



Professionele Bachelor Toegepaste Informatica



Videoconferentie in Flutter

Roel Hermans

Promotoren:

Mike Dumont
Carine Derkoningen

Yunify
Hogeschool PXL Hasselt





Professionele Bachelor Toegepaste Informatica



Videoconferentie in Flutter

Roel Hermans

Promotoren:

Mike Dumont
Carine Derkoningen

Yunify
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2018-2019

Dankwoord

Als eerste wil ik Yunify, en vooral Peter Kraus, bedanken om voor de eerste keer samen te werken met de PXL en mij hierdoor de kans te bieden om een aangename, nuttige en leerrijke stage te mogen uitvoeren.

Daarnaast wil ik mijn collega's binnen het Atlas project, met name Mike Dumont die tevens mijn stagepromotor en technische begeleider was, bedanken om altijd voor vragen open te staan, voor de goeie sfeer in het team en voor alle overgedragen kennis.

Van de PXL wil ik ook iedereen bedanken die mijn stage mogelijk heeft gemaakt. Hieronder volgen een aantal personen die ik specifiek wil noemen.

Carine Derkoningen zou ik hartelijk willen bedanken. Zelfs buiten schooluren stond zij altijd voor mij klaar en ze was altijd goed op de hoogte van wat de PXL van mij verwachtte. Ook voor haar uitgebreide feedback op de draft versies van de bestanden ben ik haar dankbaar. Ik zou me geen betere hogeschoolpromotor kunnen bedenken.

Verder wil ik de PXL taallectoren, mevrouw Simon en mevrouw Sleurs, bedanken om zich beschikbaar te stellen aan alle studenten die taal gerelateerde vragen hadden en deze altijd snel en in detail hebben willen beantwoorden.

Als laatste wil ik nog mijn vrienden, medestudenten en vooral mijn ouders bedanken voor hun mentale steun en geduld tijdens deze stageperiode.

Abstract

Binnen het Atlas-project van Yunify is er een tool nodig die voor het bedrijf de functionaliteit van een videoconference verzorgt. Yunify wil de dienstverleners serviceproviders de mogelijkheid bieden om een videoconference te houden met hun klanten. Omdat het ontwikkelen van een videoconferencetool veel tijd en resources in beslag neemt, is het bedrijf op zoek naar een externe oplossing.

Er worden binnen Atlas twee applicaties gecreëerd. Het team zelf maakt een webapplicatie met behulp van het React-framework en er wordt een mobiele applicatie gemaakt met het Flutter-framework. Vanuit deze Flutter-applicatie worden native apps voor iOS en Android gecompileerd om zo uiteindelijk een app te verkrijgen voor beide besturingssystemen. Tussen deze web- en mobiele applicaties moet de mogelijkheid bestaan om een videoconference te houden. Dit betekent dat er communicatie moet zijn tussen alle frameworks die toegang willen tot de conference. Om dit te realiseren moet de tool een implementatie bieden voor de gebruikte frameworks.

In het onderzoek worden verschillende videoconferencingtools met elkaar vergeleken om erachter te komen welke tool het beste zou passen in deze situatie. Dit wordt gedaan door vast te stellen hoe goed de tools aan de verschillende vereisten voldoen en eveneens door eigen bevindingen oordeelkundig te vergelijken met externe bronnen. De tool moet aan een aantal minimale vereisten voldoen, zoals een goede videokwaliteit en een opnamefunctie. Ook zijn er verschillende *nice-to-have* functies die invloed hebben op het onderzoek. Daarnaast dient de tool een implementatie te bieden voor het Flutter-framework dat op het moment van dit project pas recent gelanceerd is. Dit framework is reeds gekozen door het projectteam en hiervan afwijken is dan ook niet zo maar mogelijk.

Na het vergelijken worden er *proof of concepts* gemaakt van de tools die het beste uit het onderzoek zijn gekomen. Deze *proof of concepts* worden opnieuw vergeleken en beoordeeld aan de hand van de minimale vereisten die eraan gesteld zijn. Hierna wordt de beste optie verder uitgewerkt binnen de stageopdracht. De stageopdracht bestaat in het implementeren van de functionaliteit van een videoconference met behulp van een externe tool in een Flutter-applicatie.

Inhoudsopgave

Dankwoord	ii
Abstract	iii
Inhoudsopgave	iv
Lijst van gebruikte tabellen	vii
Lijst van gebruikte afkortingen.....	viii
Inleiding.....	1
1. Bedrijfsvoorstelling.....	2
1.1 IIC Group.....	2
1.2 Yunify.....	3
1.3 Atlas.....	3
I. Onderzoekstopic.....	4
1. Onderzoeksvraag.....	4
2. Onderzoeksmethode.....	5
3. Uitwerking onderzoek	6
3.1 Mogelijke implementatie methodes	6
3.1.1 Beoordelingscriteria	6
3.1.2 Verschillende methodes	6
3.1.3 Conclusie	8
3.2 Ondersteuning.....	9
3.2.1 Tools met SDK's	9
3.2.2 Gewenste SDK's.....	10
3.3 <i>Proof of concept</i> mogelijkheden.....	11
3.3.1 Vereisten	11
3.3.2 Agora io	12
3.3.3 CometChat.....	12
3.3.4 Pubnub	12
3.3.5 TokBox.....	12
3.3.6 Vidyo io.....	13
3.3.7 Voxeet.....	13
3.3.8 Webex.....	13
3.3.9 Conclusie	13
4. Proof of Concepts.....	14
4.1 <i>Scope proof of concept</i>	14
4.1.1 Initialisatiescherm	14

4.1.2	Wachtkamerscherf	14
4.2	Beoordelingsaspecten	15
4.2.1	Documentatie	15
4.2.2	Implementatie	15
4.2.3	Netwerkprestaties	16
4.2.4	Overige beschikbare functies	16
4.2.5	Prijs	17
4.3	Agora	17
4.3.1	Documentatie	17
4.3.2	Implementatie	17
4.3.3	Netwerkprestaties	18
4.3.4	Overige beschikbare functies	18
4.3.5	Prijs	18
4.4	Voxeet	19
4.4.1	Documentatie	19
4.4.2	Implementatie	19
4.4.3	Netwerkprestaties	19
4.4.4	Overige beschikbare functies	20
4.4.5	Prijs	20
4.5	Vidyo	20
4.5.1	Documentatie	20
4.5.2	Implementatie	21
4.5.3	Netwerkprestaties	22
4.5.4	Overige beschikbare functies	22
4.5.5	Prijs	23
4.6	Conclusie	23
	Algemene conclusie	24
II.	Stageverslag	25
1.	Technologieën	25
1.1	WebRTC	25
1.2	Flutter	25
1.2.1	Dart	25
1.2.2	Hot reload	25
1.2.3	Platform specifieke kanalen	25
1.2.4	Pub	26
2.	Applicatie uitwerking	26

2.1	Gebuiikersomgeving verbergen.....	26
2.2	Omkeren van de camera	26
2.3	Beweegbaar eigen scherm	27
2.4	Vergroot modus.....	27
2.5	Voorkeuren.....	27
2.6	Thema.....	27
2.7	Rating.....	28
3.	Conclusie	28
4.	Reflectie.....	29
	Bibliografie	30

Lijst van gebruikte tabellen

Tabel 1. Vergelijkingstabel verschillende methodes.....	8
Tabel 2. Potentiële tools.....	9
Tabel 3. Relevante SDK's	11
Tabel 4. Overige functies Agora	18
Tabel 5. Overige functies Voxeet.....	20
Tabel 6. Overige functies Vidyo.....	22
Tabel 7. Vergelijkingsmatrix Tools.....	23

Lijst van gebruikte afkortingen

Agora	Agora io
API	Application Programming Interface
IIC Group	Investors In Care Group
POC	Proof of Concept
SDK	Software Development Kit
URL	Uniform Resource Locator
Vidyo	Vidyo io
WebRTC	Web Real-Time Communication

Inleiding

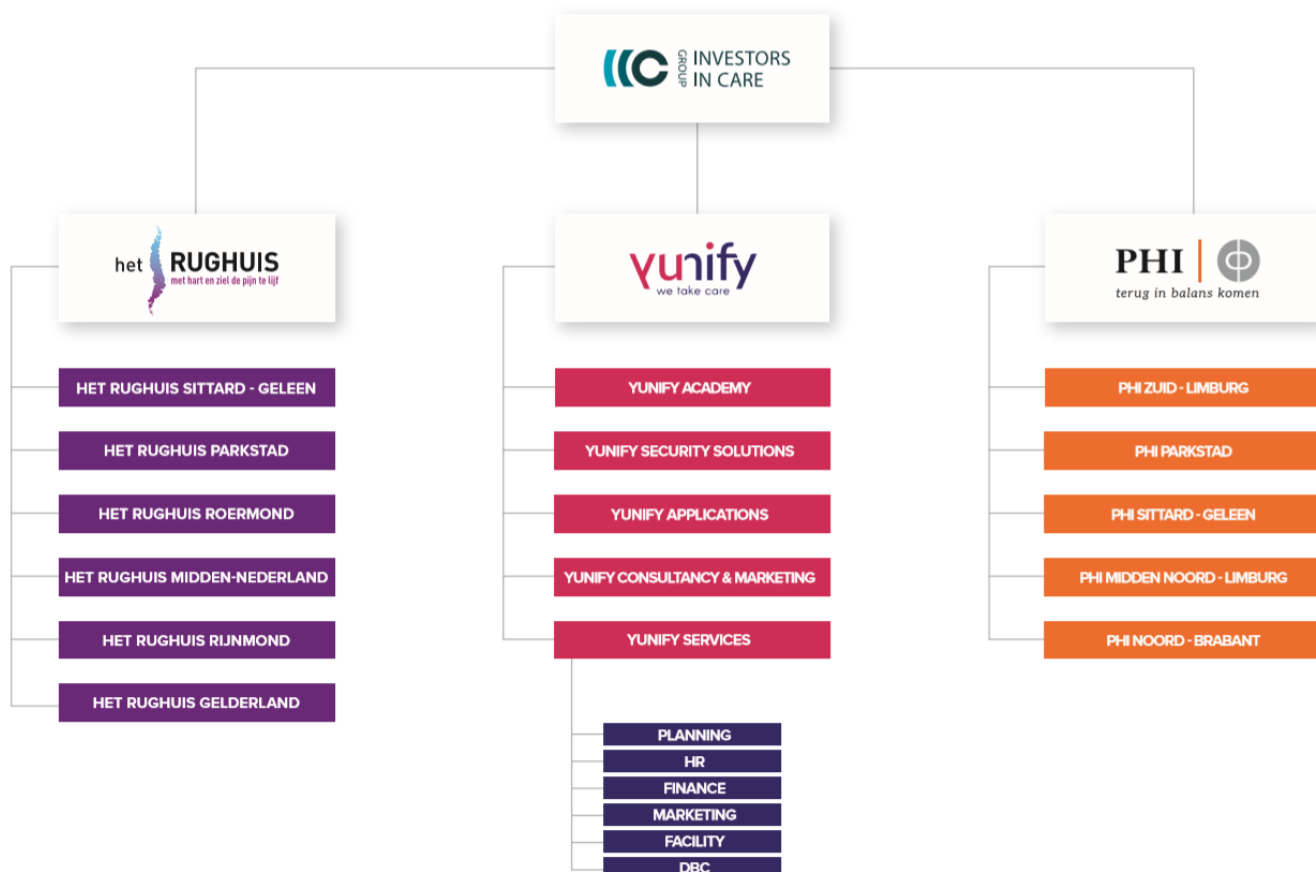
In deze tijd waar mensen niet van wachten houden en tijd geld is, is het belangrijk dat alles zo snel en efficiënt mogelijk gebeurt. Met de app genaamd Atlas, is het de bedoeling dat de connectie tussen klant en dienst geïnnoveerd wordt. Bij een huisarts moet er nog altijd gewacht worden in de wachtkamer, om een nieuwe bankrekening af te sluiten moet de persoon meestal fysiek aanwezig zijn, en zo zijn er nog talloze andere voorbeelden van plekken waar tijdbesparing mogelijk is. Deze activiteiten wil Yunify innoveren met de gastvrijheidsoplossing die zij aan het ontwikkelen zijn. Er ontbreekt echter nog een essentieel onderdeel van de oplossing, namelijk een videoconferentie houden tussen cliënten en serviceproviders.

In deze opdracht wordt er uitgezocht op welke manier de functionaliteit van een videoconferentie het best geïmplementeerd kan worden binnen het Flutter-framework. Dit moet met uiterst gebruiksgemak gebeuren. Het moet ook onder andere vloeiend werken, hoge prestaties hebben en veilig zijn. Daarbovenop draait het project er vooral om dat het functioneel moet zijn binnen het Flutter-framework.

Er wordt onderzocht wat de beste manier is om een videoconferentie te implementeren op basis van vergelijkingen tussen verschillende technologieën die op dit moment beschikbaar zijn. Als er een conclusie is getrokken, wordt de verkozen tool ook verder uitgewerkt in het stageverslag van het project.

1. Bedrijfsvoorstelling

In dit organogram wordt er uitgebeeld hoe de structuur van de organisatie in elkaar steekt en uit hoeveel verschillende divisies en afdelingen de organisatie bestaat. Dit wordt verder uitgelegd in de volgende hoofdstukken.



1.1 IIC Group

Yunify is lid van de Investors In Care Group. Deze IIC Group is een investeringsgroep en holdingmaatschappij van een aantal innovatieve bedrijven met een focus op "caring". Een van deze bedrijven is dus Yunify. De IIC Group ontwikkelt diensten en producten om mensen en organisaties gezonder te maken. Zorgen en ontzorgen zijn de sleutelbegrippen van de ondernemingen waarin deze groep investeert. [1]

Dit doen ze door zelf initiatieven te nemen en door in dienstverlenende bedrijven te investeren, het management te versterken, ondernemerservaring in het bedrijf te brengen én een honderdprocentbelang of meerderheidsbelang te nemen.

1.2 Yunify

Yunify wil gezondheid en welzijn innoveren met oplossingen die levenskwaliteit en vitaliteit verbeteren door middel van nieuwe applicaties die kunnen helpen bij het alledaagse leven in de zorgsector. Een aantal van hun *Unique Selling Points*: vitaliteit en duurzame inzetbaarheid, stress en stresspreventie en als laatste, gastvrijheid.[2] Yunify bevindt zich in The Cubic te Geleen, Nederland. Hier worden ook conferenties, trainingssessies van Yunify Academy en andere evenementen georganiseerd.

1.3 Atlas

Yunify heeft meerdere teams die aan verschillende projecten werken. De stage vindt plaats binnen het project dat als werknaam op het moment 'Atlas' heeft. Atlas is een Customer engagement systeem voor kleine en middelgrote dienstverleners. De software oplossing die zij aan het ontwikkelen zijn, verbindt dienstverleners met hun klanten, zodat deze afspraken kunnen plannen en voorbereiden, met elkaar kunnen communiceren en informatie op een veilige manier kunnen delen en opslaan. Het project gaat bijdragen aan het stukje van communiceren door middel van de videoconference.

I. Onderzoekstopic

1 Onderzoeksvraag

Het doel van het onderzoek is uit te zoeken wat **de beste manier is om de functionaliteit van een videoconferentie te integreren in een Flutter-applicatie voor de specifieke situatie binnen het project**. Er wordt vastgesteld wat de verschillende vereisten zijn en over welke functies de externe tool minimaal moet beschikken. In dit eindwerk betekent een externe tool, een service die door een ander bedrijf geleverd wordt.

Eenzijds moet de tool de mogelijkheid bieden om de functionaliteit van een videoconferentie te implementeren in de Flutter- en webapplicatie. Anderzijds moet deze functionaliteit met zo weinig mogelijk lijnen code of aanpassingen aan de code, geïmplementeerd kunnen worden. Kort samengevat ligt de focus op het zo eenvoudig mogelijk houden van het implementeren van de tool in de applicaties, terwijl hij nog altijd alle functies biedt die nodig zijn.

Bij het kiezen van een tool is het, naast het ondersteunen van Flutter-applicatie, ook belangrijk dat de tool ondersteuning biedt voor webapplicaties. De reden hiervoor is dat er samenwerking moet zijn tussen het webportaal en de Flutter-applicatie. Hierdoor wordt de tool in beide applicaties geïmplementeerd.

Er wordt onderzocht welke externe tool het beste in deze situatie past. Hieruit worden een hoofdvraag en een aantal deelvragen opgesteld:

Wat is de beste manier om de functionaliteit van een videoconferentie te krijgen binnen een Flutter-applicatie met behulp van een externe tool?

Hiervan worden de volgende deelvragen afgeleid:

Welke mogelijke implementatiemethodes zijn er om de gewenste functionaliteit te verkrijgen en welke is de beste?

Welke mogelijke tools zijn er beschikbaar die zowel web ondersteunen als Flutter?

Van welke tools kan er het beste een *proof of concept* gemaakt worden als er rekening wordt gehouden met de vereisten?

Met welke tool kan de functionaliteit van een videoconferentie het beste geïmplementeerd worden in een Flutter-applicatie?

2. Onderzoeksmethode

Het onderzoek bestaat uit twee delen. In het eerste deel wordt er een deskresearch uitgevoerd in de vorm van een literatuurstudie. Dit wil zeggen dat er onderzoek wordt verricht door te zoeken naar gerenommeerde bronnen die over tools van videogesprekken een artikel hebben gepubliceerd. Op basis van deze bronnen worden verschillende technologieën en tools naast elkaar gezet en met elkaar vergeleken. Hierbij wordt er vooral rekening gehouden met de specifieke vereisten en functies die nodig zijn voor het project. Door te vergelijken op basis van de verschillende criteria wordt hier uiteindelijk een top drie van gemaakt.

In het tweede deel wordt er een experimenteel onderzoek gedaan door een *proof of concept* op te stellen voor elke tool die geselecteerd is. Dit wordt gedaan door een vast aantal functies te definiëren en vervolgens de tools te vergelijken op basis van deze functies.

Ten slotte wordt op basis van het onderzoek een conclusie geformuleerd over welke tool de beste keuze is voor deze situatie. Deze wordt dan ook daadwerkelijk verder ontwikkeld in de stageopdracht om hem uiteindelijk te integreren in het totale project van Atlas.

3. Uitwerking onderzoek

In de volgende hoofdstukken wordt de scope van de deelvragen gedefinieerd en vervolgens voor elke deelvraag een conclusie getrokken.

3.1 Mogelijke implementatie methodes

Welke mogelijke implementatiemethodes zijn er om de gewenste functionaliteit te verkrijgen en welke is de beste?

Het doel van dit onderzoek is om te bepalen wat de beste manier is om de functionaliteit van een videoconferentie te verkrijgen binnen een Flutter-applicatie. In dit hoofdstuk worden eerst de beoordelingscriteria uitgelegd. Vervolgens worden de verschillende mogelijke implementatiemethodes kort uitgelegd met hun voor- en nadelen. Hier wordt niet in detail uitgelegd wat de verschillen zijn tussen de specifieke implementatiemethodes omdat dit niet relevant is. Als laatste wordt alles in een tabel voorgesteld om hier vervolgens een conclusie uit te kunnen trekken.

3.1.1 Beoordelingscriteria

Hier wordt meer uitleg gegeven over de drie beoordelingscriteria die specifiek voor dit onderzoek van toepassing zijn.

3.1.1.1 Aanpasbare gebruikersomgeving

Met aanpasbare gebruikersomgeving wordt bedoeld hoe makkelijk of moeilijk het is om de gebruikersomgeving aan te passen aan wat gewenst is. Indien de gebruikersomgeving meegeleverd wordt met de implementatie, wordt er ook rekening mee gehouden of de gebruikersomgeving automatisch aanpast aan het formaat van het scherm, ook wel *responsiveness* genoemd.

3.1.1.2 Implementatie

Hiermee wordt bedoeld hoe eenvoudig of ingewikkeld de methode kan worden geïmplementeerd in het bestaande project. Het kan bijvoorbeeld maar kwestie zijn van een aantal lijnen code toe te voegen of een ingewikkeld proces met meerdere verschillende onderdelen die geïmplementeerd moeten worden. Dit wordt beoordeeld ten opzichte van de methodes zelf, want eenvoudigheid van implementatie is relatief van persoon tot persoon.

3.1.1.3 Functionaliteiten

Het laatste criterium focust zich op de beschikbaarheid van alle functionaliteiten die nodig zijn voor het project. Het is mogelijk dat deze functionaliteiten reeds aanwezig zijn in een implementatiemethode of dat deze gecreëerd kunnen worden met behulp van de implementatiemethode. Beiden worden geaccepteerd, maar het is geen optie.

3.1.2 Verschillende methodes

Dit is een versimpeling van de verschillende methodes en is niet bedoeld als volledige verklaring van de methode. Het is enkel de bedoeling om het te verduidelijken in functie van de beoordelingscriteria.

3.1.2.1 Zelf creëren

De eerste mogelijkheid is de functionaliteit van een videoconferentie volledig op maat ontwikkelen. Dit is erg complex en hier komt veel bij kijken.

Het voordeel van deze methode is dat er nooit te veel of te weinig gecreëerd wordt, alles is op maat gemaakt. De ontwikkelaars hebben daarom de volledige controle over de gebruikersomgeving. Het nadeel hiervan is dat deze manier het meeste werk is van alle implementatiemethodes omdat alles van nul moet worden ontworpen en ontwikkeld.

3.1.2.2 API

Een andere manier om het probleem op te lossen is met een *application programming interface* (API). Web-API's maken interactie mogelijk tussen verschillende systemen met een set instructies die bepalen hoe de services en systemen moeten samenwerken.

Met deze methode wordt enkel de gebruikersomgeving volledig gecreëerd door de ontwikkelaar(s). De functionaliteit van een videoconferentie wordt aangeboden door de API. Deze API moet de ontwikkelaar vervolgens samengieten met de gebruikersomgeving tot één geheel.

In tegenstelling tot het zelf creëren is deze methode sneller omdat hier de functionaliteit niet meer zelf gecreëerd moet worden. Dit omdat deze verkregen wordt van de API van een externe dienst. Er is met deze methode volledige controle over hoe de gebruikersomgeving gecreëerd wordt. Een nadeel is dat er maar gedeeltelijke controle is over welke functionaliteiten er beschikbaar zijn. De functionaliteiten zitten vast aan de keuze van API. De keuze zal ook bepaald worden door het feit of er een API beschikbaar is die precies past bij de criteria van het project. Een ander nadeel bij een API is opnieuw dat het implementeren ervan veel tijd kan kosten.

3.1.2.3 SDK

Een Software Development Kit (SDK) is een development toolkit waar allerlei gereedschap in zit die gebruikt kan worden om een applicatie te ontwikkelen. Een SDK biedt veel meer dan een API, het biedt hulpprogramma's, bibliotheken, relevante documentatie, codevoorbeelden, processen en handleidingen waarmee ontwikkelaars de applicaties kunnen maken. [3]

Het belangrijkste voordeel bij deze implementatiemethode is dat dit relatief weinig werk kost om te implementeren omdat de meeste onderdelen reeds gegeven zijn door de SDK. Naast een vaste gebruikersomgeving kan het zijn dat een SDK ook de mogelijkheid biedt om een afgestemde gebruikersomgeving bij te leveren. Er is hier dus de mogelijkheid om zelf de gebruikersomgeving te creëren of die van de SDK te gebruiken. Een nadeel dat een SDK deelt met een API is dat het er maar net aan ligt welke functionaliteiten er nodig zijn en welke beschikbaar zijn in de SDK. Niet alle toolkits bieden dezelfde onderdelen aan. Het is mogelijk dat er te veel of te weinig beschikbaar is en als er een cruciaal onderdeel mist, moet er een andere SDK gezocht worden.

3.1.2.4 Externe website

De laatste methode die in dit onderzoek wordt besproken is een externe website. Hier vallen de meerderheid van de tools onder. Deze methode werkt door de gebruiker op een simpele link te laten klikken die de gebruiker naar een website stuurt. Deze website verzorgt vervolgens alle functionaliteit voor de applicatie. Zo zou de applicatie alleen een link moeten tonen waarop de gebruiker vervolgens kan klikken om dan samen te komen op een website en hier de conferentie te houden. De gebruiker kan vervolgens terugkeren naar de applicatie wanneer hij klaar is.

De implementatie van deze methode is de eenvoudigste optie, maar heeft als belangrijk nadeel dat de aanpassingsopties erg beperkt zijn. Er valt bij het gebruik van deze methode niet veel te veranderen aan de gebruikersomgeving of functionaliteiten. De ontwikkelaar heeft hier geen controle over.

3.1.3 Conclusie

Alle voor- en nadelen zijn in de volgende tabel samengevat. Om de tabel te begrijpen wordt eerst uitgelegd hoe de beoordeling wordt gevisualiseerd door middel van symbolen.

3.1.3.1 Visualisatie

Het plusteken (+) betekent dat het beoordelingscriterium positief is voor deze implementatiemethode terwijl het minteken (-) het tegenovergestelde betekent. Het plusminusteken (±) betekent dat het mogelijk is dat alle benodigde functionaliteiten aanwezig zijn maar dat er meer onderzoek nodig is voordat er een specifieke toepassing gekozen kan worden.

Beoordelingscriteria	Zelf	API	SDK	Web
Aanpasbare gebruikersomgeving	+	+	+	-
Implementatie	-	-	+	+
Functionaliteiten	+	±	±	-

Tabel 1. Vergelijkingstabel verschillende methodes

3.1.3.2 Samenvatting

De eerste methode, waar alles handgemaakt wordt, is een mogelijkheid, maar valt buiten de scope van dit onderzoek en zou tevens te veel tijd en resources kosten. Indien dit onderzoek zich niet zou focussen op het zo eenvoudig mogelijk implementeren van de functionaliteit van een videoconferentie zou dit wel een mogelijke oplossing zijn.

De API zou ook een mogelijke keuze kunnen zijn maar in functie van deze specifieke beoordelingscriteria lijkt deze sterk op de eerste methode. Deze valt daarom in dezelfde categorie als de eerste methode. Het zou een mogelijkheid zijn, maar in het project ligt de focus op het vinden van een zo eenvoudig mogelijke implementatie en dit is niet het geval met deze methode.

De SDK biedt het beste van beide werelden: een aanpasbare gebruikersomgeving en een eenvoudige implementatie. Er moet echter wel nog onderzocht worden welke specifieke SDK gebruikt zal worden met deze implementatiemethode. Een ander nadeel aan deze methode is dat de SDK meer levert dan nodig is, met alle mogelijke gevolgen van dien.

Web is een eenvoudig te implementeren methode maar er is geen mogelijkheid tot een afgestemde gebruikersomgeving. Net zoals bij de API- en SDK-methodes zou hier eerst onderzocht moeten worden welke specifieke toepassing van externe websites er gebruikt zal worden voordat er verder aan de slag wordt gegaan.

3.1.3.3 Conclusie

Uit deze tabel kan worden geconcludeerd dat een **SDK de beste oplossing is** omdat deze methode de voordelen van de API- en webimplementatiemethode samenvoegt. Er bestaan met deze methode nog altijd enkele nadelen, want het is mogelijk dat er onderdelen in de SDK zitten die niet nodig zijn voor dit specifieke project en die dus het eindproduct groter maken dan als het op maat gemaakt zou zijn. Een ander nadeel is dat er nog onderzocht moet worden welke specifieke SDK er gekozen wordt.

De implementatiemethode waarbij een API wordt gebruikt en de methode waarbij het volledig zelf wordt gecreëerd, kosten relatief te veel tijd om te implementeren en worden daarom uitgesloten in dit onderzoek. Een SDK is hierdoor een goede middenweg. Vanaf dit punt wordt er in dit bestand verondersteld dat er gekozen wordt voor een tool die een SDK aanbiedt.

3.2 Ondersteuning

Welke mogelijke tools zijn er beschikbaar die zowel web ondersteunen als Flutter?

Om deze vraag te beantwoorden moet er eerst onderzocht worden welke tools er beschikbaar zijn om vervolgens de tools uit te sluiten die niet de juiste ondersteuning bieden. Dit wordt gedaan door een algemene lijst op te stellen van tools die de functionaliteit bieden van een videoconferentie. Vervolgens worden de tools in meerdere fases gefilterd om het scala aan mogelijkheden te verkleinen en tot slot blijven de beste tools over.

3.2.1 Tools met SDK's

Zoals er besloten is in het vorige hoofdstuk moet de tool SDK's aanbieden in plaats van een van de andere implementatiemethodes. De volgende lijst is opgemaakt uit alle onderzochte tools die de functionaliteit bieden van een videoconferentie. In de lijst wordt de tool in het groen gemarkeerd indien deze een implementatie biedt aan de hand van SDK's. Diegene die niet gemarkeerd zijn vallen onder een van de andere implementatiemethodes die besproken zijn in subsectie 3.1.2.

#1 Chat Avenue	Fuze Meeting	MegaMeeting	TelyLabs
Adobe Connect	GatherPlace	Microsoft Office Live Meeting	TenHands
Agora	Glance	Movicha	Tinychat
Anymeeting	GoToMeeting	Netviewer Meet	TokBox
AppearIn	HearMe	Omnijoin	Twilio
Avistar	HP SkyRoom	Paltalk	Userplane
Biscotti	IBM SameTime	Persony	Veamea
Bistri	iChat	Polycom PVX	Veodia
Blackboard Collaborate	iLinc	Pubnub	VideoMost
Bluejeans	iMeet	Quickblox	Vidyo io
BoostCam	InstantPresenter	Rabbit	Voxeet
Brosix	InterCall	Radvision SCOPIA	Voximplant
Cisco Unified Meeting Place	iocom	Respoke	VSee
Citrix Online	ISL Online	Rounds	VTEL
CometChat	IspQ	Saba Centra	Vyew
Conference Group	iVisit	ShowKit	Webcamconsult
DigiGone	Jitsi	SightCall	Webex
Faceflow	Kandy	Sinch	WeChat
Facetime	Line	Skype	WorldVuer
Firefox Hello	Logitech - Lifesize	Stickam	YuuGuu
FlashMeeting	Meet.fm	SwyMe	Zoom
Forge	MeetingBurner	Tango	
Fring	Meetings.io	Teamviewer	

Tabel 2. Potentiële tools

In deze eerste fase van filteren zijn er uit de 90 tools die onderzocht zijn in totaal 13 die ondersteuning bieden door middel van SDK's. Vanuit deze 13 tools kan verder gefilterd worden.

3.2.2 Gewenste SDK's

In de tweede fase wordt er gefilterd op de tools die ook wel degelijk de correcte SDK's bieden die gewenst zijn voor het project. Om dit te kunnen filteren moet er eerst bepaald worden welke SDK's nodig zijn. In het project worden twee apps gemaakt, een webapplicatie en een mobiele applicatie. Zoals eerder verklaard is, moet de tool in beide apps geïmplementeerd worden. Om de tool te kunnen implementeren in deze apps moet de tool ondersteuning bieden voor de gebruikte frameworks. In dit hoofdstuk wordt eerst uitleg gegeven over de verschillende manieren waarop mobiele en webapplicaties ondersteund kunnen worden. Vervolgens wordt er weergegeven in een tabel welke tools er overblijven na het filteren op gewenste SDK's.

3.2.2.1 Webondersteuning

De webapplicatie wordt gemaakt met behulp van het React-framework en de tool moet dan ook samenwerken met dit framework. Er zijn twee mogelijkheden om de tool te implementeren in de webapplicatie:

1. de tool biedt een specifieke React-SDK aan;
2. de tool biedt een algemene Web-SDK aan en deze wordt met de hand geïntegreerd in React.

Omdat dit eindwerk zich focust op het Flutter-framework is het van belang dat de scope zich niet uitbreidt tot andere frameworks. Indien er voor de tweede optie wordt gekozen, zou het integreren in het React-framework buiten de scope vallen. Daarom wordt er gekozen voor optie één. Indien dit uiteindelijk leidt tot een beperkt aantal tools, bestaat er een mogelijkheid om toch voor de tweede mogelijkheid te kiezen om zo het scala aan potentiële tools nog te verbreden. Voor nu wordt er aangehouden dat React een van de gewenste SDK's is voor de webapplicatie.

3.2.2.2 Mobiele ondersteuning

Nu er bepaald is welke ondersteuning de tool moet hebben voor de webapplicatie, moet hetzelfde gebeuren voor de mobiele applicatie. Om de tool te implementeren in de mobiele applicatie zijn er opnieuw twee mogelijkheden:

1. de tool biedt een specifieke Flutter-SDK aan;
2. de tool biedt een Android- en iOS-SDK aan en deze worden apart geïmplementeerd in Flutter.

Omdat Flutter nog een recent framework is, is het aantal tools die ondersteuning bieden voor Flutter schaars. Hier kan echter omheen worden gewerkt door iOS en Android apart te implementeren in Flutter. Hoe dit precies in zijn werk gaat, wordt verduidelijkt in het onderdeel stageverslag. Voor de mobiele applicatie worden beide mogelijkheden aanvaard. De gewenste SDK's kunnen ofwel de aparte SDK voor iOS en Android zijn of een SDK voor Flutter specifiek.

3.2.2.3 Gewenste ondersteuning

In de onderstaande tabel wordt weergegeven welke gewenste SDK's de 13 overgebleven tools ter beschikking stellen. In de tabel worden enkel de SDK's weergegeven die relevant zijn tot het project en dit moet niet gezien worden als een totaaloverzicht van SDK's die deze tools ter beschikking stellen.

Nr.	Tool Naam	SDK				
		Web		Mobile		
		Web	React	Android	iOS	Flutter
1	Agora	X	X	X	X	X
2	CometChat	X	X	X	X	-
3	Pubnub	X	X	X	X	-
4	Quickblox	X	-	X	X	-
5	Sinch	X	-	X	X	-
6	Skype	X	-	X	X	-
7	TokBox	X	X	X	X	-
8	Vidyo	X	X	X	X	-
9	Voxeet	X	X	X	X	-
10	Voximplant	X	-	X	X	-
11	VSee	X	-	X	X	-
12	Webex	X	X	X	X	-
13	Zoom	X	-	X	X	-

Tabel 3. Relevante SDK's

Zoals te zien in Tabel 3 is er voor de mobiele applicatie maar één tool die een Flutter-SDK aanbiedt en voor de webapplicatie zijn er 7 tools die een React-SDK aanbieden. Er blijven nog 7 tools over als de tools die React niet ondersteunen er uit worden gefilterd. De mogelijke tools die beschikbaar zijn die zowel React als een vorm van mobiel ondersteunen zijn: Agora, CometChat, Pubnub, TokBox, Vidyo, Voxeet en Webex. Deze worden in het volgende onderdeel individueel behandeld.

3.3 Proof of concept mogelijkheden

Van welke tools kan er het beste een *proof of concept* gemaakt worden als er rekening wordt gehouden met de vereisten?

Hier worden de 7 tools die overgebleven zijn individueel besproken op basis van de vereisten. Eerst worden de vereisten overlopen waar rekening mee gehouden wordt en vervolgens worden de tools een voor een overlopen om te controleren op alle vereisten. Indien een tool een vereiste niet bezit, wordt de tool afgekeurd. Een tool afkeuren betekent dat er geen *proof of concept* van de tool wordt gemaakt en dat er verder geen onderzoek naar wordt gedaan omdat de tool niet geschikt is. Ten slotte blijven de resterende tools over waarvan er een *proof of concept* gemaakt wordt.

3.3.1 Vereisten

3.3.1.1 Scope breedte

Eén van de vereisten is de breedte van de scope van de SDK. Indien de SDK functies bezit die voor het project overbodig zijn, is het mogelijk om op basis van deze reden de tool af te keuren. Indien de SDK de minimale functies niet bezit, wordt de tool ook afgekeurd.

De minimale functies waarover de tool moet beschikken zijn, de mogelijkheid om:

- de microfoon aan en uit te zetten;
- de video aan en uit te zetten;
- de videoconferentie te verlaten.

Hiernaast zijn er ook een aantal optionele functies, dit zijn *nice-to-have* functies en ze zijn een pluspunt indien de tool er een of meerdere beschikbaar stelt, maar het is geen nadeel indien de tool er geen heeft. Deze zijn:

- de mogelijkheid om een videoconferentie op te nemen en later terug te bekijken;
- de mogelijkheid om de videoresolutie aan te passen;
- de mogelijkheid om bestanden te delen;
- de mogelijkheid om exclusief tekstueel te communiceren;
- een wachtkamer waarin de gebruiker zich bevindt voordat er anderen aanwezig zijn.

Indien de tool wordt afgekeurd op een andere vereiste, wordt de scopebreedte niet overlopen omdat dit overbodig is.

3.3.1.2 Serverhosting

Een andere vereiste vanuit Yunify is dat de tool de backendserverhosting moet aanbieden. Dit heeft te maken met het feit dat de tool zo eenvoudig mogelijk geïmplementeerd moet worden (zie 3.1.1). De tool moet hierom zijn eigen servers voorzien zodat deze niet zelf door Yunify verzorgd moeten worden. Indien de tool dit niet aanbiedt, wordt hij afgekeurd. Bij het evalueren van de tools zal dit aspect enkel vermeld worden indien de tool geen serverhosting ondersteunt.

3.3.1.3 Documentatie

Als laatste wordt er ook rekening gehouden met de kwaliteit van de documentatie. Als er te weinig tot geen documentatie is, is het moeilijker om een *proof of concept* en later een applicatie te ontwikkelen. Alleen op basis hiervan zal een tool niet afgekeurd worden, maar het telt zeker mee in het totaalbeeld.

3.3.2 Agora io

Agora is, zoals te zien in Tabel 3, de enige tool die op het moment van dit eindwerk een directe ondersteuning biedt voor Flutter. Hierdoor moet er geen aparte implementatie gemaakt worden voor iOS en Android. Dit betekent dat de implementatie van deze tool het eenvoudigste is van alle tools. Qua scope bevat deze tool alle minimale en een aantal optionele vereisten.

3.3.3 CometChat

Deze tool en alle volgende tools bieden geen SDK aan voor het Flutter-framework zoals wordt weergegeven in Tabel 3. Met deze tool is het mogelijk om een videoconferentie op te stellen maar dit is niet de focus van de tool. De focus van deze tool is een chat, met als bijkomende functie een videoconferentie. Omdat er geen chat nodig is in de applicatie van Yunify en de tool geen Flutter-SDK aanbiedt, valt deze tool af.

3.3.4 Pubnub

Pubnub biedt geen backendserverhosting aan. [4] Omdat Yunify een tool zoekt die de serverhosting ook verzorgt, valt deze tool op basis van dit feit af.

3.3.5 TokBox

TokBox geeft aan dat er twee hoofddelen geïmplementeerd moeten worden om gebruik te maken van hun tool, één daarvan is een server. [5] Deze tool wordt afgekeurd omdat er geen serverhosting wordt aangeboden door de tool zelf.

3.3.6 Vidyo io

De scope van Vidyo ziet er op eerste zicht te klein uit, maar na dieper onderzoek ontbreken er geen kritieke onderdelen en komt de scope goed overeen met de vereisten van Yunify. De tool biedt ook een backendhostingservice aan. [6]

3.3.7 Voxeet

De scope van de SDK is goed want de tool bezit alle minimale vereisten en ook een aantal optionele vereisten. De ontwikkelaarsdocumentatie is minimaal en dit is wellicht een probleem bij het ontwikkelen van de *proof of concepts* voor deze tool. Omdat de scope goed is en een backend server biedt, wordt deze tool niet afgekeurd op de slechte documentatie.

3.3.8 Webex

Webex is op dit moment aan rebranding bezig: wat vroeger de naam *Cisco spark* had, is nu *Webex teams* geworden. [7] Het gevolg hiervan is dat de documentatie verspreid is over verschillende domeinen en dit zorgt voor verwarring. De rebranding zorgt er ook voor dat het niet duidelijk is wat precies bij het product hoort en wat niet. Het is wel duidelijk dat de scope van dit product breder is dan nodig. Het biedt veel meer functies aan dan alleen een videoconference. Aangezien de documentatie niet duidelijk is en de scope veel te breed is, valt deze tool af.

3.3.9 Conclusie

CometChat en Webex hebben een andere scope als die wat Yunify nodig heeft en Pubnub en TokBox bieden geen backend server aan. Hierdoor blijven Agora, Voxeet en Vidyo over om *proof of concepts* van te maken om verder te evalueren welke tool het meest geschikt is.

4. Proof of Concepts

Met welke tool kan de functionaliteit van een videoconferentie het beste geïmplementeerd worden in een Flutter-applicatie?

In dit hoofdstuk wordt getracht een *proof of concept* te maken van Agora, Voxeet en Vidyo om de tools verder te controleren op welke het meest geschikt is om een videoconference mee te houden. In dit hoofdstuk wordt eerst de scope van de *proof of concepts* en de verschillende beoordelingsaspecten verklaard, hierna worden de drie verschillende POC vergeleken in een vergelijkingsmatrix op basis van deze beoordelingsaspecten en als laatste wordt er een conclusie getrokken.

4.1 Scope proof of concept

Om een *proof of concept* te maken moet er eerst gedefinieerd worden welke functies er minimaal in moeten komen om deze vervolgens ook te kunnen vergelijken. De minimale functies waarover een POC moet beschikken zijn opgedeeld in twee schermen. In de volgende twee subsecties worden de schermen en functies hiervan verder verduidelijkt.

4.1.1 Initialisatiescherm

Het eerste scherm waar de gebruiker in de POC in terecht komt is een simpel scherm met twee tekstvelden en een knop. In het eerste tekstveld wordt de naam van de wachtkamer ingevuld en in het tweede tekstveld wordt de naam van de gebruiker ingevuld. Als er op de knop wordt gedrukt wordt de gebruiker naar het wachtkamerscherm verzonden.

4.1.2 Wachtkamerscherm

Het tweede scherm wordt voorgesteld als een wachtkamer waar de gebruiker in wordt geplaatst indien zich er nog geen ander gebruikers in dezelfde wachtkamer bevinden. De gebruiker kan hier in een klein scherm zijn eigen representatie zien zoals de overige gebruikers hem zouden zien indien ze ook in dezelfde wachtkamer zouden zitten. Indien twee gebruikers dezelfde naam van een wachtkamer invoeren, komen deze in dezelfde wachtkamer terecht. Vanaf het punt dat er twee of meer gebruikers zijn, zal er automatisch een videoconference plaatsvinden in dit scherm. In deze videoconference kan de gebruiker nog altijd zichzelf in een klein scherm zien en de overige gebruikers worden gelijkmatig verdeeld over het scherm.

Er zijn in dit scherm drie knoppen beschikbaar voor de gebruiker. Er is de mogelijkheid om het geluid te dempen, zijn video uit- en aan te zetten en als laatste kan de gebruiker de videoconference verlaten. Indien de gebruiker op de knop drukt van het geluid, wordt enkel nog zijn video verzonden naar de andere gebruikers in de videoconference. Eveneens gebeurt hier het gelijkwaardige met de knop van de video, indien de gebruiker hierop drukt wordt enkel nog zijn geluid verzonden. Indien de gebruiker drukt op de verlaat knop dan komt de gebruiker terug in het initialisatiescherm en zullen ook de ander gebruikers de gebruiker niet meer zien in de videoconference.

4.2 Beoordelingsaspecten

Bij het vergelijken wordt er rekening gehouden met bepaalde aspecten die beoordelingsaspecten worden genoemd. Om de verschillende POC tegen elkaar op te staven, worden hier eerst de beoordelingsaspecten verklaard. Ook wordt er toegelicht hoe deze aspecten getest of beoordeeld worden.

4.2.1 Documentatie

Onder dit aspect vallen drie sub-aspecten die behoren tot het onderwerp documentatie. Deze worden kort toegelicht.

4.2.1.1 Openbare documentatie

Met dit onderdeel wordt bedoeld hoe uitgebreid en relevant de openbare documentatie is van de tool die zij beschikbaar stellen op hun website.

4.2.1.2 Code commentaar

Het is belangrijk om in de code ook in commentaar er bij te verklaren wat de code precies doet omdat dit het ontwikkelen van een applicatie kan versnellen. De term "Goede code is zelfdocumenterend." wordt wel eens vermeld, maar dit is niet altijd waar en het kan nog altijd beter zijn om de code te becommentariëren ondanks hoe goed de code blijkt te zijn. [8]

4.2.1.3 Voorbeelden

Met voorbeelden wordt bedoeld of de tool voorbeeldprojecten beschikbaar stelt die het implementeren van de tool vergemakkelijken door voorbeelden van hun eigen tool ten toon te stellen.

4.2.2 Implementatie

Bij dit aspect wordt er besproken hoe ingewikkeld de tool is om te implementeren. De documentatie heeft hier invloed op, maar dit wordt apart beoordeeld omdat implementatie een belangrijk deel is van het maken van een POC en weegt daarom meer door dan een sub-aspect van documentatie. Implementatie wordt opgedeeld in twee sub-aspecten.

4.2.2.1 Klantenondersteuning

Hier wordt de kwaliteit van de klantenondersteuning beschreven. Hoe snel wordt er geantwoord op problemen, hoe is de kwaliteit van de antwoorden, worden er toepasselijke antwoorden gegeven? Dit zijn de vragen die beantwoord worden met dit sub-aspect.

4.2.2.2 Algemeen

Dit is eerder aangehaald in 3.1.1.2 maar daar wordt de nadruk gelegd op de specifieke implementatie van een tool in plaats van de algemene eenvoudigheid van een implementatiemethode.

4.2.3 Netwerkprestaties

Om de *proof of concept* te beoordelen is het ook belangrijk om te testen hoe de tool reageert op een lagere netwerksnelheid of zelfs het netwerk dat compleet wegvalt midden in het gesprek. Dit is vooral belangrijk bij het bekijken van een videoconference zoals hier het geval is. De *proof of concepts* worden hierop getest door middel van de internetsnelheid te limiteren en te kijken hoe de tool hier op reageert.

4.2.4 Overige beschikbare functies

In onderdeel 3.3.1.1 werd reeds aangehaald wat de mogelijke optionele beschikbare functies zijn maar hier werd de focus gelegd op niet optionele functies waarover de tool kan beschikken. Met dit aspect wordt de focus gelegd op de mogelijkheid dat Yunify later wil uitbreiden met een andere functie. Deze functie zou dan ook beschikbaar moeten zijn in de tool. Indien deze functie niet beschikbaar is, is het niet mogelijk om met de gekozen tool verder te gaan en zou er op dat moment alsnog een andere tool gekozen moeten worden.

De functies die in aanmerking komen voor dit beoordelingsaspect zijn:

- de mogelijkheid om bestanden te delen;
- de mogelijkheid om een videoconference op te nemen en later terug te kijken;
- de mogelijkheid om scherm te delen zodat anderen het scherm zien;
- de mogelijkheid om exclusief tekstueel te communiceren;
- de mogelijkheid om de videoresolutie aan te passen;
- een wachtkamer waar de gebruiker zich in bevindt voordat er iemand anders aanwezig is.

Om hier een overzicht van te krijgen, worden ze in een tabel voorgesteld bij de betrokken tool en hierbij worden ze respectievelijk afgekort als:

- bestanden delen;
- opnemen;
- scherm delen;
- tekst chat;
- videoresolutie;
- wachtkamer.

Bij elke tool wordt er eerst een overzicht getoond met daarin aangekruist welke functies de tool bezit en vervolgens wordt de redenering erachter uitgelegd.

4.2.5 Prijs

Er zijn verschillende prijsmodellen die gebruikt worden. Zo kan er betaald worden per klant die de tool gebruikt, per minuut die gebruikt wordt en/of een vaste prijs per tijdseenheid, bijvoorbeeld per maand of jaar.

Niet alle bedrijven maken hun prijsmodel publiek en in deze gevallen ligt de prijs aan een onderhandeling die plaatsvindt nadat het aantal gebruikers bekend is. Voor Atlas zijn er nog geen schattingen bekend dus indien dit het geval is, kan er niet beoordeeld worden.

De prijs is voor Yunify uiteindelijk een doorslaggevende factor. In dit onderzoek zal er echter geen doorslaggevende factor aan gegeven worden omdat er nog niet genoeg informatie beschikbaar is om een goede schatting te maken. De prijzen van de verschillende tools kunnen wel vergeleken worden ten opzichte van elkaar indien ze vergelijkbare prijsmodellen voeren.

4.3 Agora

Hier wordt dieper ingegaan op de aspecten die gedefinieerd zijn. In dit deel wordt de POC van Agora besproken op de aspecten die hiervoor aangehaald zijn.

4.3.1 Documentatie

4.3.1.1 Openbare documentatie

Deze tool beschikt over een uitgebreide documentatie die alle relevante functies overzichtelijk groepeert per onderdeel. Bij elke functie wordt meer uitleg gegeven over het gebruik van de functie, de mogelijke parameters, errorcodes en retourwaardes. [9]

4.3.1.2 Code commentaar

Er wordt in de code van de SDK bij elke functie eerst kort toegelicht waar de functie precies voor dient en daarna in detail extra uitleg gegeven hoe deze gebruikt dient te worden, met daarbij alle nodige informatie vermeld. De methodes worden ook voor de duidelijkheid onderverdeeld in kern-, media-, statistische en diverse methodes. [10]

4.3.1.3 Voorbeelden

Agora biedt voorbeeldprojecten aan voor zowel Android-, iOS- als Flutter-SDK implementaties. [11] Er worden verschillende mogelijkheden aangeboden om snel van start te gaan en er zijn instructies aanwezig om het project eenvoudig te integreren met hun SDK. In de projecten zijn niet alle functies die ze aanbieden aanwezig. Het zijn minimale voorbeelden met enkel de basis functionaliteiten van een videoconferentie zonder alle extra functies die ze aanbieden. Voor deze extra functies kan er wel in de documentatie gekeken worden maar in de voorbeeldprojecten zijn er geen voorbeelden van.

4.3.2 Implementatie

4.3.2.1 Klantenondersteuning

Om met Agora contact te krijgen moet er eerst een account geregistreerd worden. Als er eenmaal contact is gelegd, is er binnen één werkdag een uitgebreid antwoord dat goed geformuleerd is en goed de vraag beantwoordt.

4.3.2.2 Algemeen

Omdat er directe ondersteuning is voor Flutter met deze tool, is het ontwikkelen van een *proof of concept* van Agora relatief snel gedaan. De uitgebreide documentatie en de voorbeeld projecten maken de implementatie van de SDK in een applicatie nog eenvoudiger.

4.3.3 Netwerkprestaties

Bij het uitvallen van een internetconnectie probeert de tool automatisch terug te verbinden binnen een vastgestelde tijd. Indien de gebruiker na deze tijd nog niet terug is verbonden, worden de andere gebruikers op de hoogte gebracht dat de verbinding is verbroken. Indien de gebruiker wel terug een verbinding kan krijgen binnen de tijd, wordt deze door de tool automatisch terug in de conference geplaatst.

Indien de verbinding van de gebruiker vermindert terwijl hij in een gesprek is, wordt de videokwaliteit automatisch aangepast naar een gepast formaat zodat de gebruiker zonder problemen het gesprek kan voortzetten.

Omdat het beoordelen van de andere tools op dit aspect niet mogelijk is, heeft het weinig zin om hier op het moment uitgebreid op te testen.

4.3.4 Overige beschikbare functies

Overige functie	Agora
Bestanden delen	-
Opnemen	X
Schermdelen	X
Tekst chat	-
Videoresolutie	X
Wachtkamer	-

Tabel 4. Overige functies Agora

Bij het opnemen van een videogesprek biedt Agora twee opties aan: "On-premise" en "cloud" opnamen. [12] "On-premise" houdt in dat de infrastructuur lokaal wordt opgezet, dit is niet wat Yunify zoekt en het is daarom geen optie om de lokale route te kiezen. De tweede optie is daarentegen wel mogelijk. Vergeleken met lokale serveropname vereist de "cloud" opname niet de inzet van een server, waardoor de druk op de ontwikkeling wordt vermindert. Omdat dit wel binnen het plaatje valt, wordt de "cloud" opname geaccepteerd als een correcte oplossing.

Zoals te zien is in de tabel hierboven is de mogelijkheid om een scherm te delen beschikbaar bij Agora en de videoresolutie kan worden ingesteld door middel van profielen. [13][14] Bestanden delen, een wachtkamer en een tekst chat is mogelijk om zelf erbij te ontwikkelen maar Agora biedt hier geen directe ondersteuning voor.

4.3.5 Prijs

Agora biedt de eerste 10 000 communicatie-minuten gratis aan. Hierna wordt er \$3.99 per 1000 minuten aan video gehanteerd. [15] Deze communicatie-minuten worden berekend door middel van gebruikte minuten, het aantal gebruikers en de gebruikte resolutie. [16] De formule hiervoor is eenheidsprijs videoformaat x Totaal aantal communicatie-minuten.

4.4 Voxeet

Hier wordt de POC van Voxeet besproken aan de hand van de aspecten die hiervoor gedefinieerd zijn.

4.4.1 Documentatie

4.4.1.1 Openbare documentatie

Deze tool heeft een acceptabele documentatie die alle functies in een lijst uitlegt. Bij de meeste functies wordt er een korte toelichting gegeven waar in staat wat de functie doet. Ook wordt hierbij een voorbeeld van het gebruik van de functie, de mogelijke parameters en retourwaardes getoond. [17]

4.4.1.2 Code commentaar

Er is geen commentaar beschikbaar in zowel de code van de SDK, als in de code van de voorbeeldprojecten. [18]

4.4.1.3 Voorbeelden

Ze bieden geen voorbeeld aan voor Flutter maar wel voor iOS en Android. [19] De Android en iOS voorbeeldprojecten zijn uitgebreid en tonen een meerderheid van de functies die ze aanbieden. De voorbeelden die er zijn, zijn eenvoudig op te zetten omdat er duidelijke aanwijzingen aanwezig zijn.

4.4.2 Implementatie

4.4.2.1 Klantenondersteuning

Voxeet is te bereiken via email, er is echter niet duidelijk naar welke email vragen gericht moeten worden. Als er een mail is verstuurd wordt er binnen één werkdag een minimaal antwoord teruggestuurd. In dit antwoord worden de vragen die gesteld waren niet beantwoord en er is ook geen belangstelling om de vragen verder te onderzoeken om er later op terug te komen.

Naast email is Voxeet ook te bereiken via issues op hun Github pagina. Ze zijn de laatste maand actief geweest aan de publieke *repositories* van de SDK's en deze worden goed onderhouden.

4.4.2.2 Algemeen

De documentatie en voorbeeldprojecten waren niet genoeg om de Android- en iOS-SDK's te implementeren in de Flutter applicatie. Omdat er geen commentaar in de code stond is het ingewikkeld om de logica achter de diepere code te achterhalen. Het feit dat er geen medewerking was vanuit Voxeet maakt het implementeren ervan ook niet makkelijker. Voor een volledig rapport over het verloop van de implementatie van deze tool, kan er verwezen worden naar onderdeel 4.5.2.2 waar een nagenoeg identieke situatie in detail is uitgeschreven.

4.4.3 Netwerkprestaties

Omdat het implementeren van de Android- en iOS-SDK in de Flutter applicatie niet gelukt is en de documentatie en klantondersteuning niet goed is, kan dit onderdeel niet beoordeeld worden voor deze tool.

4.4.4 Overige beschikbare functies

Overige functie	Voxeet
Bestanden delen	-
Opnemen	X
Schermdelen	X
Tekst chat	-
Videoresolutie	-
Wachtkamer	-

Tabel 5. Overige functies Voxeet

Zoals geïllustreerd is in de tabel hierboven is het bij Voxeet mogelijk om een opname te maken en om een scherm te delen. Hieronder wordt meer uitleg gegeven over elke functie.

Het is niet mogelijk om bestanden te delen met andere leden van de videoconferentie. Het is wel mogelijk om een bestand te presenteren aan de andere leden, maar dit behoort niet tot de functies die in aanmerking komen voor dit beoordelingsaspect. [17]

Voxeet heeft bij opname, net zoals Agora, twee opties. Een “WebRTC” en een “mp4” modus zoals zij het noemen. [20] De “WebRTC” optie slaat de opnames op in de servers van Voxeet met de mogelijkheid om deze op te vragen en terug te kijken wanneer gewenst. Dit is wat Yunify zoekt.

Schermdelen biedt Voxeet enkel aan voor de web variant van hun SDK. Er kan dus op mobiel een scherm van web worden weergegeven maar niet andersom. Dit is echter goed genoeg voor de applicatie die Yunify wil ontwikkelen.

De mogelijkheid om een tekst chat te implementeren wordt niet vanuit Voxeet verzorgd en er is geen mogelijkheid om de videoresolutie zelf in te stellen.

De wachtkamer is, net zoals bij Agora, zelf te ontwikkelen maar Voxeet biedt hier ook geen directe ondersteuning voor.

4.4.5 Prijs

Voxeet werkt met een vaste prijs van \$0.00375 per *subscribed minute*. Deze minuten worden uitgerekend als $N \times (N-1) \times \text{lengte van het conference}$, waar N het aantal deelnemers betekent. [21]

4.5 Vidyo

Hieronder worden de aspecten besproken maar nu gaat het zijn voor Vidyo.

4.5.1 Documentatie

4.5.1.1 Openbare documentatie

Deze tool beschikt over een minimale documentatie die niet elke functie overloopt. Van de functies die wel vermeld worden, wordt er enkel een korte uitleg gegeven over hoe de functie gebruikt dient te worden. Er wordt wel bij elke vermelde functie een voorbeeld implementatie gegeven in elke programmeertaal die ze ondersteunen. Er worden geen mogelijke parameters, errorcodes of retourwaardes aangehaald. [6]

4.5.1.2 Code commentaar

Er is zowel in de SDK, als in de code van de voorbeeldprojecten, nergens commentaar bijgevoegd.

4.5.1.3 Voorbeelden

Vidyo biedt voorbeeld projecten aan voor Android en iOS, maar aangezien ze geen Flutter-SDK bieden, hebben ze geen voorbeeld van Flutter. De voorbeelden die ze beschikbaar stellen, bezitten wel alle functies die ze beschikbaar hebben. Het opzetten hiervoor is overal duidelijk uitgelegd, maar er wordt benadrukt dat het beter is om hun support team te contacteren indien er problemen zijn. [22] Met hulp van dit team zijn de voorbeelden eenvoudig op te zetten.

4.5.2 Implementatie

4.5.2.1 Klantenondersteuning

Om vragen te stellen aan Vidyo is er een duidelijk e-mailadres beschikbaar dat ook gelinkt is met hun *ticketing-systeem*. De vragen worden zelfs op niet-werkdagen binnen één dag erkend. Ze worden beantwoord en waar niet direct antwoord mogelijk is, worden de vragen doorverwezen naar de relevante professionals om hier alsnog een antwoord te verkrijgen.

4.5.2.2 Algemeen

Ondanks dat de support van Vidyo heeft proberen te helpen en het feit dat ze een uitgebreide documentatie hadden, is de implementatie van een Android- en iOS-SDK in een Flutter applicatie niet gelukt. Er zijn een aantal manieren geprobeerd om de SDK's te implementeren in de Flutter-applicatie.

Als eerste is er geprobeerd om de SDK's direct te implementeren door middel van de *method-channels* die het Flutter framework biedt. Deze bieden de mogelijkheid om te communiceren met de directe implementaties van de specifieke platformen, in dit geval Android en iOS. Hierdoor werd de code rommelig. Dit omdat de code met een minimale factor van drie werd vergroot, omdat er voor elke originele lijn code nu drie lijnen code in de plaats komen. Deze drie waren: de lijn die verwijst naar de platform specifieke code en de code voor beide implementaties in hun eigen platform. Hierdoor werd de code ingewikkeld en onleesbaar.

Hierdoor is er onderzocht welke mogelijkheden er zijn om dit probleem op te lossen. Er zijn twee relevante mogelijkheden die dit probleem op lossen: een *plugin* en een *wrapper*.

Een *plugin* is een abstractie van de communicatie laag tussen Flutter en zijn platform specifieke implementaties. Deze *plugins* worden meestal gepubliceerd zodat andere mensen met hetzelfde probleem deze zelfde *plugin* kunnen gebruiken om het probleem op te lossen. Omdat Flutter nog een jong platform is, zijn er op dit moment nog weinig publieke *plugins* beschikbaar als er wordt vergeleken met vergelijkbare technologieën die het zelfde doel hebben als Flutter. Er is hierdoor geen *plugin* beschikbaar die de Vidyo-SDK's implementeert. Hierna is er onderzocht hoe dit het beste zelf gecreëerd kan worden.

Bij een *wrapper* wordt er een extra laag toegevoegd over de twee platform implementaties die de implementaties omsluit ofwel 'wrapped'. Het doel van deze extra laag is de communicatie tussen de Flutter-applicatie en de platform implementaties eenvoudiger te maken. Dit zorgt er op zich weer voor dat de code duidelijker wordt. Een verschil met een *plugin* is dat een *wrapper* niet abstract is omdat er altijd een directe implementatie voor de toepassing van de *wrapper* gemaakt wordt. In dit geval is de toepassing, de communicatie tussen de applicatie en de Vidyo-SDK's.

Na het proberen van het creëren van een handgemaakte *plugin* of *wrapper* is het duidelijk geworden dat deze beiden op hetzelfde probleem stuiten. Er is een bepaald object nodig dat niet te verkrijgen is, omdat de platform implementaties achterliggend onderdeel zijn van het Flutter-framework. Dit zorgt ervoor dat de informatie die de SDK's nodig hebben niet te verkrijgen is in een Flutter applicatie. Het is mogelijk dat Flutter in latere stadia dit probleem gaat oplossen maar op dit moment is dit probleem niet op te lossen.

Een mogelijke oplossing is dieper onderzoek doen naar de werking van de *Flutter-engine* en hier aanpassingen aan maken, maar dit valt buiten de scope van het project en zou een onderzoek op zich kunnen zijn.

Een andere mogelijke oplossing is om te vragen aan het bedrijf, in dit geval Vidyo, om de SDK aan te passen zodat het werkt op een manier die wel samenwerkt met Flutter.

4.5.3 Netwerkprestaties

Omdat het implementeren van de Android- en iOS-SDK in de Flutter applicatie niet gelukt is door de hierboven beschreven redenen, kan deze tool niet beoordeeld worden op netwerkprestaties.

4.5.4 Overige beschikbare functies

Overige functie	Vidyo
Bestanden delen	-
Opnemen	-
Schermdelen	X
Tekst chat	X
Videoresolutie	X
Wachtkamer	-

Tabel 6. Overige functies Vidyo

Bestanden delen is niet mogelijk met de SDK van Vidyo.

Voor het opnemen van het videogesprek bieden ze een optie aan die werkt met Docker. [23] Met deze optie zou het nodig zijn om deze Docker integratie op de servers van Yunify te installeren. Alhoewel dit relatief eenvoudig zou zijn, gaat dit in tegen een van de eisen dat de opnames worden opgeslagen op de servers van de tool zelf, in dit geval de servers van Vidyo. Hierom wordt dit niet aanvaard als een correcte implementatie van een opname functie.

Het scherm delen, is net zoals die van Voxee, enkel mogelijk op de Web variant van de SDK. Het is mogelijk om van de Web implementatie een scherm te delen en deze zichtbaar te krijgen maar niet in de andere richting. [24]

Een tekst chat is een mogelijkheid met deze tool. [25] Er is geen Android of iOS documentatie aanwezig, maar voor ondersteuning of hulp bij het implementeren kan men terecht bij het support team.

De resolutie is voor Vidyo niet direct in te stellen maar wel de videokwaliteit. [26] Dit wordt ook geaccepteerd omdat de relatie tussen videokwaliteit en resolutie klein is en het ook mogelijk is dat Yunify de videokwaliteit aan wil passen.

Het is mogelijk om de wachtkamer zelf te ontwikkelen en er is net zoals bij de andere twee tools geen directe ondersteuning voor in de SDK.

4.5.5 Prijs

Deze tool werkt met een maandelijkse betaling van \$65.00 per maand. [27] Voor deze prijs worden 6500 *participant minutes* gegeven en voor elke minuut die er over gaat wordt \$0.01 gerekend. Deze minuten worden berekend per minuut per deelnemer. [28]

4.6 Conclusie

Hier gaan de verschillende aspecten met betrekking tot de tools tegenover elkaar gesteld worden en een besluit gemaakt worden van welke de beste is. In de vergelijkingsmatrix worden de aspecten schematisch geïllustreerd zodat er een duidelijk overzicht is van de aspecten en tools. Hieruit kan vervolgens elk aspect besproken worden om zo tot een conclusie te komen.

In de matrix wordt erg positief aangegeven door ++ en erg negatief met --, met ± als neutraal in het midden. N/A wordt aangegeven waar het niet mogelijk is om een score te geven.

Beoordelingsaspect	Agora	Voxeet	Vidyo
Documentatie	++	±	-
Implementatie	+	--	-
Netwerkprestaties	+	N/A	N/A
Overige beschikbare functies	±	-	±
Prijs	N/A	N/A	N/A

Tabel 7. *Vergelijkingsmatrix Tools*

De documentatie van Agora kwam bij alle drie de sub-aspecten als beste uit en krijgt daarom de hoogste score hierop. Omdat Voxeet en Vidyo hier beide niet op elke sub-aspect goed scoorden is Agora de beste als het aankomt op documentatie.

Bij de implementatie is Agora de enige die Flutter ondersteuning biedt en is daarom ook bij dit onderdeel de beste.

Omdat er geen POC van Voxeet en Vidyo in Flutter beschikbaar is geweest, zijn er geen testen uitgevoerd die onderzoeken hoe deze reageren op onverwachte netwerk gebeurtenissen. Hierdoor is Agora de enigste die een score heeft gehaald op dit aspect en daardoor ook hier de beste is.

Agora en Vidyo hebben beide de helft van de overige functies ter beschikking, terwijl Voxeet er een minder heeft. Hierdoor staan Agora en Vidyo als hoogste bij dit aspect.

Omdat de prijs bij alle drie de tools anders wordt berekend is hier geen vergelijking mogelijk omdat er nog geen schattingen bekend zijn. Indien er voorbeeld schattingen opgesteld worden, zouden deze een vertekend beeld geven op de resultaten als ze veel verschillen met de werkelijke waardes.

Zoals in bovenstaande matrix te zien is, staat Agora in elk aspect dat van toepassing is bovenaan. Daarom is Agora de beste tool om de functionaliteit van een videoconferentie het beste te implementeren in een Flutter-applicatie.

Algemene conclusie

Dit is een algemene conclusie over het gehele hoofdstuk van het onderzoek.

Over videoconferencing implementeren in Flutter met behulp van een externe tool zijn nog geen wetenschappelijke artikels geschreven. Er zijn wel een aantal artikels geschreven door de bedrijven die een tool beschikbaar stellen.[29][30] In die artikels wordt hun eigen tool meestal als beste gekozen, en dit soort artikels zijn daarom niet nuttig voor een onderzoek als dit.

In dit onderzoek is er gestart met 90 verschillende tools die een vorm van videoconferences bieden. Deze tools zijn vervolgens meerdere keren gefilterd op bepaalde criteria en uiteindelijk is er getracht om van de beste drie een *proof of concept* applicatie te maken. Na het vergelijken van deze *proof of concepts*, is gebleken dat Agora de beste tool is om de functionaliteit van een videoconferentie te implementeren in een Flutter-applicatie.

In de toekomst zou er onderzoek gedaan kunnen worden naar de exacte details van de Flutter-*engine*, dit om het uiteindelijk wel mogelijk te maken om een videoconferencetool die alleen Android en IOS ondersteuning biedt ook te implementeren in Flutter. Hierdoor zou ook de videokwaliteit en stabiliteit tussen de verschillende tools vergeleken kunnen worden. Dit was nu niet mogelijk doordat er maar één tool is die Flutter direct ondersteund.

Ondanks dat dit onderzoek een aantal specifieke criteria heeft gekregen vanuit Yunify, bestaat de wetenschappelijke relevantie uit het feit dat het onderzoek een lacune in de literatuur rond videoconferences in Flutter tracht op te vullen.

II. Stageverslag

In het onderzoek is er besloten dat Agora de beste tool is om de functionaliteit van een videoconference te implementeren in een Flutter-applicatie. Er is hier ook een *proof of concept* gemaakt van de implementatie van Agora. In dit gedeelte wordt deze *proof of concept* verder uitgewerkt tot een volledige applicatie rond videoconferenties. Hieronder worden de technieken, frameworks en protocollen die achter de applicatie zitten meer uitgelegd. Daarna worden ook alle functies uitgelegd die in de uitwerking zijn toegevoegd en ten slotte wordt er een conclusie getrokken over de gehele uitwerking van het project.

1. Technologieën

Hier worden de belangrijkste technieken, frameworks en protocollen die achter de applicatie zitten uitgelegd.

1.1 WebRTC

Web Real-Time Communication (WebRTC) is het protocol dat gebruikt wordt bij real time audio- en videogesprekken. Dit protocol wordt tevens bij alle onderzochte SDK's gebruikt.

1.2 Flutter

Dit framework is reeds gekozen door het projectteam en hiervan afwijken is dan ook niet zo maar mogelijk. Flutter is een open source framework voor de ontwikkeling van mobiele applicaties dat is gemaakt door Google. Het wordt voornamelijk gebruikt om applicaties voor Android en iOS, maar ze zijn ook aan het uitbreiden tot Flutter voor Web. Op 04/12/2018 werd de eerste "stabiele" versie uitgebracht van het Flutter-framework.

1.2.1 Dart

Dart is de taal waarin flutter wordt geschreven. Dart is door Google gekozen omdat het een aantal specifieke onderdelen biedt die niet terug te vinden zijn in andere talen. Hierdoor is het ook mogelijk om de *hot-reload* functie te hebben.

1.2.2 Hot reload

De *hot reload* functie maakt het mogelijk om tijdens het ontwikkelingsproces live updates te krijgen waarbij ook de huidige state niet verloren gaat. Dit helpt enorm bij het ontwikkelen om sneller iteraties van een applicatie te maken.

1.2.3 Platform specifieke kanalen

Method channels zijn een vanuit Flutter gegeven manier om te communiceren naar Android-code of iOS code. De communicatie gebeurt aan de hand van berichten. Deze worden met behulp van *channels* of kanalen tussen de Flutter code en host doorgegeven. Host verwijst hier naar het Android- en iOS-platform. Berichten en antwoorden worden asynchroon doorgegeven zodat de gebruikersomgeving blijft reageren terwijl dit bezig is. Indien nodig kunnen de berichten ook in omgekeerde richting worden verzonden.

1.2.4 Pub

Pub is de pakketbeheerder voor de Dart-programmeertaal, die herbruikbare bibliotheken en pakketten voor Flutter en algemene Dart-programma's bevat. Met een pakket wordt het delen en de herbruikbaarheid van de code tussen alle Dart-apps mogelijk.

2. Applicatie uitwerking

Hier wordt meer ingelicht over de belangrijkste functies die extra aan de *proof of concept* van Agora zijn toegevoegd.

2.1 Gebruikersomgeving verbergen

Als de gebruiker in een conference zit, worden alle elementen van de gebruikersomgeving verborgen. Dit betekent dat er op het scherm meer ruimte beschikbaar is waar de gebruiker de andere mensen in de conference op kan zien. Als de gebruiker de conference verlaat of de conference wordt beëindigd, komen alle elementen van de gebruikersomgeving weer terug zoals ze horen.

De elementen waar het hier over gaat zijn: het keyboard, de *navigation-*, *app-* en *statusbar*. Het keyboard en de *navigationbar* bevinden zich aan de onderkant van het scherm terwijl de *app-* en *statusbar* zich aan de bovenkant bevinden.

De *navigationbar* bestaat enkel op Android-apparaten en er zijn meestal drie knoppen aanwezig: de *back*, *home* en *overview button*. In sommige gevallen is het mogelijk om deze knoppen naar wens aan te passen en zullen er andere knoppen aanwezig zijn. De applicatie zal dan nog altijd de gehele *navigationbar* verbergen. Als deze balk aanwezig is, bevindt hij zich altijd aan de onderrand.

Het keyboard is standaard niet aanwezig op het scherm, enkel als de gebruiker iets moet typen. Hij bevindt zich net boven de *navigationbar* als deze bestaat en anders bevindt het keyboard zich aan de onderrand.

De *statusbar* bevindt zich aan de bovenrand van het scherm en bevat meldings- en systeempictogrammen, zoals de tijd, mobiele provider, netwerkstatus en batterijniveau.

Als laatste de *appbar*, dit is een speciale ruimte waar de locatie van de gebruiker in de app aan wordt gegeven. Hier zijn vaak ook belangrijke acties terug te vinden op een voorspelbare manier, zoals de navigatie- of zoekknop.

2.2 Omkeren van de camera

In de conference is een knop aanwezig die ervoor zorgt dat er een andere camera gebruikt wordt dan de huidige. Als de camera aan de voorkant wordt gebruikt, en er wordt op deze knop gedrukt, gaat de camera aan de achterkant gebruikt worden en vice versa. Indien er maar een enkele camera, of zelfs geen, voorhanden is zal deze knop simpelweg niks doen.

Met de hedendaagse snel evoluerende mobiele telefoons is het ook mogelijk dat er meer dan twee camera's op een mobiel aanwezig zijn. Indien dit het geval is, wordt bij elke druk op de knop de volgende camera gepakt en als ze allemaal geweest zijn begint deze weer van voor af aan.

2.3 Beweegbaar eigen scherm

Als de gebruiker in de conference zit, is het mogelijk om zichzelf te zien door middel van een klein schermje. Indien dit schermje in de weg staat, het zou bijvoorbeeld voor iemand anders zijn gezicht kunnen staan, is het mogelijk om dit schermje te verplaatsen. Dit wordt gedaan door te slepen, de gebruiker drukt op zijn eigen schermje en beweegt het naar zijn nieuwe gewenste positie op het scherm en laat het hier los. Het schermje zal nu op deze gewenste positie blijven staan totdat de gebruiker hem weer verplaatst.

Een bijzonderheid hiervan komt voor wanneer de gebruiker het schermje gedeeltelijk buiten het scherm van de mobiel probeert te plaatsen. In dit geval zal de applicatie de positie van het nieuwe schermje aanpassen zodat het altijd volledig op het scherm van de mobiel blijft. De nieuwe positie zal aan de rand van het scherm zijn, zo dichtbij mogelijk de positie waar de gebruiker het schermje oorspronkelijk probeerde te plaatsen.

2.4 Vergroot modus

Terwijl de gebruiker in de videoconference zit, is het mogelijk om te dubbelklikken op het schermje waarin hij zichzelf ziet om zijn eigen schermje te vergroten. Dit vergrote scherm wordt geplaatst in de onderste helft van het scherm van de mobiel, waarbij de overige personen in de conference gelijkmatig verspreid worden over de bovenkant van het scherm. Het originele kleine schermje wordt tevens verwijderd van het scherm omdat het grote scherm ervoor in de plaats is gekomen.

De gebruiker kan vervolgens opnieuw dubbelklikken op het vergrote scherm, dit zorgt ervoor dat alles terug gaat naar de originele modus. Het kleine schermje wordt opnieuw zichtbaar en alle andere personen in de conference worden over de volledige achtergrond verdeeld.

2.5 Voorkeuren

In de applicatie worden de voorkeuren van de gebruiker onthouden zodat de gebruiker zonder problemen kan wisselen tussen conferences, maar ook wanneer de applicatie opnieuw wordt opgestart. De twee dingen die op dit moment worden onthouden zijn uitgelegd in de vorige twee onderdelen.

Het eerste dat onthouden wordt, is de positie van het kleine schermje waarin de gebruiker zichzelf kan zien. Het schermje is verder uitgelegd in onderdeel 2.3. Als de gebruiker het schermje verplaatst en vervolgens de conference verlaat of de applicatie sluit, ziet hij bij de volgende keer als hij in een conference komt, dat het schermje nog altijd op deze nieuwe plaats staat.

Het tweede is de modus van het scherm, dit is in het onderdeel hierboven meer ingelicht. De applicatie onthoudt of de gebruiker voor het laatst in de vergrote modus zat of niet. De volgende keer als de gebruiker in een conference komt, zal hij in dezelfde modus terecht komen waarmee hij de vorige conference heeft verlaten.

2.6 Thema

Alle kleuren in de applicatie zijn niet vastgezet en worden allemaal van één centraal thema afgehaald. Dit maakt de implementatie van andere thema's, in dit geval het thema van Yunify, zeer eenvoudig. Dit zorgt er ook voor dat het eenvoudig is om meerdere thema's aan de gebruiker voor te stellen zodat deze hieruit kan kiezen. Dit laatste is echter niet in de applicatie verwerkt omdat dit niks toevoegt rond het onderwerp van videoconferencing.

2.7 Rating

Als de conference is afgelopen wordt er een venster aan de gebruiker getoond waarin hij de voltooide conference kan beoordelen. Hij kan dit doen aan de hand van een beoordeling op vijf sterren. Er is ook de mogelijkheid om geen beoordeling te geven door buiten het venster te klikken, hierdoor wordt het venster automatisch gesloten.

Bij het implementeren van de sterbeoordeling, is er een Pub pakket gebruikt die de logica achter het slepen van het bereik van de sterren verzorgt.

Op dit moment worden de ratings die ingegeven worden nergens opgeslagen. Er is echter wel rekening gehouden met het feit dat dit later wel de bedoeling kan zijn. Het is hierom zo gemaakt dat het eenvoudig is om dit later wel te implementeren in de huidige applicatie.

3. Conclusie

De *proof of concept* implementatie van Agora is in het project tot een volledig bruikbare applicatie uitgebreid. Er is waar mogelijk rekening gehouden met het feit dat deze applicatie geïmplementeerd gaat worden in een groter geheel. Het is voor Yunify mogelijk om deze applicatie zonder veel veranderingen te implementeren in de volledige applicatie die zij aan het ontwikkelen zijn.

In de toekomst is het mogelijk om deze applicatie uit te breiden met bepaalde functies. Een van deze functies is de mogelijkheid om het gesprek op te nemen en later terug te bekijken. De SDK van Agora levert deze functie op het moment van het project nog niet mee, er is echter via e-mail vernomen dat deze functie rond augustus van dit jaar volledig beschikbaar wordt.

4. Reflectie

Ik ben veel werkervaringen rijker door deze stage. Ik heb geleerd hoe het er dagdagelijks aan toe gaat in een softwareontwikkelingsbedrijf. Hoe de agile technieken die ik heb geleerd op school zoals de dagelijkse stand-up meeting of de code reviews er in het echt aan toe gaan. Ik heb leren omgaan met het feit dat het lijkt alsof iedereen meer weet dan mij. Ik heb in de praktijk mijn professionele schrijfstijl leren toepassen bij het uitreiken naar de bedrijven waarnaar ik onderzoek deed. Ik heb veel geleerd over Word en Excel bij het maken van het eindwerk zelf en ik kan die skills ook in andere gebieden gebruiken. Ik heb meegemaakt hoe de lagen van management in een (middelgroot) bedrijf werken.

Maar niet te vergeten heb ik leren werken met het Flutter-framework. Ik ben nog ver van een expert hierin maar ik weet wel dat het een heel andere manier van programmeren is dan dat ik gewend ben vanuit school. Hierdoor heb ik ook geleerd dat het eigenlijk niet belangrijk is wat je precies kent maar eerder dat het belangrijk is dat je de mogelijkheid hebt om snel iets nieuws te leren.

Mijn stage was vooral een onderzoeksstage, ik heb in de eerste 6 weken onderzoek gedaan naar de beste mogelijkheden voor de toepassing wat Yunify nodig had. Hierna heb ik drie weken gewerkt aan de beste drie tools die uit het onderzoek waren gekomen te implementeren in een proof of concept applicatie. De resterende drie weken heb ik verder gewerkt aan de beste van deze drie verder uit te werken in een applicatie. Het project waarin ik werkte was nog in de beginfase, hierdoor was het zelfs voor hun niet altijd duidelijk wat er precies nodig was voor het project. Het was natuurlijk een unieke ervaring om zo vroeg bij een project te zijn maar dit maakten het onderzoeken van de beste tool er niet makkelijker op omdat de requirements van de tool een aantal keer veranderden.

Yunify had een tool nodig die ze de functionaliteit van een videoconference leverde in een Flutter-applicatie. Het Flutter-framework is pas recent gelanceerd. Dit, in combinatie met het feit dat de meeste tools nog geen Flutter ondersteunen, zorgde ervoor dat ik voor veel problemen kwam te staan bij het implementeren van de tools in Flutter. Ik heb dit eerst proberen op te lossen door mijzelf in de tools te verdiepen en toen dit niet genoeg was, heb ik zoveel mogelijk hulp gevraagd aan mensen rond me, de bedrijven van de tools en op internetfora. Uiteindelijk heb ik hieruit twee dingen geleerd. Als eerste dat ik soms moet opgeven als er te veel tijd en resources aan iets wordt besteed omdat het na een bepaalde tijd niet meer waard is om er nog meer tijd aan te besteden. En als tweede heb ik de effecten van slechte documentatie gevoeld en hierdoor gerealiseerd hoe belangrijk het is om goede documentatie te schrijven.

Ik ben blij dat het uiteindelijk toch is gelukt om één van de drie tools te implementeren en verder uit te werken tot een volledig functionele app rond videoconferencing. En dat de app die ik gemaakt heb ook geïntegreerd zal worden in de volledige app van Yunify.

Door de ervaringen die ik in mijn stage heb opgedaan denk ik dat ik mijzelf wel kan redden in het bedrijfsleven.

Bibliografie

- [1] "Investeringsmaatschappij en holding | IIC GROUP." [Online]. Available: <https://iicgroup.nl/#homePage>. [Accessed: 08-Mar-2019].
- [2] "Wij innoveren gezondheid en welzijn | Yunify - We Take Care." [Online]. Available: <https://yunify.nl/nl/>. [Accessed: 08-Mar-2019].
- [3] "What is the Difference Between an API and an SDK? | Nordic APIs |." [Online]. Available: <https://nordicapis.com/what-is-the-difference-between-an-api-and-an-sdk/>. [Accessed: 04-Apr-2019].
- [4] "Does PubNub provide WebRTC and video chat? : PubNub Global Support." [Online]. Available: <https://support.pubnub.com/support/solutions/articles/14000043715-does-pubnub-provide-webrtc-and-video-chat->. [Accessed: 29-Mar-2019].
- [5] "OpenTok Basics." [Online]. Available: <https://tokbox.com/developer/guides/basics/#opentok-platform>. [Accessed: 29-Mar-2019].
- [6] "Vidyo.io Documentation | WebRTC Platform for Video, Voice and Text." [Online]. Available: <https://developer.vidyo.io/#/documentation/4.1.25.49%23KeyConcepts>. [Accessed: 29-Mar-2019].
- [7] "Cisco Rebrands Spark as Webex Teams and Updates Platforms | GetVoIP." [Online]. Available: <https://getvoip.com/blog/2018/04/18/cisco-spark-rebrand/>. [Accessed: 29-Mar-2019].
- [8] "Putting comments in code: the good, the bad, and the ugly." [Online]. Available: <https://medium.freecodecamp.org/code-comments-the-good-the-bad-and-the-ugly-be9cc65fbf83>. [Accessed: 05-May-2019].
- [9] "Agora API Reference for All Platforms." [Online]. Available: https://docs.agora.io/en/Video/APIReference/cpp/classagora_1_1rtc_1_1_i_rtc_engine.html#ac71db65e66942e4e0a0550e95c16890f. [Accessed: 09-May-2019].
- [10] "Agora Flutter SDK." .
- [11] @plutoless, "Agora-Android-Tutorial." [Online]. Available: <https://github.com/AgoraIO/Basic-Video-Call/tree/master/One-to-One-Video/Agora-Android-Tutorial-1to1>. [Accessed: 03-May-2019].
- [12] "Agora Recording Overview." [Online]. Available: https://docs.agora.io/en/Recording/product_recording?platform=All Platforms. [Accessed: 06-May-2019].
- [13] "Share the Screen." [Online]. Available: https://docs.agora.io/en/InteractiveBroadcast/screensharing_android?platform=Android. [Accessed: 06-May-2019].
- [14] "Set the Video Profile." [Online]. Available: https://docs.agora.io/en/Video/videoProfile_android. [Accessed: 06-May-2019].
- [15] "Agora.io Pricing - Agora.io." [Online]. Available: <https://www.agora.io/en/price/>. [Accessed: 09-May-2019].
- [16] "Pricing and Billing." [Online]. Available: https://docs.agora.io/en/Agora Platform/billing_faq. [Accessed: 09-May-2019].

- [17] "Screensharing API for Mobile & Web made Easy on Voxeet." [Online]. Available: <https://www.voxeet.com/launching-3-new-critical-collaboration-functions-voxeet/>. [Accessed: 09-May-2019].
- [18] "Android Voxeet sample." [Online]. Available: <https://github.com/voxeet/android-sdk-sample/blob/master/app/src/main/java/fr/voxeet/sdk/sample/activities/MainActivity.java>. [Accessed: 09-May-2019].
- [19] "Get started with WebRTC Android Video Chat API | Voxeet." [Online]. Available: <https://developer.voxeet.com/getstarted/android/getstarted-Android/>. [Accessed: 09-May-2019].
- [20] "Do you offer recording? Do you store recordings on your server? – Voxeet." [Online]. Available: <https://voxeet.zendesk.com/hc/en-us/articles/360018580952-Do-you-offer-recording-Do-you-store-recordings-on-your-server->. [Accessed: 09-May-2019].
- [21] "Pricing | Voxeet." [Online]. Available: <https://www.voxeet.com/pricing/>. [Accessed: 09-May-2019].
- [22] "Vidyo.io Documentation | WebRTC Platform for Video, Voice and Text." [Online]. Available: <https://developer.vidyo.io/#/documentation/4.1.25.49>. [Accessed: 09-May-2019].
- [23] "How to record a vidyo.io conference using docker and AWS EC2." [Online]. Available: <https://vidyo.io/blog/how-to/record-video-chat/>. [Accessed: 14-May-2019].
- [24] "Vidyo Connector Developer Guide 4.1.25.49." [Online]. Available: <https://static.vidyo.io/4.1.25.49/docs/VidyoConnectorDeveloperGuide.html#screensharing>. [Accessed: 09-May-2019].
- [25] "Video Chat Adding Text Chat Features | Advanced Video Chat Signalling." [Online]. Available: <https://vidyo.io/blog/how-to/video-chat-add-text-chat/>. [Accessed: 09-May-2019].
- [26] "Increase the Video Quality on Mobile – Vidyo.io." [Online]. Available: <https://support.vidyo.io/hc/en-us/articles/360000268074-Increase-the-Video-Quality-on-Mobile>. [Accessed: 09-May-2019].
- [27] "Pricing for Vidyo.io Video Chat API | Video Chat SDK | Video Chat WebRTC." [Online]. Available: <https://vidyo.io/pricing/>. [Accessed: 09-May-2019].
- [28] "How are minutes calculated for billing purposes? – Vidyo.io." [Online]. Available: <https://support.vidyo.io/hc/en-us/articles/115006344708-How-are-minutes-calculated-for-billing-purposes->. [Accessed: 09-May-2019].
- [29] VSee, "Video Conference and Screen Share Product Reviews - VSee." [Online]. Available: <https://vsee.com/videoconference>. [Accessed: 02-Apr-2019].
- [30] "Comparing Online Meetings Products - Cisco." [Online]. Available: <https://www.cisco.com/c/en/us/products/conferencing/collaboration-competitive-comparison.html#~competitive=0+1+2+3+4+5+6+7>. [Accessed: 19-Mar-2019].

