minst wenselijk zijn aangezien deze minder mogelijkheden aanbiedt, zoals minder integratie en rapporteringsmogelijkheden en data verzameling.

A.5 Verwachte conclusies

De te verwachte conclusie is dat Google Analytics de meest geschikte tool is doordat deze de meeste integratie en rapporteringsmogelijkheden heeft. Deze is ook het meest gebruiksvriendelijke om te implementeren, bovendien is de Google Analytics website zeer overzichtelijk en makkelijk te beheren ook in combinatie met andere Google Producten.

Bibliografie

- Editorial. (2018). What is Google Analytics and how does it work. Verkregen van <https://www.antevenio.com/usa/what-is-google-analytics-and-how-does-it-work/>
- Elkin, A. (2018). Continuous Learning for Your iOS App Using App Center. Verkregen van <https://www.sitepoint.com/continuous-learning-ios-app-using-app-center/>
- Expo. (2019). What is Expo? Verkregen van <https://docs.expo.io/versions/latest/>
- Facebook. (2019). React Native. Verkregen van <https://github.com/cnpmjs.org/facebook/react-native>
- Garnter, G. (2020). Google Analytics by Google in Web and mobile App Analytics. Verkregen van <https://www.gartner.com/reviews/market/web-mobile-app-analytics/vendor/google/product/google-analytics/reviews?sort=-helpfulness>
- Gartner, M. (2020). Visual Studio App Center by Microsoft in Application Development, Integration and Management - Others. Verkregen van <https://www.gartner.com/reviews/market/application-development-integration-and-management-others/vendor/microsoft/product/visual-studio-app-center/reviews?sort=-helpfulness>
- Google. (2019). The process of collecting Analytics data. Verkregen van <https://support.google.com/analytics/answer/6383007?hl=en>
- Google. (2020). De hiërarchie van accounts, gebruikers, property's en dataweergaven. Verkregen van <https://support.google.com/analytics/answer/1009618?hl=nl>
- Grover, D. (2018). Building an e-commerce search app with React Native. Verkregen van <https://medium.com/hackernoon/building-an-e-commerce-search-app-with-react-native-2c87760a2315>
- Hughes, J. (2019). What is Web Analytics? Your 101 on Analytics and How to Get Started. Verkregen van <https://themeisle.com/blog/what-is-web-analytics/>
- Korolev, S. (2019). Mobile App Development Approaches Explained. Verkregen van <https://railsware.com/blog/native-vs-hybrid-vs-cross-platform/>

- Microsoft. (2019). Visual Studio App Center SDK for React Native. Verkregen van <https://github.com/microsoft/appcenter-sdk-react-native>
- Microsoft, D. (2019). Get Started with React Native. Verkregen van <https://docs.microsoft.com/en-us/appcenter/sdk/getting-started/react-native>
- Microsoft+contributors. (2019). App Center Analytics (React Native). Verkregen van <https://docs.microsoft.com/en-us/appcenter/sdk/analytics/react-native#wait-for-js-to-enable-app-center-analytics>
- Oladapo, D. (2019). Application Analytics with App Center. Verkregen van <https://daraoladapo.com/application-analytics-with-app-center/>
- ryanvanderpoel. (2019). Expo Analytics. Verkregen van <https://github.com/ryanvanderpol/expo-analytics>
- Su, B. (2017). What is Google Analytics, and why is it important to my business. Verkregen van <https://medium.com/analytics-for-humans/what-is-google-analytics-and-why-is-it-important-to-my-business-8c083a9f81be>
- Trends, G. (2020). Explore what the world is searching. Verkregen van <https://trends.google.com/trends/explore?date=all&q=%2Fm%2F02gcn9>
- TrustRadius. (2020). List of Analytics tools. Verkregen van <https://www.trustradius.com/products?q=analytics>