Onderwijsgroep Professionele Opleidingen

Toegepaste Informatica

IntegrAAL

Integration of Active Assistive Living Components for Innovative Care Pathways

Bachelorproef aangeboden door
Koumeyl Belkhidar
tot het behalen van de graad van
Bachelor in Toegepaste informatica

Bachelorproefbegeleider Odisee**: Yvan Rooseleer**

Academiejaar 2016 – 2017

Inhoudstafel

1. Voorwoord

Deze bachelorproof werd afgewerkt met Mjouh Soufiane. Het werk werd verdeeld en sommige onderdelen werden samen afgewerkt andere werden zelfstandig afgewerkt.

Er werd gekozen samen met mijn medestudent om een eigen project te maken met samenwerking van het project IntegrAAL, onder begelijding van Ellen De Cuyper, Mieke Beckwé en Yvan Rooseleer.

Na veel afspraken en veel geleverde werk. Zijn we uiteindelijk tot een goed resultaat gekomen en hebben we ons integratieproject kunnen afwerken.

Ik zou graag mijn familie willen bedanken voor hun ondersteuning. Ze waren altijd klaar om me aan te moedigen. Ik zou ook mijn vrinden willen bedanken met wee ik altijd kon spreken en ook hun visie van ons applicatie met mij konden delen.

Het was voor ons wel een uitdaging om een project van deze omvang van het begin tot het einde goed te managen en te timen. Maar dankzij het project hebben we veel geleerd en hebben we een goede visie van wat er gedaan moet worden en wat de risico's zijn.

2. Inleiding

Als laatstejaarsstudenten in de bachelor Toegepaste Informatica zijn er twee opleidingsonderdelen dat belangrijk zijn en waarop veel gefocust moet worden. Het stage en het Integration Project die een grondig voorbereiding is op de stage.

Dit hebben we zeker niet onderschat en we wisten dat het onderwerp hiervan heel belangrijk zou zijn.

Het project op zichzelf is een uitdaging dat ons veel zou leren. Een applicatie ontwikkelen voor IntegrAAL was het soort project dat we nodig hadden. Niet alleen omdat het onderwerp boeiend was maar ook omdat het project nieuwe technologien bevat waaruit we veel zouden uitleren.

3. BESCHRIJVING VAN DE OPDRACHTGEVER

IntegrAAL is een project dat als doel heeft het leven van oudere mensen te verbeteren. De wereld is in constant evolutie en vandaag de dag zijn er technologieën beschikbaar waarmee mensen geholpen kunnen worden. Bejaarde mensen hebben tot nu toe geen of weinig contact met de voordelen van dit evolutie. IntegrAAL wilt het uitdaging grijpen en wilt de oudere mensen helpen met de dagelijkse problemen waarmee ze geconfronteerd worden. Dit door innovatieve technologieën in te zetten. IntegrAAL wilt een volledig infrastructuur ontwikkelen om zorgverleners te helpen bij de organisatie en het managen van de behoeftes van een oudere persoon. Vanaf eind 2015 is Integraal bezig met het ontwikkelen en testen van nieuwe technologiesystemen. De applicatie wordt getest op 3 locaties. Belgie, Portugal en de Verenigd-Koningrijk (voor meer informatie over het integraal project zie het bijlage functionele analyse 1.3)

4. CONTEXT EN DEFINITIE VAN DE OPDRACHT

Het IntegrAAL kwam bijna aan zijn einde. Toch werd er door het bedrijf geconstateerd dat hun applicatie minpunten bevat. IntegrAAL heeft hun omgeving beschikbaar gesteld zodat derden eventueel met de applicatie en het omgeven verder kunnen ontwikkelen. Dit onderdeel is beschikbaar maar werd nog niet getest. Een tweede probleem dat ondervonden werd is dat het applicatie te ingewikkeld is met de applicatie en moeilijk ermee kunnen omgaan. Hier werd aan ons gevraagd om een applicatie de maken dat gebruikersvriendelijk zou zijn voor bejaarde mensen. Het is de bedoeling om een zelfstandig applicatie te maken dat de gegevens van het bedrijf gebruikt.

De applicatie moet verschillende vereisten hebben. Het is de bedoeling zoals eerder gezegd dat het gebruikersvriendelijk is voor bejaarde mensen. De applicatie moet ook een call systeem hebben met video zodat de zorgverleners het persoon kunnen zien of het persoon effectief in orde is. Het oudere persoon moet het call systeem gemakkelijk kunnen gebruiken. Er werd ook gevraagd om te zien of het mogelijk is om verschillende functionaliteiten vocaal te kunnen uitvoeren zodat bejaarde mensen gemakkelijk sommige functies kunnen gebruiken.

Het project werd zo ingedeeld. Soufiane Mjouh was verantwoordelijk voor het functioneel analyse. Hier moet Soufiane onderzoeken wat de applicatie moet kunnen en wat het niet moet kunnen. Het is ook de bedoeling dat Soufiane de verschillende functionaliteiten gaat uittesten op het systeem van integraal om te zien of het gebruiken van het systeem mogelijk is en of er verder een applicatie gemaakt kan worden.

Ik was verantwoordelijk voor de opzoeking van de technologieën dat gebruikt konden worden. Het was de bedoeling om te onderzoeken welke voice technologieën er bestaan en met welke technologieën is het compatible. Welke programmeertaal ondersteund zo een technologie. De programmeertaal moet ook de mogelijkheid hebben om een video call te kunnen maken. Uiteindelijk moest ik ook beslissen op welke device het het best ging werken. Is het beter op een smartphone of op een pc? Android of PHP? Dit zijn de soorten vragen dat ik probeer te beantwoorden.

5. FUNCTONELE VEREISTEN

Aangezien IntegrAAL al een systeem met app heeft ontwikkeld hebben we ons hierop eerst gebaseerd.

Het project probeert te weten te komen hoe ze in het eerste instantie een beter comprehensie kunnen hebben over de uitdaging dat oudere mensen dagelijks kennen. Nadat dit begrepen is hoe zou IntegrAAL aan de hand van beschikbare technologieën aan de behoeftes van oudere mensen kunnen antwoorden en hun leven verbeteren.

Het IntegrAAL werkt op twee vlakken.

Coördinatie creëren tussen de verschillende personen dat voor het oudere mens zorgen.

Ondersteunen van personen dat voor zichzelf verzorgen met steun van de personen om hen heen.

Hiervoor werden er drie "softwareblokken" samengebracht

- Needs Assessment en Care
- Zorgopnames en naleving monitoring
- Self-management Reminders en Escalation

Need assessment en care evalueert door een proces of een persoon hulp nodig heeft of niet. Dankzij dit assessment kunnen de zorgverleners de noden van het persoon meten. Het dient ook om erna een ondersteuningsplan voor de persoon op te stellen.

Zorgopnames en naleving monitoring is een software dat de planning van oudere persoon organiseerd.

Self-managment reminders en escalation is een systeem dat het opvolging van alarmen beheert. Een alarm kan afgevuurd worden als er iets niet goed is gebeurd met het bejaarde persoon. Het alarm zal alleen afgevuurd worden naar op de persoon dat met het probleem direct betrekking heeft. Als het persoon niet antwoord wordt het alarm aan een andere persoon doorgestuurd. Dankzij dit methode kan het persoon reageren vooraleer andere personen betrokken worden. Voor andere meer gedetailleerde informatie verwijs over het werking van IntegrAAL ik u naar het bijlage: "Functional Analysis: Nourishcare".

6. TECHNISCHE VEREISTEN

Google voice action

Als we het over spraaktechnologie hebben denken we eerst aan Google Voice Action. Google voice action is een bekende spraaktechnologie dat de mogelijkheid geeft om taken uitvoeren via een Android toestel. Het technologie is gelinkt aan Knowledge Graph.

Knowledge Graph is een technologie dat het informatie over iets en de relaties die eraan gekoppeld zijn bijhoud. Dankzij Knowledge Graph is het kans dat een opzoeking niet slaagt van 20 % naar 8% verminderd in de laatste jaren.

Google Voice Action geeft de mogelijkheid om System Voice Actions en Custom Voice Actions toe te voeren. System Voice Actions zijn standaard acties dat mensen met hun Android apparaat willen doen. Google identificeert de acties en gaat dan definiëren welke app dit actie gaat kunnen uitvoeren. Voorbeelden van zo actions zijn:

- Alarm actions
- Communication actions
- Search actions
- Enz.

Custom Voice Actions daarentegen zijn acties dat door ontwikkelaars ge designed zijn om hun eigen acties met spraak te voeren.

Na mijn analyse heb ik het documentatie van google doorgenomen en hebben we ondervonden dan Custom Voice Actions niet meer ondersteund zijn. Maar dankzij dit analyse zijn we te weten te komen dat google een firma heeft aangekocht dat aan spreaaktechnologie en AI(Artificial Intelgence) doet namenlijk api.ai (voor meer informatie over Google voice action zie bijlage: Goog voice action.

7. Api.ai

API.AI (vroeger bekend als Speaktoit) is een Amerikaans startup gelanceerd in 2011. Het bedrijf biedt een gemakkelijke manier om intelligente en geavanceerde conversational userinterfaces in applicaties, apparaten en diensten te integreren. Dit om de gebruikerservaring en de stroomlijn van bedrijfsprocessen te verbeteren. Het platform bevat onder andere tools om conversaties te managen zodat ze op een natuurlijke manier verlopen. Wat een pluspunt is dat het platform integreerbaar is in tiental programeertalen. Api.ai is voledig gratis. Verder werden er verschillende testscenario's uitgevoerd in api.ai (voor meer informatie zie bijlage api.ai).

Project evaluatie

Aangezien mijn begeleiders Ellen en Mieke een meeting hadden in Portugal om het prototype voor te stellen, heeft Ellen haar evaluatie per mail doorgestuurd. Een kopij ervan vindt u in bijlage "Mail Projectevaluatie".

8. Bronnenlijst

Voice recognition technologie vergelijking

https://www.quora.com/What-are-the-top-ten-speech-recognition-APIs

https://www.microsoft.com/cognitive-services/en-us/speech-api

https://developer.apple.com/sirikit/

http://www.businessinsider.com/siri-vs-google-assistant-cortana-alexa-2016-11/#call-me-an-uber-6

Google-assistant en Google Voice Actions

Action on google playlist :

https://www.youtube.com/watch?v=ulAKc2gezfU&list=PLOU2XLYxmsIKgPTizdYWPPYEpU96FCJrQ

https://developers.google.com/actions/develop/apiai/tutorials/getting-started

For android :

https://github.com/api-ai/api-ai-android-sdk

https://developers.google.com/voice-actions/system/

http://www.pocket-lint.com/news/137722-what-is-google-assistant-how-does-it-work-and-which-devices-offer-it

Google voice Actions :

http://blog.prolificinteractive.com/2015/11/06/implementing-google-voice-actions-into-your-android-app/

https://www.cnet.com/how-to/the-difference-between-google-now-and-google-assistant/

https://androidworld.nl/apps/ok-google-spraakcommandos-nu-ook-het-nederlands/

http://io2015codelabs.appspot.com/codelabs/voice-interaction#1

https://developers.google.com/voice-actions/interaction/voice-interactions

video: https://www.youtube.com/watch?v=OW1A4XFRuyc

https://www.youtube.com/watch?v=mgudsc-Z468

Launch app with googleNow:

http://www.techrepublic.com/article/pro-tip-use-google-now-to-launch-your-next-android-app/

API.AI en integratie in Android en Voice on google

https://docs.api.ai/docs/welcome

http://daslhub.org/unlv/wiki/doku.php?id=api_ai_tutorial

https://github.com/api-ai/api-ai-android-sdk

https://en.wikipedia.org/wiki/Speaktoit

https://www.crunchbase.com/organization/api-ai#/entity

https://api.ai/blog/2016/09/19/api-ai-joining-google/

http://venturebeat.com/2016/09/19/google-acquires-natural-language-understanding-startup-api-ai/

http://www.zdnet.com/article/google-acquires-api-ai-to-build-conversational-interfaces/

https://www.facebook.com/pg/apiaiofficial/about/?ref=page_internal

https://www.linkedin.com/company/speaktoit

https://en.wikipedia.org/wiki/Natural_language_understanding

https://en.wikipedia.org/wiki/Natural_language_processing

Hardware Technical Analyse

http://coolpile.com/gadgets-magazine/mini-ios-and-android-multi-media-bluetooth-remote-control-by-satechi

http://developer.samsung.com/tv/develop/api-references/samsung-product-api-references/microphone-api

http://www.samsung.com/be_fr/consumer/tv-audio-video/tv-audio-accessories/tv-accessories/CY-STC1100/XC/

Rasbery analyse

https://nl.wikipedia.org/wiki/Singleboardcomputer

https://en.wikipedia.org/wiki/Raspberry_Pi

https://nl.wikipedia.org/wiki/Raspberry_Pi

https://fr.wikipedia.org/wiki/Raspberry_Pi

https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/

https://www.raspberrypi.org/magpi/android-raspberry-pi/

https://www.youtube.com/watch?v=Df-bMWONIYk

how to install android:
https://www.youtube.com/watch?v=SI20FBkc6EA&feature=youtu.be

https://www.youtube.com/watch?v=B_qHkdjHmz0

https://www.youtube.com/watch?v=Ylc0i2OybE0

Os:
https://android.jlelse.eu/getting-started-with-android-things-b73be3295b42#.wg4m3t3b0

https://rtandroid.embedded.rwth-aachen.de/downloads/raspberry-pi/

http://www.androidauthority.com/what-is-android-things-gary-explains-740572/

Remote:
https://www.youtube.com/watch?v=5LkdeE2zA3w

Android en device

Android app on boot: https://thebitplague.wordpress.com/2013/04/05/kiosk-mode-on-the-nexus-7/

https://android.jlelse.eu/getting-started-with-android-things-b73be3295b42#.wg4m3t3bo

Project management

https://www.projectmanagement-training.net/category/six-phases/

https://www.medialon.com/support/faq/how-to-write-a-functional-analysis/

FireBase

https://www.youtube.com/watch?v=wVCz1a30gqk

Amazon: Alexa

Bijlage VIII: API.AI

Door: Koumeyl Belkhidar

API.AI

Introductie

In kader van ons project gaan we ons voor de technische analyse meer verdiepen in de verschillende spraak en artificiële intelligentie technologieën. API.AI is een van de pioniers hiervan op de markt. We gaan de mogelijkheden hiervan analyseren en testen, om te zien of het wel past in de context van ons project.

API.AI

API.AI (vroeger bekend als Speaktoit) is een Amerikaans startup gelanceerd in 2011. Het bedrijf is eerst bekend geworden met Assistant, een intelligente personeel assistent applicatie dat meer dan 10 miljoen keer gedownload werd op de Google Play Store. Toch koos het bedrijf in december 2014 om een NLU (natural language understanding) platform voor softwareontwikkelaars te worden. Zo biedt API.AI een gemakkelijke manier om intelligente en geavanceerde conversational userinterfaces in applicaties, apparaten en diensten te integreren. Dit om de gebruikerservaring en de stroomlijn van bedrijfsprocessen te verbeteren.

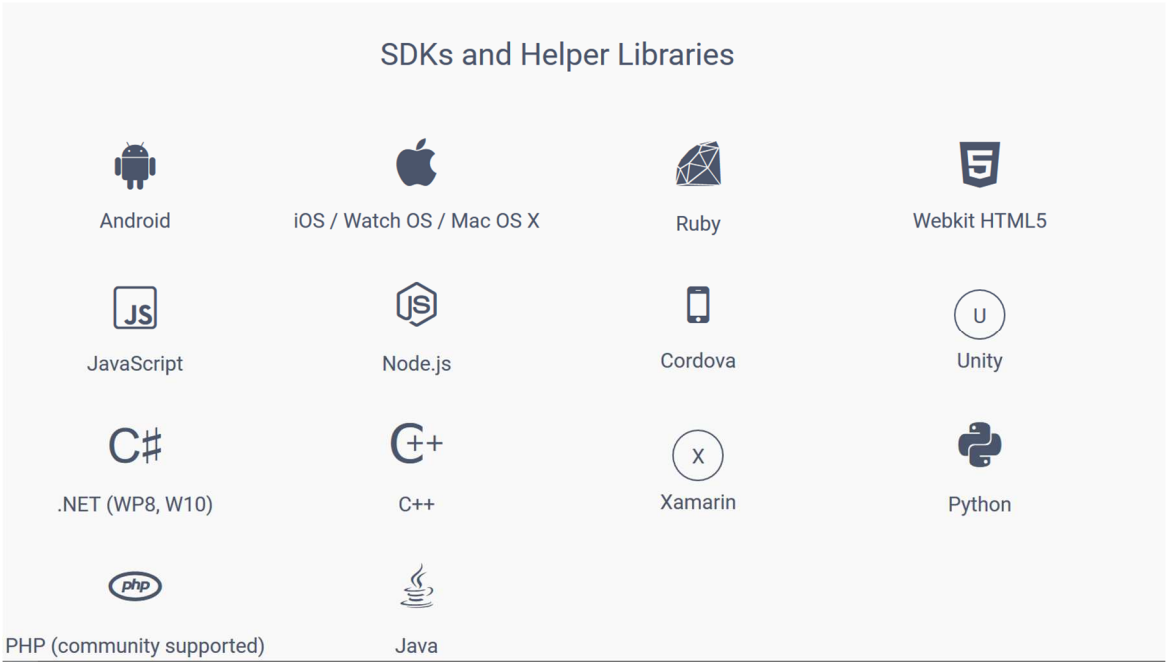
Het platform bevat onder andere tools om conversaties te managen zodat ze op een natuurlijke manier verlopen. Het platform gebruikt "machine learning", een technologie dat computers de capaciteit geeft om alleen bij te leren om gebruikers beter te kunnen begrijpen. Het biedt ook een Automatic Speech Recognition (ASR) systeem om gesproken woorden in leesbare tekst te transcriberen. Het platform biedt integraties met Webhook, Facebook Messenger, Alexa enz...

[1]

In 2016 werd api.ai overgenomen door Google. Het platform wordt vandaag gebruikt door meer dan 60000 developers en is volledig gratis. Het is ook performant en werd op meer dan een miljard aanvragen getest. API.AI is beschikbaar in 15 talen maar sommige nieuwe features zijn momenteel alleen in het Engels beschikbaar.

[1] https://api.ai/pricing/

API.AI is multiplatform. Het ondersteund maar ook levert verschillende sdks en libraries. Hier vind u het lijst hiervan.



Testscenario's uitwerken in API.AI.

We gaan enkele scenario's uitwerken om te evalueren of de technologie voldoet aan de eisen van ons project. Om de nodige scenario's te schrijven gaan we ons baseren op de features dat momenteel beschikbaar zijn in het IntegrAAL applicatie.

Eerste scenario

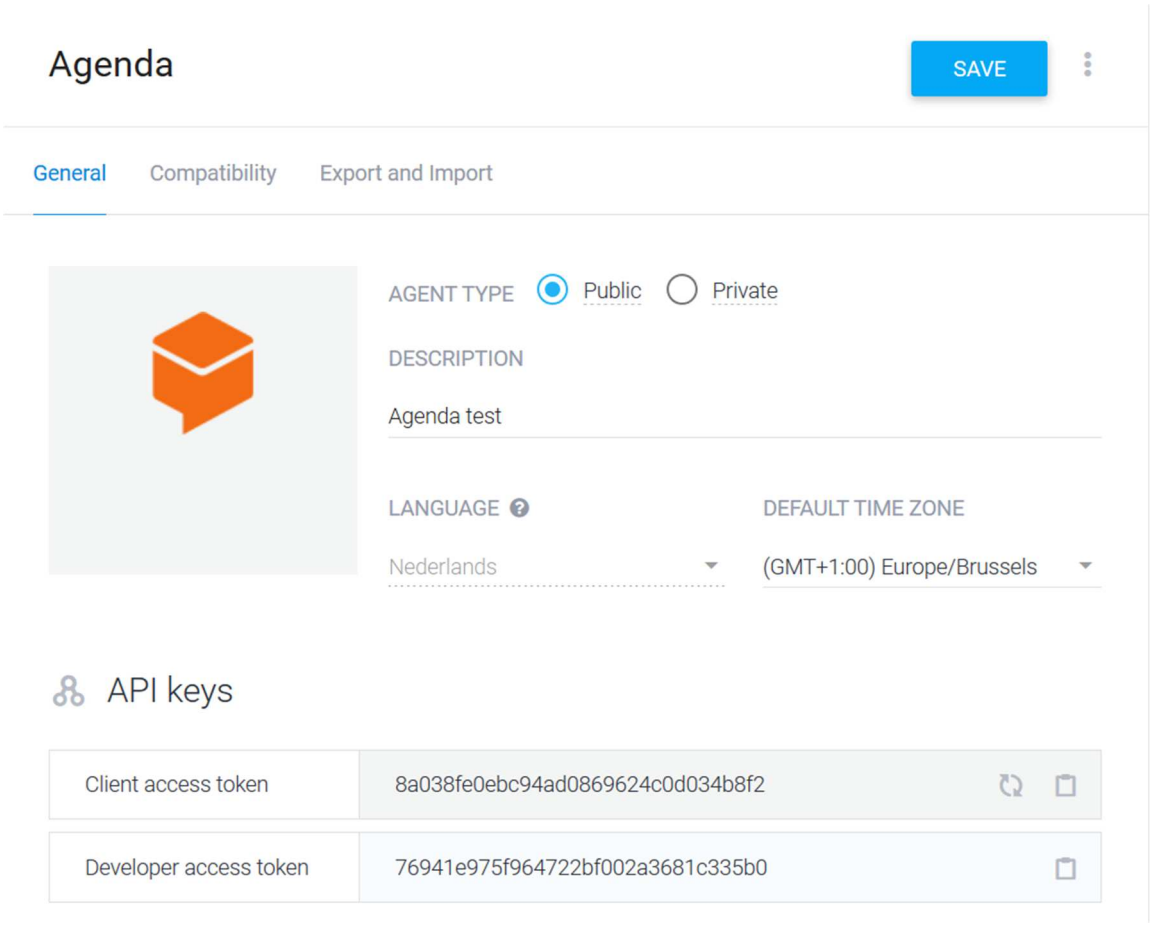
Eerste scenario is het mogelijkheid om een activiteit in een agenda op te slaan. De gegevens die we moeten opslaan zijn de volgende.

1. Een activiteit dat de gebruiker wilt opslaan.
2. Een uur waarop hij dit activiteit wilt plaatsen in de agenda.
3. Het dag waarop de activiteit zal plaatsvinden.

Uitwerking

Agent

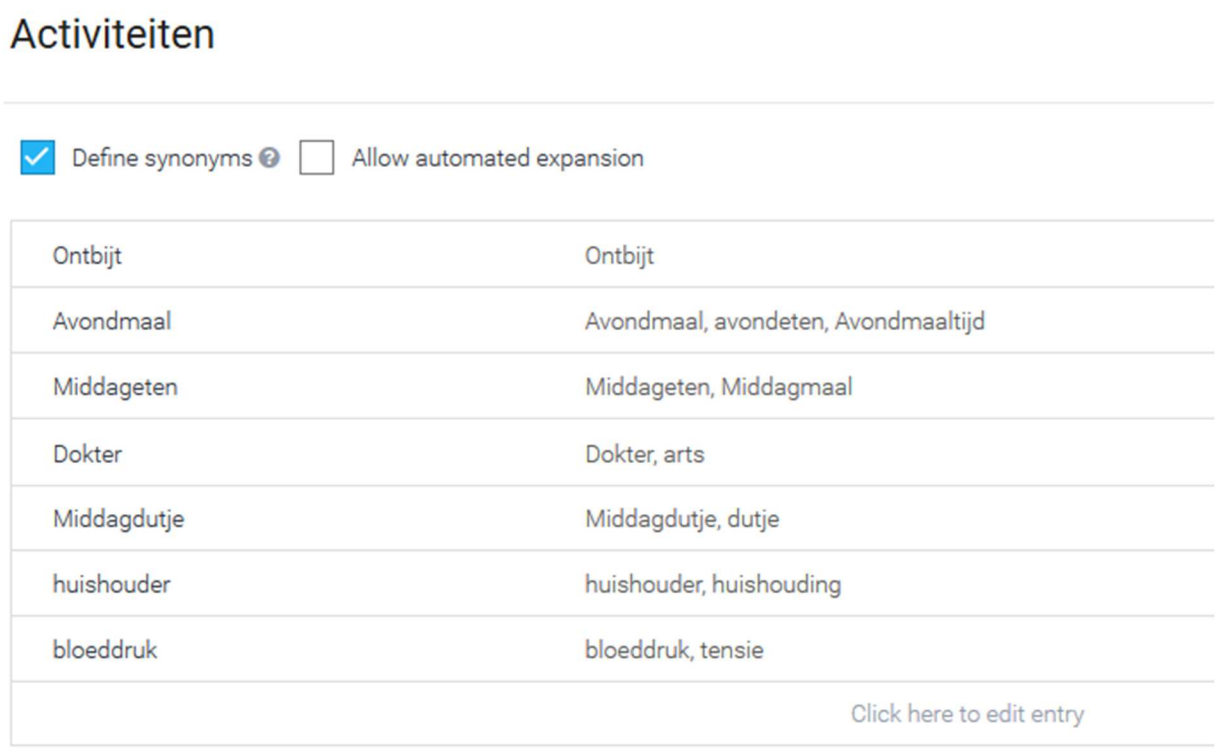
We gaan eerst een nieuwe agent moeten maken. Ons agent is eigenlijk het AI (Artificial Intelligence) programma dat we gaan maken waarin on scenario bewerkt gaat worden.



We gaan het agent een naam geven in ons case "Agenda". We moeten het taal van het agent aangeven. Het taal kunnen we achteraf niet veranderen. Als men de applicatie in een andere taal willen maken moeten we het volledig hermaken. Er zullen ook tokens gegeneerd worden. Via dit token krijgen externe applicaties toegang tot het agent.

Entities

Eerst moeten we de nodige entities definiëren. Dit zal API.AI gebruiken om de woorden die we nodig hebben te herkennen.

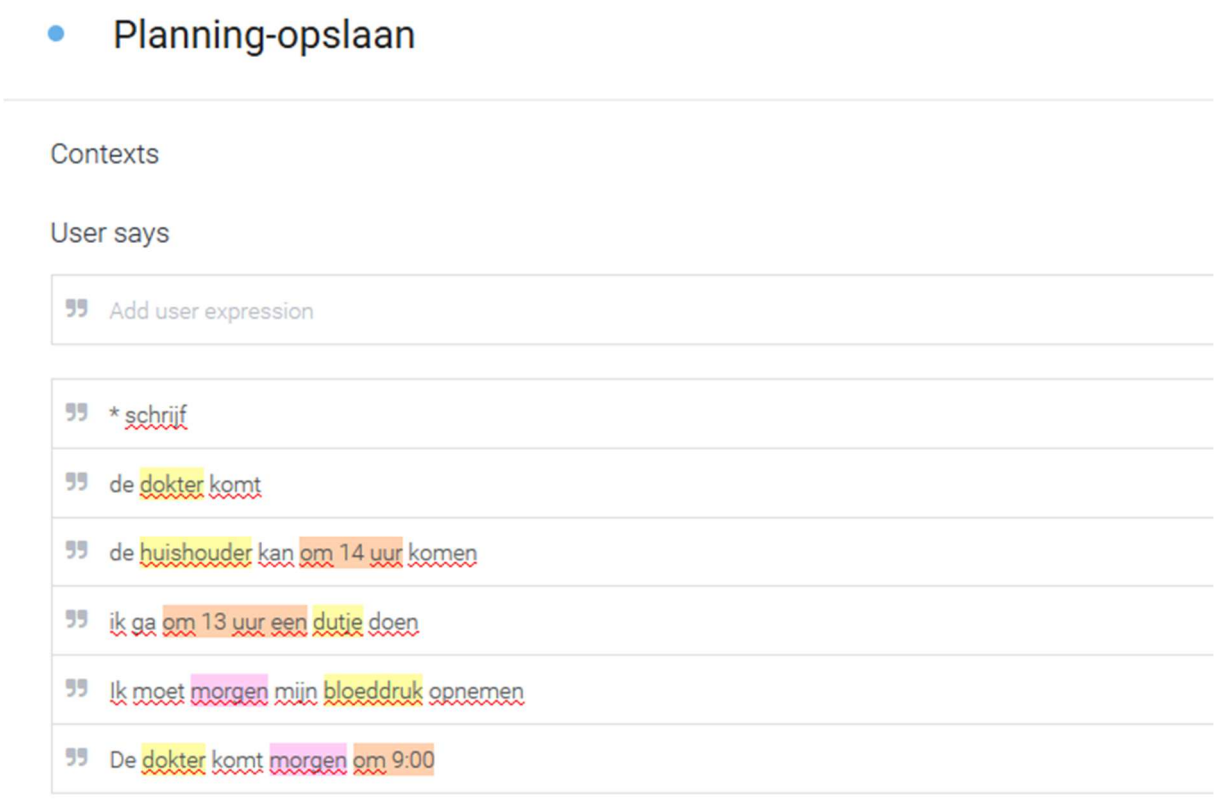


We gaan een "Activiteiten" entiteit definiëren met mogelijke activiteiten die in het kader van ons project in een agenda zouden kunnen staan. Voor elk activiteit kunnen we ook synoniemen definiëren.

Intents

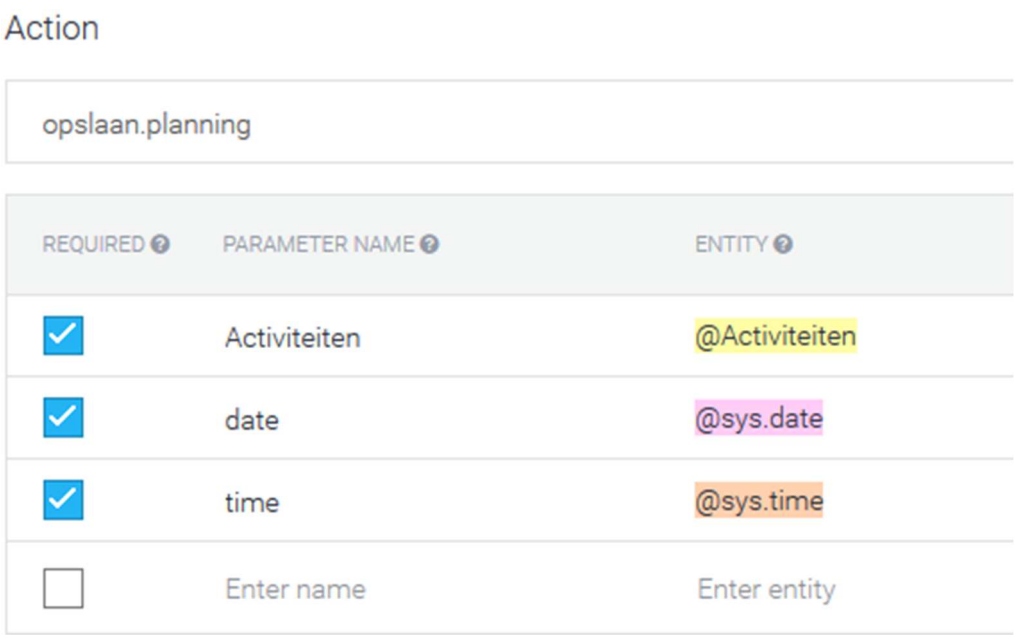
Hier gaan we definiëren wat API.AI moet doen en welke gegevens we nodig hebben. We gaan ook definiëren wat hij moet opnemen en hoe hij met de gebruiker moet dialogeren. Voor elke actie dat API.AI moet doen gaan we een intent aanmaken. Wanneer API.AI een woord herkent kan hij rechtstreeks naar de juiste intent komen.

Hier ziet u dat sommige intents al default gemaakt zijn door API.AI. De Fallback Intent zal gebruikt worden om de gebruiker te informeren dat API.AI de vraag niet heeft begrepen. Een andere intent werd aangemaakt om de gebruiker te verwelkomen.

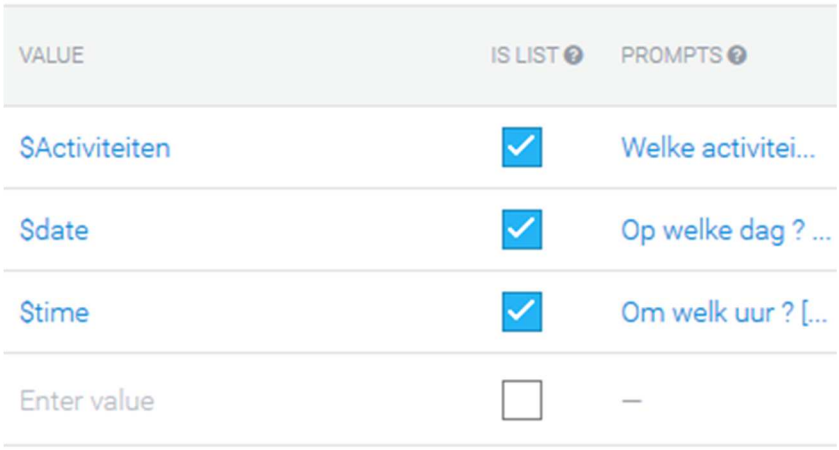


We gaan dus een "Planning-Opslaan" intent aanmaken waar we API.AI gaan aanleren hoe hij een planning moet opslaan en welke informatie de gebruiker hem moet laten weten om dit te kunnen doen.

We gaan beginnen door sommige mogelijke zinnen te tippen dat een gebruiker zou kunnen zeggen. We zien dat API.AI rechtstreeks verschillende woorden herkent. Hij herkent ons activiteit dat we in ons entities hebben gedefinieerd. Hij ziet ook dat er sprake is van tijd wat API.AI default herkent. Hij ziet ook het dag als default entitie.

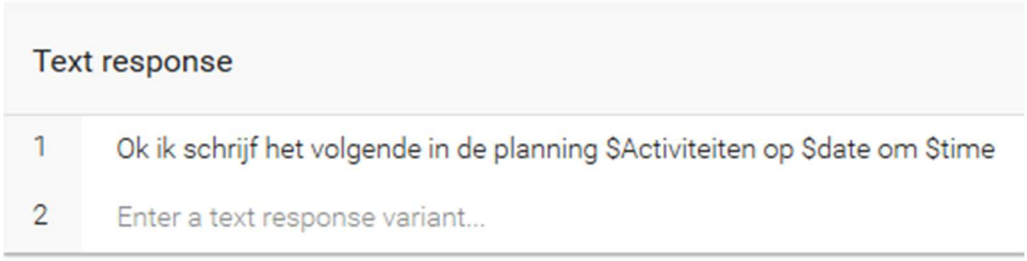


Daarna gaan we aan API.AI vertellen welke actie hij moet uitvoeren. We zien hier dat de drie nodige entries die we nodig hebben om iets in ons agenda op te slaan gedefinieerd zijn. We gaan aanvinken dat de taak niet uitgevoerd kan worden als we niet alle drie entries hebben. Dit doen we door ze aan te vinken als required.



In "Prompts" gaan we definiëren wat we aan de gebruiker gaan vragen als hij de nodige informatie nog niet bekend gemaakt heeft. Dit doen we voor de drie entries dat we nodig hebben.

Hier is wat we bijvoorbeeld voor het "Activiteiten"-entiteit kunnen schrijven.



Wanneer we alle informatie gekregen hebben gaan we uiteraard een antwoord definiëren. We gaan ook dezelfde woorden gebruiken dan de gebruiker. Dit doen we met het "$entities".

Uitvoer

We gaan nu ons app uittesten. Hier enkele schermshots van het gesprek.

schrijf in mijn agenda

Please use test console above to try a sentence.

Agent

USER SAYS COPY CURL

schrijf in mijn agenda

RESPONSE PLAY

Welke activiteit wilt u opschrijven?

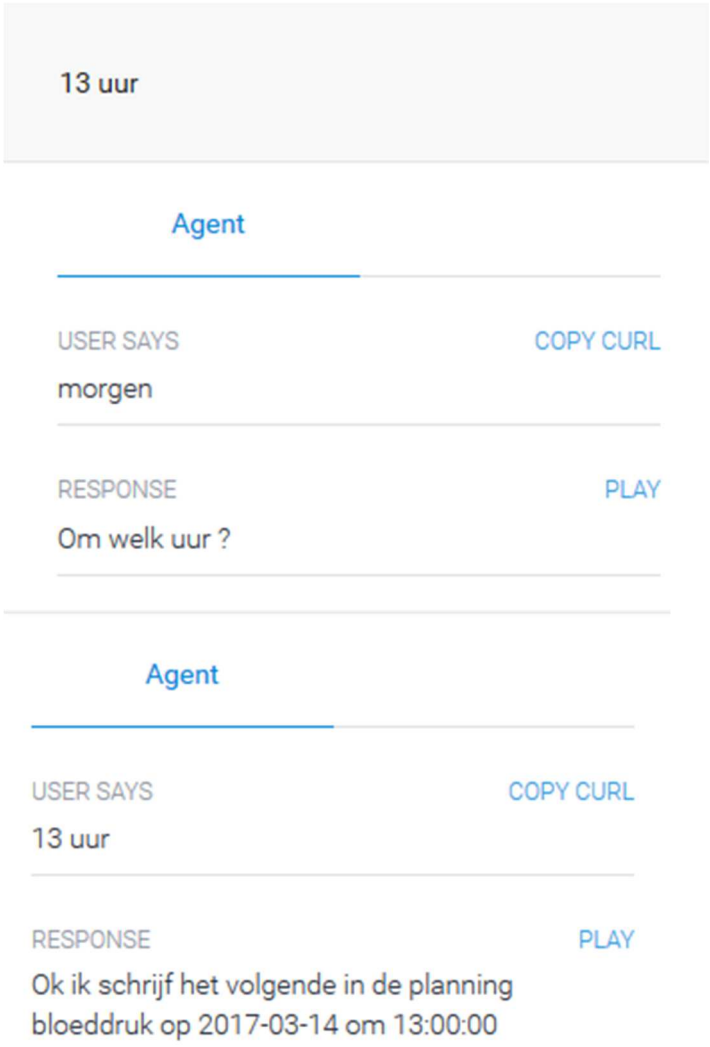
Agent

USER SAYS COPY CURL

bloeddruk opnemen

RESPONSE PLAY

Op welke dag ?



Na het conversatie krijgen we een JSON-file met alle gegevens van de conversatie. Dit is uiteindelijk de file die we gaan gebruiken bij de implementatie.

API.AI en Android.

API.AI heeft een Android SDK (Sofware devlopment kit) beschikbaar gesteld. De SDK is gemakkelijk implementeerbaar en is goed gedocumenteerd. Toch is er een minpunt. Zoals eerder gezegd is API.AI beschikbaar in verschillende talen. Maar hun SDK dat spraaktechnologie toelaat werkt niet in het Nederlands. Als we via het spraaktool van het SDK iets in het Nederlands mondeling meedelen herkend het programma default Engels en dus niet wat we gezegd hebben. Uiteraard als we de opdracht via het typen doorsturen werkt het wel. We hebben dus beslist dat we de spraak naar tekst omvorming native via Android gaan uitvoeren en niet via het SDK. We gaan daarna de omgevormde tekst via het SDK doorsturen naar API.AI. We hebben het proces uitgetest om de betrouwbaarheid hiervan te beproeven.

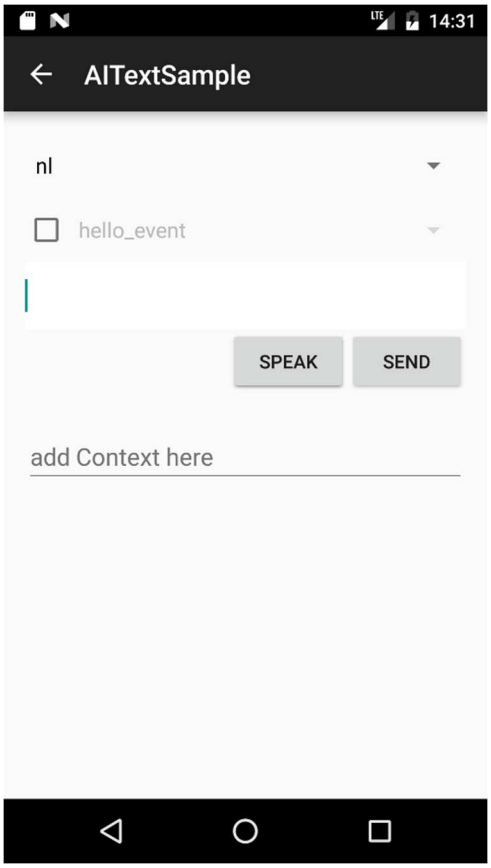
Uitwerking in Android

Voor het Androidapplicatie gaan we het testapplicatie gebruiken dat API.AI ter beschikking stelt. We gaan het bewerken om te zien als het werkt volgens onze eisen.

```
public static final LanguageConfig[] languages = new LanguageConfig[]{
        new LanguageConfig("nl", "8a038fe0ebc94ad0869624c0d034b8f2"),
        new LanguageConfig("en", "a11ea1d839e3446d84e402cb97cdadfb"),
        new LanguageConfig("ru", "c8acebfbeeaa42ccb986e30573509055"),
        new LanguageConfig("de", "ae2afb2dfd3f4a02bb0da9dd32b78ff6"),
        new LanguageConfig("pt", "b27372e24ee44db48df4dccbd57ea021"),
        new LanguageConfig("pt-BR", "a4e08b5bc87a41098237e3f23a5e1351"),
        new LanguageConfig("es", "49be4c10b6a543dfb41d49d88731bd49"),
        new LanguageConfig("fr", "62161233bc094a75b3acfe16aeeed203"),
        new LanguageConfig("it", "57f80c9c9a2b4e0eae1739349a46e342"),
        new LanguageConfig("ja", "b92617a3f82e4b52b3db44436d2d4b8b"),
        new LanguageConfig("ko", "447a595234d74561a76b669a88ab3d99"),
        new LanguageConfig("zh-CN", "52d2b2bd992749409fc3a7d0605c3db4"),
        new LanguageConfig("zh-HK", "760c7a5efe5d43b9a90d62f73251de6a"),
        new LanguageConfig("zh-TW", "9cadea114425436cbaeaa504ea56555b")
};
```

Eerst gaan we aan het applicatie moeten aanduiden dat ons agent in het nederlands gemaakt werd. Dit doen we door een lijntje bij te voegen in het "LanguageConfig" methode. We gaan ook gaan definiëren dat we de "Agenda" agent gaan gebruiken. Het Token van de agent gaan we dus in het methode plaatsen.

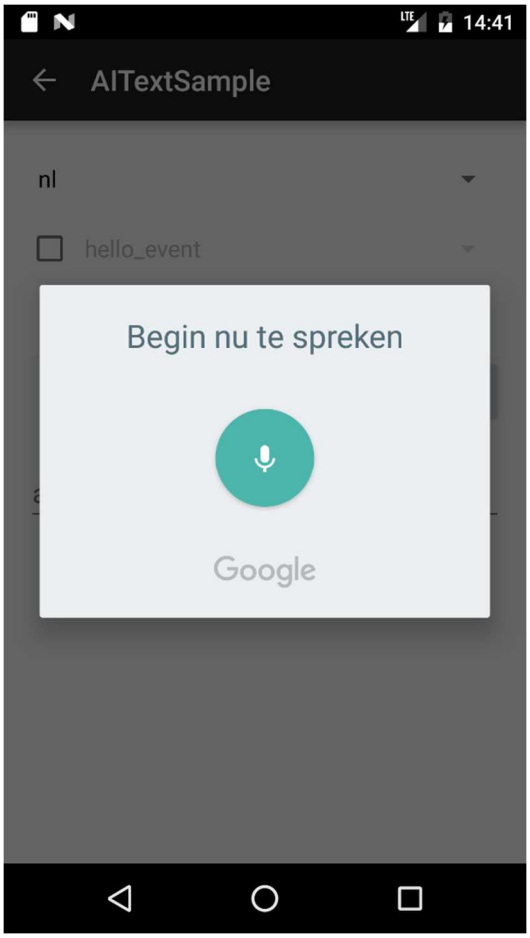
Het applicatie zelf werkt als volgt. We gaan een "speak" button aanmaken.

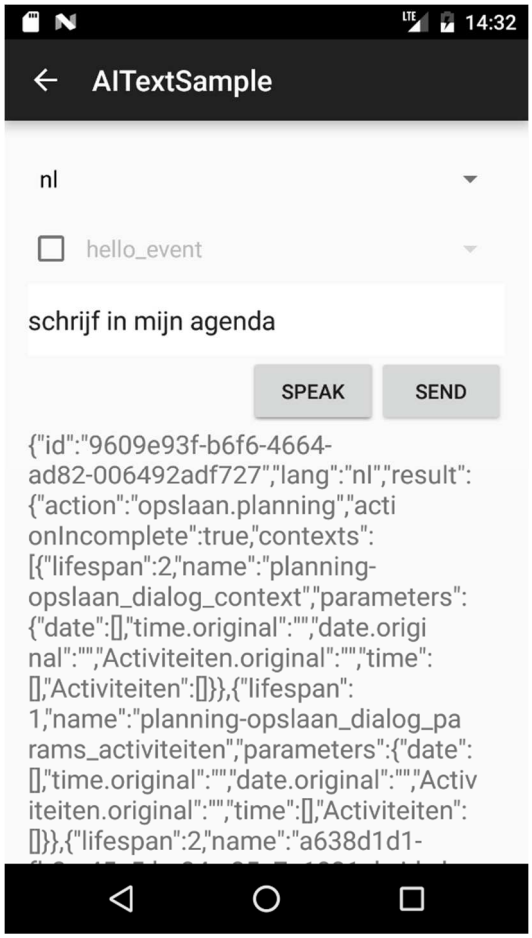



```
// Create an intent that can start the Speech Recognizer activity
@TargetApi(Build.VERSION_CODES.LOLLIPOP)
private void displaySpeechRecognizer() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.forLanguageTag("nl-BE"));
    try{
        startActivityForResult(intent,200);
    }catch (ActivityNotFoundException a){
        Toast.makeText(getApplicationContext(),"Intent problem", Toast.LENGTH_SHORT).show();
    }
}
```

Hier ziet u het "SpeechRecogniszer" methode we gaan aanduiden dat het in Nederlands moet werken. Het is ook heel belangrijk dat het device waarop de applicatie getest wordt in het Nederlands ingesteld is.

Wanneer we op de button gaan drukken krijgen we een scherm te zien dat ons de mogelijkheid laat om iets in te spreken. Daarna krijgen wanner we op de "send" button drukken krijgen we een vocale antwoord we zien ook de gegevens dat we terug krijgen van het agent.





We kunnen hier zien dat het applicatie de juiste intent heeft aangesproken. Het antwoord van de applicatie is "Op welke dag?" en werd door de luidsprekers gegeven. De integratie in Android is dus goed voorlopen.

Api.ai heeft zoals eerder gezegd een goed integratie met Google en meer specifiek met "Action On Google" (niet te verwarren met Google Voice Action) dat ons de mogelijkheid geeft om applicaties te maken voor apparaten zoals de Google Home. Google maakt een emulator beschikbaar om het te testen via API.AI. Het emulator is alleen beschikbaar in het Engels en we gaan dus een scenario in het Engels uitvoeren om het te kunnen testen.

Tweede scenario

Het tweede scenario is dat een user een commentaar wilt schrijven op een activiteit van een bepaalde dag.

De gegevens die we moeten opslaan zijn de volgende.

1. Een activiteit waarop de user een commentaar wilt geven.
2. Het dag waarop het plaats heeft gevonden activiteit.
3. Het commentaar dat de user wilt opslaan.

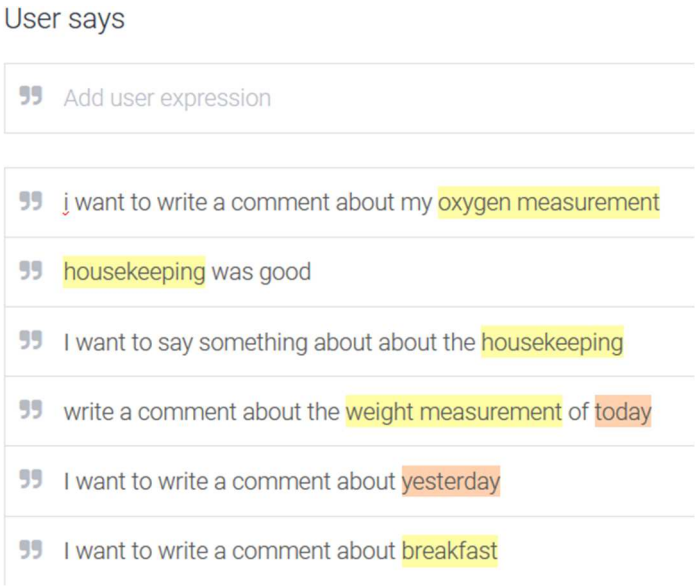
Uitwerking

We gaan eerst een agent aanmaken en het in het Engels instellen. Vervolgens gaan een "activities" entiteit aanmaken zoals voor het vorige test applicatie maar deze keer in het Engels. Hier een lijst van activities.

Morning Snack	Morning Snack
Breakfast	breakfast
Laundy pick up	Laundy pick up, Laundry
Housekeeping	Housekeeping, Housekeeper
Dressing	Dressing
Weight Measurement	Weight Measurement
Dinner	Dinner
Temperature Measurement	Temperature Measurement, Temperature
Recreation	Recreation
Oxygen Saturation Measurement	Oxygen Saturation Measurement, Oxygen Measurment
oxygen measurement	oxygen measurement
	Click here to edit entry

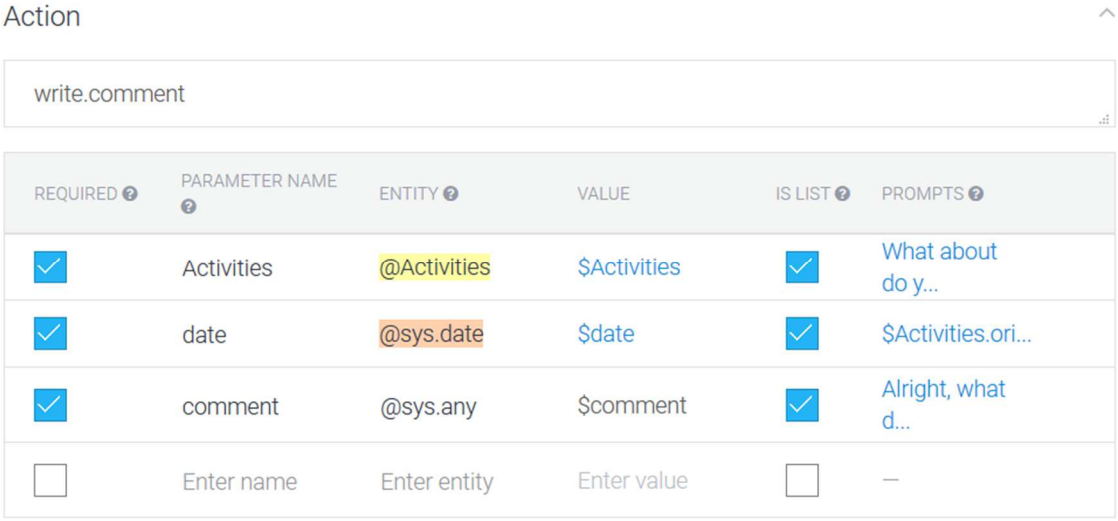
Intents

We gaan nu een WriteComment intent aanmaken. We gaan zoals het vorige intent die aangemaakt hebben eerst de mogelijke zinnen opschrijven dat een gebruiker zou kunnen zeggen.



Actions

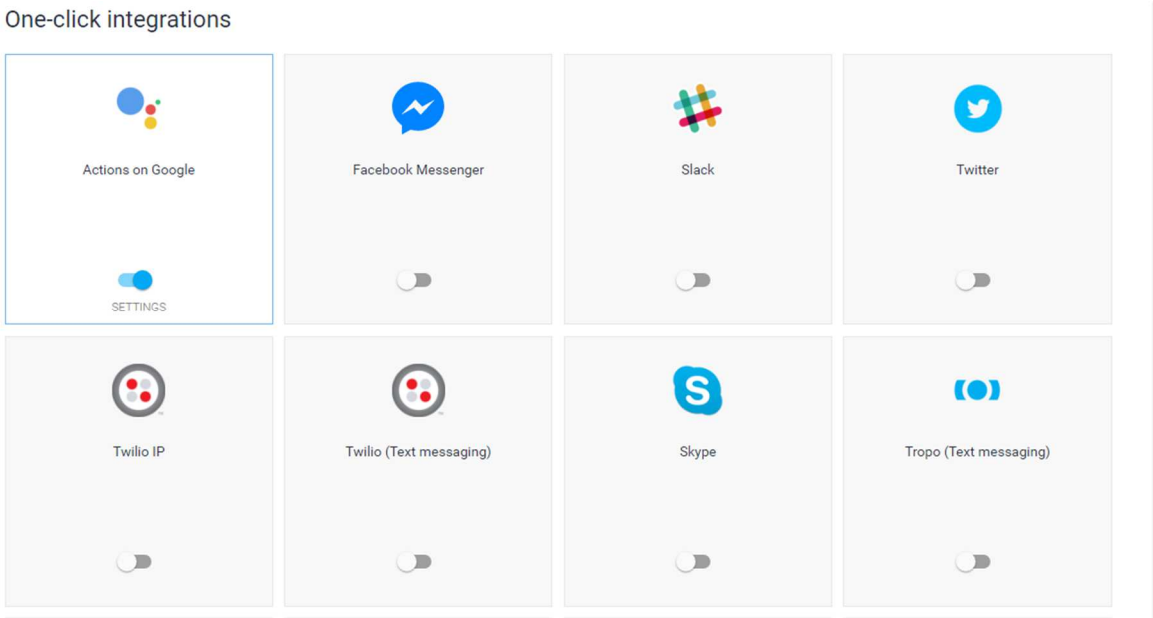
Bij het aanmaken van de action hadden we een probleem ondervonden. Een commentaar kan niet als entiteit gedefinieerd worden omdat elke commentaar anders kan zijn. Een gebruiker moet de mogelijkheid hebben om eender wat als commentaar te schrijven. We gaan dus een default system entiteit in het documentatie moeten zoeken dat dit toelaat. Na opzoeking hebben we ondervonden dat het default system entiteit "sys.any" dit toe laat.



We gaan hier terug via prompts de nodige vragen stellen aan de gebruiker om alle gegevens te hebben.

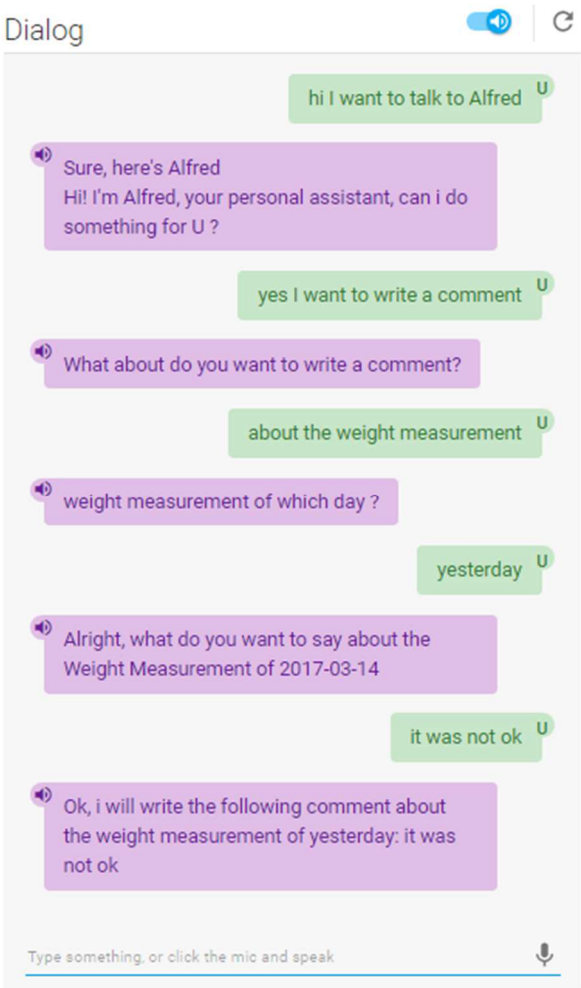
En via response een uitvoer een confirmatie geven dat het commentaar opgeschreven werd.

We gaan daarna het API.AI applicatie aan Google On Action integreren via het Integration tabblad.



We moeten een naam geven aan ons applicatie voor dat we het integreren. Het is de naam dat Actions On Google gaat herkennen om ons app te kunnen openen. We hebben het voor de test Alfred als naam gegeven. We gaan nu ons applicatie kunnen testen.

Op de afbeelding hiernaast ziet u een voorbeeld conversatie van het programma. We gaan eerst vragen om aan "Alfred" te spreken. Daarna wordt rechtstreeks de default "Welkom" intent afgevuurd. We hadden het ingesteld om dit als begroeting tekst te krijgen. vervolgens gaan we ons applicatie vragen om een commentaar te schrijven. Alfred vraagt ons dan om hem de nodige gegevens te geven. Uiteindelijk geeft hij ons een antwoord met confirmatie dat hij de commentaar zal schrijven op het gevraagde dag en het gevraagde activiteit. Goed om aan te duiden dat we niets moesten typen en dat Alfred altijd sprekend antwoorde. We zien hier dat Actions On Google goed begrijpt wat we zeggen en goed communiceert mes API.AI. We krijgen ook het JSON-file met heel de conversatie te zien. Naast het "Dialog"-Venster.



Conclusie en advies voor het project.

Pro's	Contra's
+ Gratis	**- Spraakherkenner in Android SDK niet beschikbaar in alle talen.**
+ Goed gedocumenteerd	**- Nieuwe features alleen beschikbaar in het Engels**
+ 15 talen ondersteund	
+ Spraaktechnologie is stabiel	
+ Multiplatform	
+ Volledig customiseerbaar	

API.AI is een stabiele technologie dat zeer goed gedocumenteerd is. Hun systeem is heel intuïtief en ondersteund 15 talen inclusief Frans en Nederlands, wat voor ons heel interessant is. We kunnen het AI-agent volledig aanmaken en verwerken volgens ons eisen. Het antwoord goed op spraak en herkent ook goed de woorden die we uitspreken. Voor de implementatie is het goed voor alle talen. Alleen is het niet volledig mogelijk om hun spraak-oplossing via hun SDK te gebruiken in het Nederlands. Maar het spraakdetectie kunnen we via de Androidmethodes aanmaken. API.AI is een technologie dat volledig aan ons eisen antwoord en is een goed optie voor ons project.

Bijlage IX: Design

Door: Koumeyl Belkhidar

Design

Introductie

Het volgende fase na de prepare fase is de design fase. In dit fase werd er een compleet technisch ontwerp gemaakt van de applicatie. In eerste instantie werd een analyse gemaakt over hoe oudere personen met smartphones omgaan. Dit op het psychologisch en fysisch vlak. Daarna werden er verschillende mockup's gemaakt. Een mockup is een ontwerp dat niet definitief is maar dat een soort schets en geraamte wordt waarop het definitief design op gebaseerd zal worden. Ten laatste werd er een definitief design gemaakt dat aantoont hoe het applicatie definitief eruit zal zien.

Oudere personen en mobiele applicaties.

Voordat er met design begonnen werd was het belangrijk om te weten hoe het doelpubliek met mobiele applicaties kunnen omgaan. Het was essentieel om na te komen hoe het app eruit moest zien op het vlak van kleur, fonttype, fontgrote, hoeveelheid componenten op het scherm enz. Ook hoe ze op het psychologisch vlak met een app omgaan was belangrijk om te weten.

Na verschillende opzoekingen kwamen we terecht op een artikel geschreven door een firma dat een applicatie voor oudere personen gemaakt hebben namelijk KoalaPhone[2]. Het artikel geeft belangrijke punten waarop er opgelet moet worden bij het maken van een applicatie.

Kleur

Oudere mensen hebben met de tijd een verminderd zicht. Hun ooglens ontwikkeld ook een licht geelachtig kleur. Het is daardoor voor hun moeilijk om het onderscheid te maken tussen tinten van blauw en paars.

Geluid

Omdat het mens met de tijd een vermindert gehoor kan ontwikkelen is het aangeraden om een geluidswaarde van 90 db te hebben als er met geluid gewerkt wordt.

[2] https://medium.com/@tomasslavicek/designing-a-mobile-interface-for-older-people-1c9b70fd645c

Voordeel

Het is niet waar dat oudere mensen geen nieuwe technologieën willen leren of niet ermee willen omgaan. Een bejaarde persoon kan een compleet nieuw technologie leren alleen moet het persoon het nuttig vinden. Het moet het moeite waard zijn en een oudere personen moeten een voordeel erin vinden. Het is dus belangrijk in het project om nuttige features te ontwikkelen.

Interface

De basis navigatie knoppen moeten altijd zichtbaar zijn. Ze moeten ook elke keer terugkomen en het manier van navigeren over het app moet consistent blijven over heel het interface.

Iconen

Alle iconen moeten tekst bevatten over wat het effectueert. Dit omdat een oudere persoon zeker wil zijn bij het drukken van een knop.

Items

Het is belangrijk om een gelimiteerde hoeveelheid items te hebben per scherm. Zodat het persoon niet verward wordt door het aantal mogelijkheden. De app zou in geval van meer items moeilijk worden en niet overzichtelijk genoeg zijn. Een scherm of een lijst zou maximum 7 tot 8 items moeten bevatten.

Font

Wat het font betreft gaf het artikel geen informatie. Het nochtans een belangrijk onderdeel van een app. Er werden dus opzoekingen gemaakt en hier is wat eruit komt.

Zoals eerder gezegd hebben oudere personen een verminderd zicht maar hoe wordt het verminderd? Paul Nini van AIGA ("the professional association for design")[3] legt uit dat oudere personen op verschillende vlakken hun zicht verminderd krijgen.

[3] http://www.aiga.org/typography-and-the-aging-eye

Verlies van licht

Afbeelding 1.1: zicht doorheen de jaren.

Op het afbeelding 1.1 zien we het verschil van licht dat door de ooglens doorgaat. We zien hier het zicht van een mens dat 20 jaar oud is (links), 60 jaar oud is (midden) en 75 jaar oud is (rechts). Bij het ontwikkelen van het app ga het belangrijk zijn om verschillende kleuren te gebruiken en de kleuren goed te differentiëren.

Verlies van focus

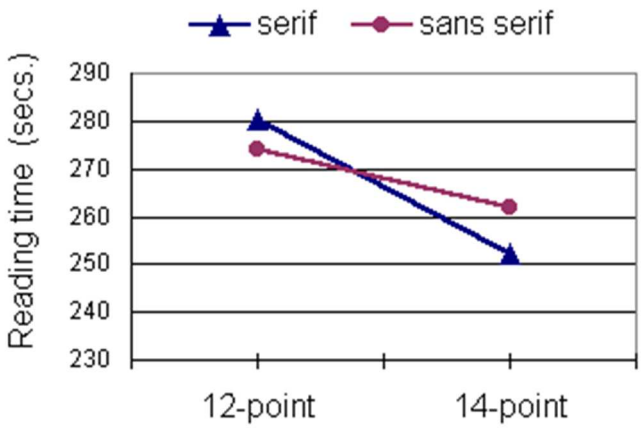
Dit constateren we al bij een mens van 40-50 jaar oud. De focus verminderd en begint wazig te worden. Dit is veel bij het lezen het geval. Op de afbeelding hiernaast ziet u het verschil tussen een wazig zicht en een normaal zicht.

Afbeelding 1.2: verschil tussen normaal zicht en wazig zicht.

In ons context is dit belangrijk om te weten vanaf het moment dat we een font moeten kiezen met een gepaste fontgrote, fonttype en kleur. We gaan dus hier bij het kiezen van het fonttype ons focussen op een type waar de letters een minimum gespreid zijn. De vorm van de letters is hier ook belangrijk. We moeten een goed onderscheid kunnen maken tussen een "a" en een "e" bijvoorbeeld.

Afbeelding 1.3: Voorbeeld van het zicht bij het lezen va twee verschillende lettertypes.

In de fontypes maken we een onderscheid tussen twee categorieën. De "serif" en "sans serif". Wat de twee fonts verschillend hebben is dat de "serif" fonts meer floraal en gekruld zijn. Er werd een studie gemaakt[4] dat aantoont dat oudere mensen vlugger lezen als het font "serif" is. Ook de fontgrote werd getest en oudere mensen lezen beter als het font groter is. Android gebruikt Roboto als lettertype. Roboto is een "serif" lettertype dit kan dus zonder probleem gebruikt worden.



Afbeelding 1.4: Tijd nodig om een passage te lezen in seconden.

[4] http://usabilitynews.org/determining-the-best-online-font-for-older-adults/

Mockuping

Dankzij het analyse kan het systeem ontwerpt worden op een correcte manier. Er werden begonnen met mockups aan te maken om een beeld te hebben van de applicatie. Dit beeld geeft het mogelijkheid om aan het bedrijf concrete voorbeelden te tonen. Er werd dus aan het bedrijf verschillende mogelijkheden voorstelt waarop we met het bedrijf konden discussiëren om samen tot het beste oplossing te komen en om de eisen van het bedrijf zo goed mogelijk te vervullen.

Het tool dat hiervoor gebruikt werd is "Balsamiq mockup". Een bekende tool voor het aanmaken van mockups dat ook de mogelijkheid geeft om te wireframen. Een wireframe is een snelle schets waar we in grote lijnen kunnen zien hoe het applicatie eruit zal zien. Niet te verwarren met mockups dat meer gedetailleerd zijn en waar we de finale functionaliteiten kunnen bezien maar op een statisch vorm. Balsamiq mokups geeft ook de mogelijkheid om op de aangemaakte schermen te navigeren. Dit om al een idee te hebben van de gebruikerservaring van de applicatie.

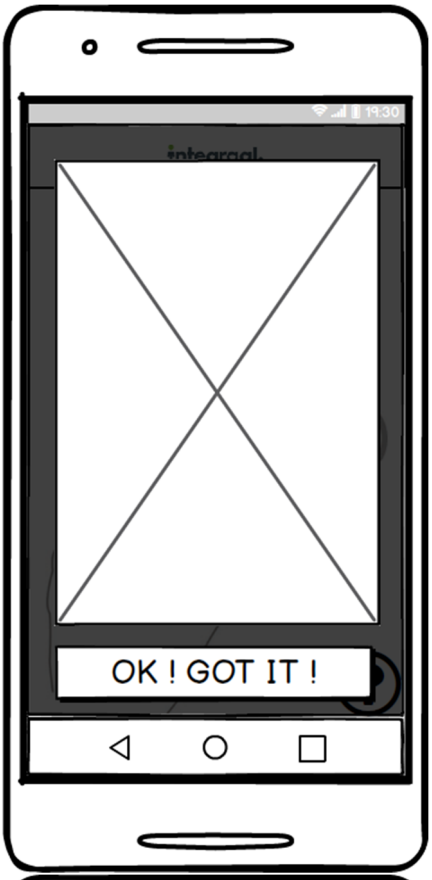
Structuur van het applicatie

1. Startscherm

Dit is het eerste scherm da we te zien krijgen bij het lanceren van het app. Er werd hier een het logo van het bedrijf op gezet. Dit scherm is een splashscherm waarop het applicatie geladen wordt bij het lanceren.

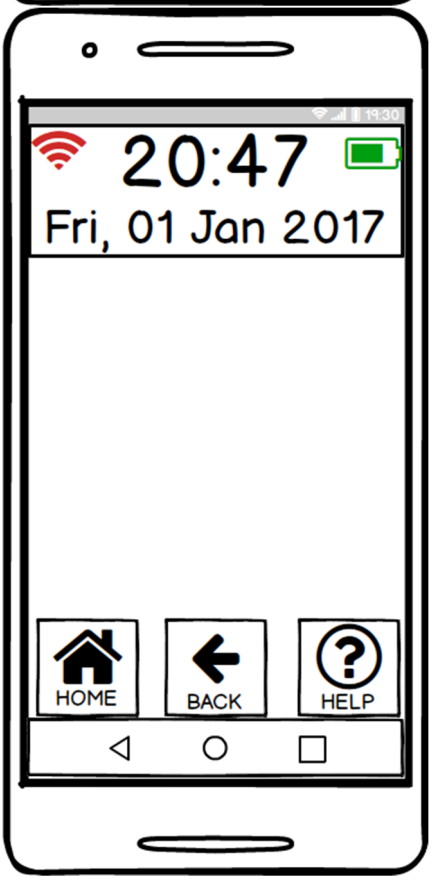
2. Loginscherm

Het volgende scherm is het loginscherm. We zien hier dat het gebruiker zich met een NFC kaart zal moeten inloggen. Op het NFC kaart zal dan een unique code staan dat het gebruiker dat gewoon gaat moeten scannen. Dit werd zo bedacht omdat het doelpubliek zo minder mogelijk moet typen in het programma. Het klavier dat Android aanbiedt is niet gebruikersvriendelijk en u zult zien naargelang het verloop van de scherm's dat we hiervoor oplossingen hebben bedacht. We zien ook dat er een help button gedesigneerd werd. Dit button zal op elke scherm beschikbaar zijn. En gaat het helpscherm openen.



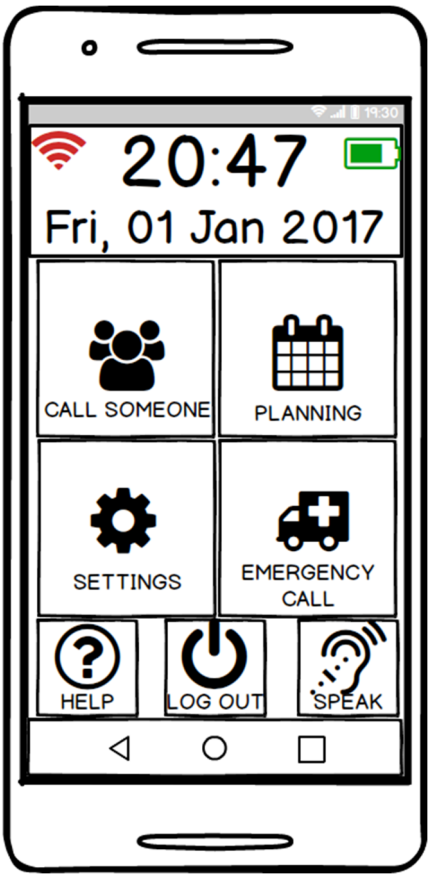
3. Helpscherm

Dit pop-up-scherm komt tevoorschijn wanneer we op het hulpbutton drukken. Het scherm bevat hulp over wat verwacht wordt op het scherm. Het kan een video zijn, instructies enz. Bijvoorbeeld als we op de hulpbutton van het login scherm drukken. Krijgen we een video te zien over hoe we in moeten loggen.



4. Basescherm

Er werd daarna een basescherm gedesignd. Dit scherm is geen effectief scherm van het programma maar een patroon waarop alle andere schermen op gebaseerd zullen worden. Er werd beslist om met een header en footer te werken dat op elke scherm beschikbaar zullen zijn. Op de header zien we de datum en tijd. We zien ook de status van de wifi en de batterij van het toestel. Zo is het batterij groen als het boven een bepaald percentage staat dan oranje en dan rood als het batterij bijna plat licht. Dit werkt op hetzelfde manier voor het wifisignaal. De footer zal zoals de header ook beschikbaar zijn op elke scherm. Hier zullen we de fundamentele knoppen vinden om gemakkelijk te kunnen navigeren doorheen het programma. Het is ook de bedoeling om elke keer dezelfde icoons te tonen en ze op dezelfde plaats te zetten.



5.Mainscherm

Dit is het menu van de applicatie waar de gebruiker naar de verschillende features kan navigeren. We zien hier verschillende knoppen. "Call somoene", "Planning", "Settings" en "Emergency call".

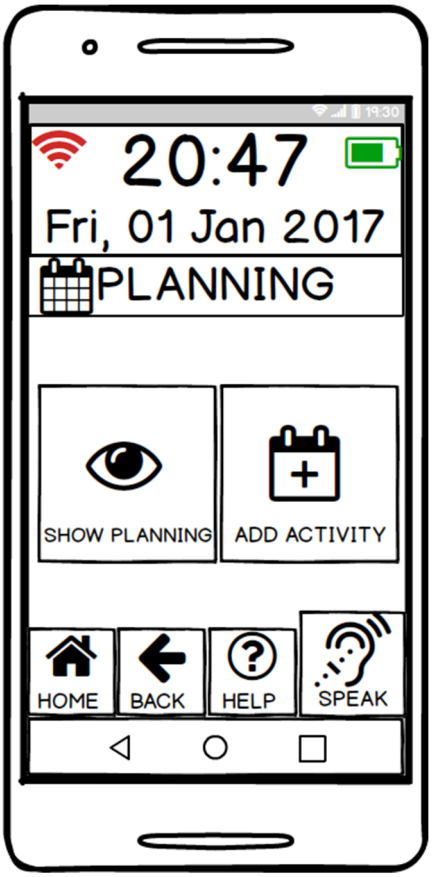
"Call somoene" geeft het mogelijkheid om een persoon dat zicht in zijn kring bevindt te bellen via videocall.

"Planning" geeft de gebruiker de mogelijkheid om de verschillende activiteiten dat hij gepland heeft op een door hem gekozen dag te raadplegen en hierop een notitie te schrijven. Het geeft ook de mogelijkheid om een activiteit toe te kunnen voegen.

"Settings" dient om verschillende componenten te managen zoals het ringtoon of het foto van hun profiel.

Tenslotte kunne we via het "Emergency call" button kunnen we een noodoproep afvuren.

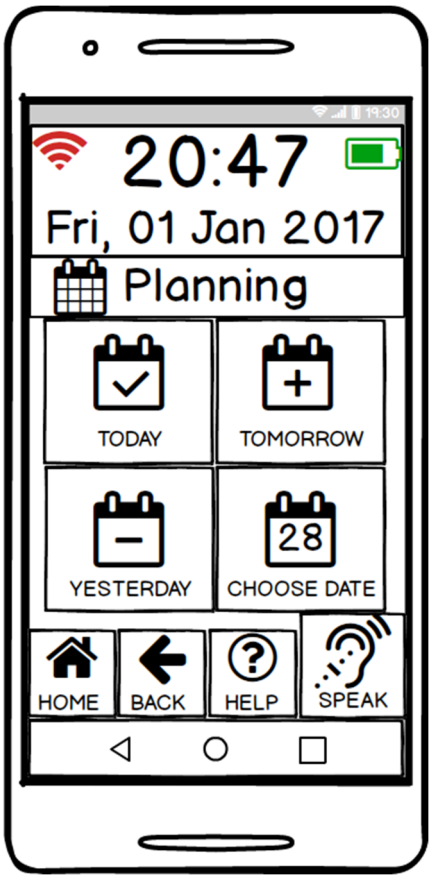
Hiernaast zien we in de footer de logout functie om uit te loggen en het speak functie die later uitgelegd zal worden.



6. Planning

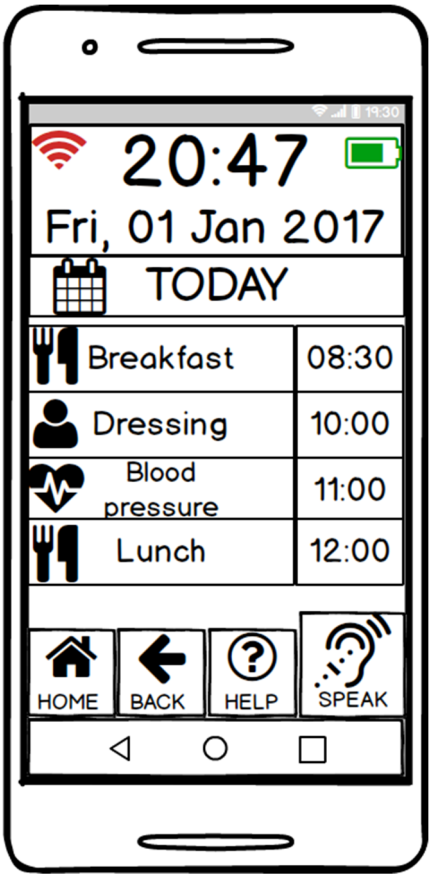
De gebruiker kan hier kiezen om zijn planning te bezien of een activiteit toe te voegen aan zijn planning. We kunnen hier ook zien dat we in de footer een back button en een home button. Home om naar de mainscherm te gaan en back om terug naar het vorige scherm te navigeren.

6.1. Show Planning



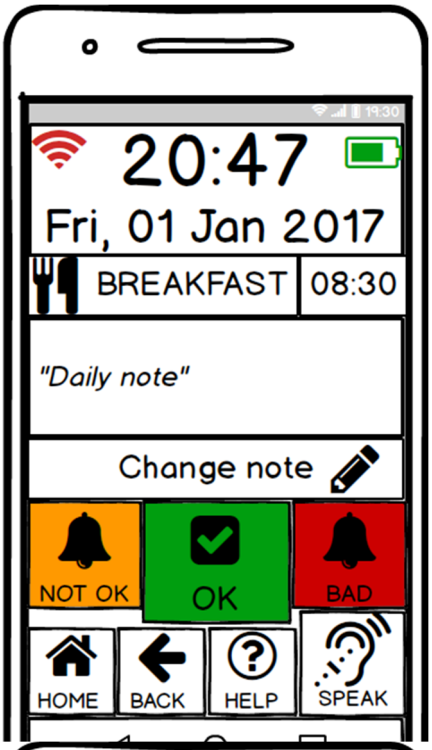
Hier kunnen we het datum kiezen waarop we on planning willen bezien. De 3 meest gebruikte dagen kunnen rechtstreeks geraadpleegd worden via hun respectievelijke knop. De gebruiker kan ook via een gekozen datum zijn planning raadplegen.

6.1.1. Show Planning – Today



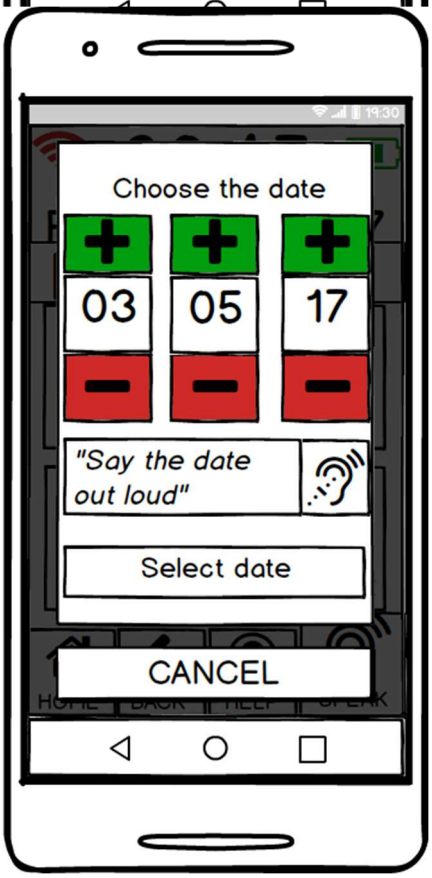
Hier zien we het planning van de dag. We krijgen het activiteit te zien met een icon dat rechtstreeks toont welk soort activiteit het is. We krijgen ook het uur te zien waarop het activiteit plaats zal vinden. We kunnen op de activiteit drukken dat een andere pagina opent.

6.1.1.1. Show Planning – Today – Breakfast

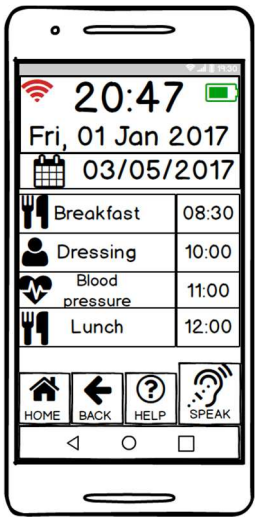


Als het gebruiker op een activiteit wordt er het volgende scherm getoond. Hier kan er een noot gesaved worden om te zeggen hoe het activiteit verlopen is. We kunnen ook een op 3 knoppen drukken. Als we op een oranje en rode button drukken kunnen we een alarm afvuren. Oranje staat voor: het is niet goed verlopen. De verzorgers krijgen het waarschuwing een week te zien. En het rode knop staat voor: het is zeer slecht verlopen en de verzorgers moeten het rechtstreeks raadplegen en regelen. Het ok button staat voor: het is goed verlopen.

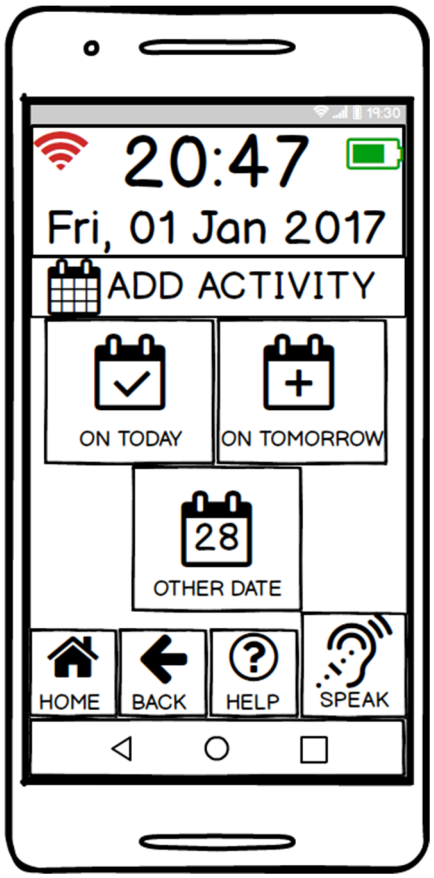
6.1.2. Show Planning – Choose date



Hier kunnen we het datum kiezen waarop we ons planning willen bezien. We kunnen het datum kiezen via de plus en min knoppen design en aanvaard door het bedrijf. We kunnen ook ons datum vocaal inspreken. Dit weer met de optiek om zo weinig mogelijk tikwerk te geven aan het gebruiker. Als we de datum kiezen krijgen we de datum tevoorschijn op de showplanning scherm (zie afbeelding rechts).

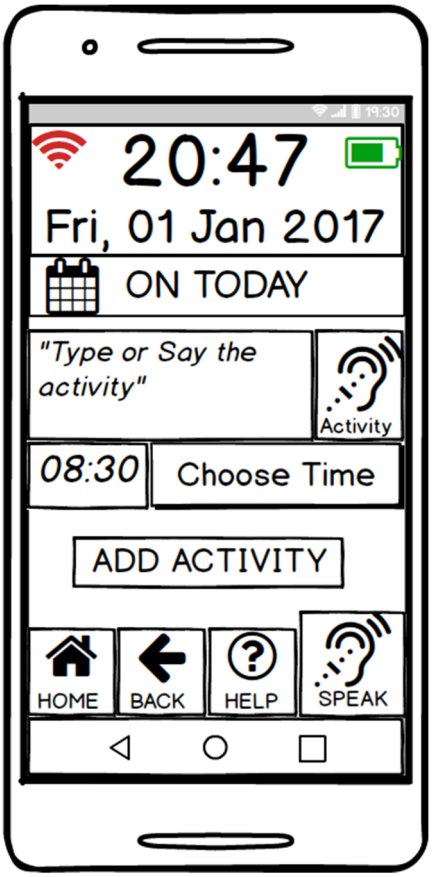


6.2. Add Activity

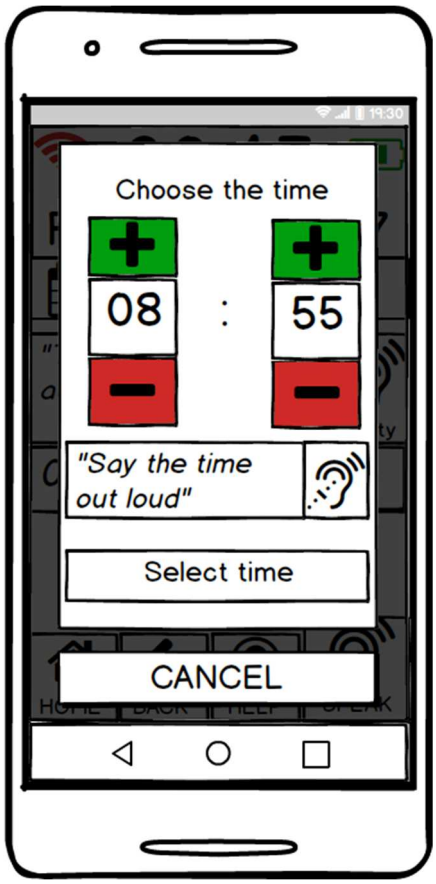


Op dezelfde manier dan showplanning kan de gebruiker hier ook de dag kiezen waarop een activiteit toegevoegd moet worden.

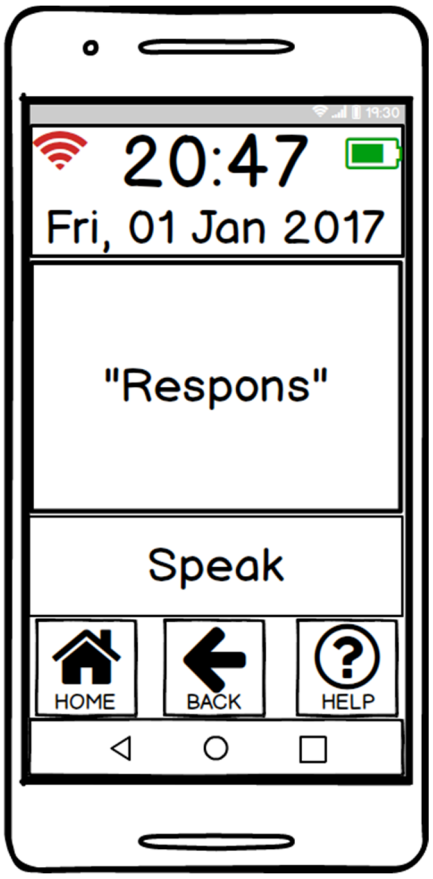
6.2.1. Add Activity – on today



Hier kan de gebruiker het activiteit toevoegen en het tijd (afbeelding recht) toevoegen. Het kan tekstueel of op een vocale manier. Wanneer alle velden ingevuld zijn kan het doorgestuurd worden.



7. Speakscherm

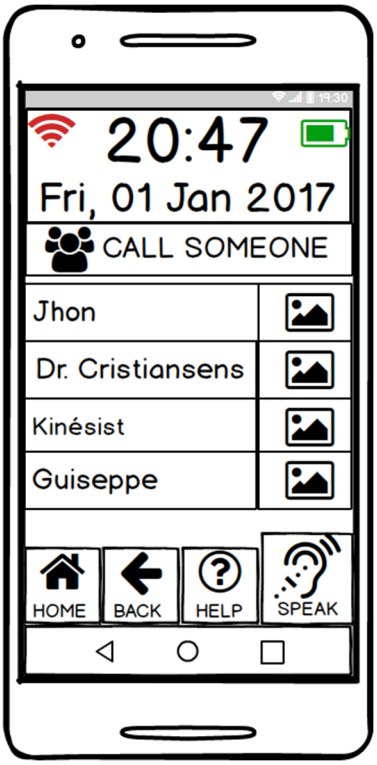


Op elk scherm is een er een speak button beschikbaar. Dit button kan alle features die we gezien hebben vocaal uitvoeren. Een persoon moet op het speak button drukken en aan het systeem vragen wat hij wilt doen. Uiteindelijk wordt er een dialoog gemaakt met een artificiële Intelligence dat de nodige vragen stelt voor het uitvoeren van de opdracht. Het antwoord gebeurt vocaal en wordt ook op het scherm getoond. Een gebruiker kan bijvoorbeeld aan het systeem vocaal vragen om een activiteit of een notitie toe te voegen. De gebruiker kan hier dus taken uitvoeren zonder te navigeren doorheen de applicatie. Er werd aan het bedrijf voorgesteld om ook de navigatie door de app via dit knop te doen. Maar het bedrijf heeft besloten dat het geen goed idée zou zijn.

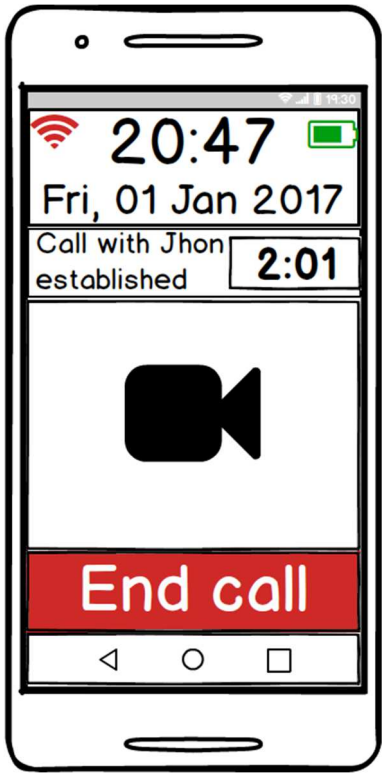
8. *Emergency scherm*

Wanneer een gebruiker op het emergency knop duwt wordt er een sms afgevuurd naar gekozen persoon dat op voorhand in het systeem toegevoerd werd. We krijgen hier 10 seconden de tijd om het alert te stoppen. Na de 10 seconden wordt het sms doorgestuurd.

9. Call Somoene



Hier krijgen we twee ontwerpen van het Call Somoene scherm te zien het scherm dat is het bijgehouden werd scherm dat links staat. Dit omdat het overzichtelijker is. We krijgen hier het foto en het naam te zien van het persoon. Als men op het persoon drukt word er een call gemaakt naar het persoon het kan het call eindigen door op het End call button te drukken.



Finaal design

Nu dat de er een visie bestaat en dat de functionaliteiten in groote lijnen gedesigneerd werden. Gaan we nu beginnen aan het finaal design. Hier worden de grootte van de icoons de kleuren enz. definitief vastleggen.

[5]Het tool die dat hiervoor gebruikt werd is Adobe Experience design. Het is een tool gespecialiseerd in het ontwerpen van applicaties. Het tool is heel compleet.

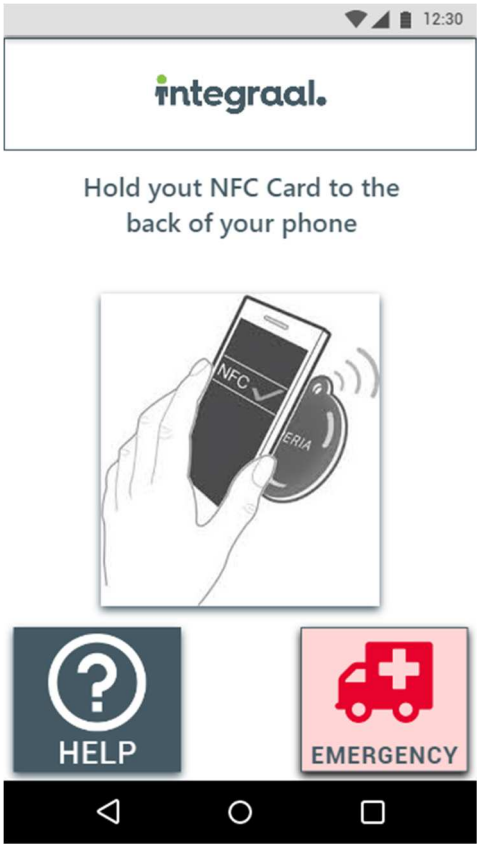
Structuur van het finaal design

De structuur is zoals eerder gezegd gebaseerd op de mockups toch zijn er verschillende onderdelen verander bij het final design.

[5] https://helpx.adobe.com/content/dam/help/mnemonics/xd_app_RGB.svg

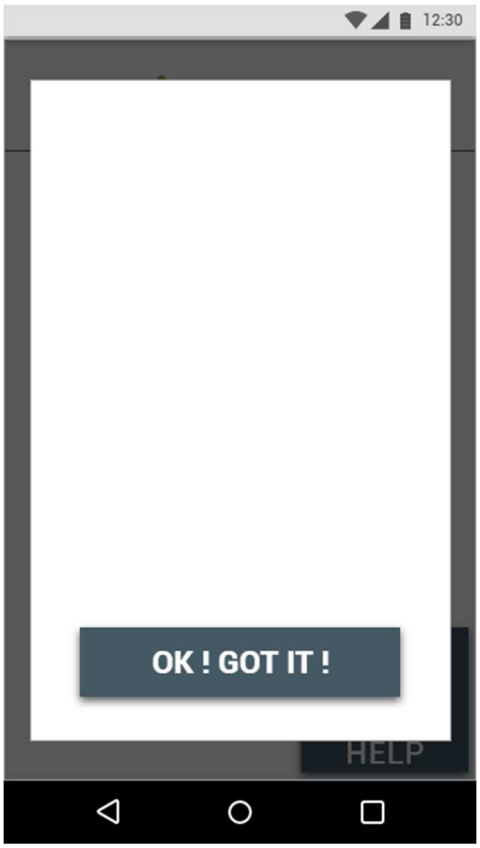
Splashscherm

Loginscherm

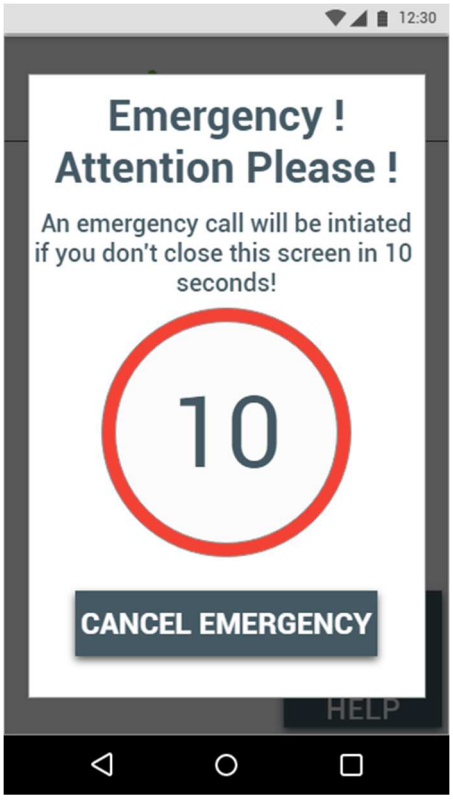


Dit is het finaal loginscherm. Bij het ontwerpen hiervan was het belangrijk om met de kleuren van Integraal te werken. Het hulp functie is van plaats veranderd omdat het hulpfunctie in de andere schermen altijd links staat. Er werd ook hier een Emergency button toegevoegd. Omdat het Emergency button altijd beschikbaar moet zijn. Voor de rest blijft het dezelfde als de mockups. Op de afbeeldingen heer beneden ziet u de verschillende schermen.

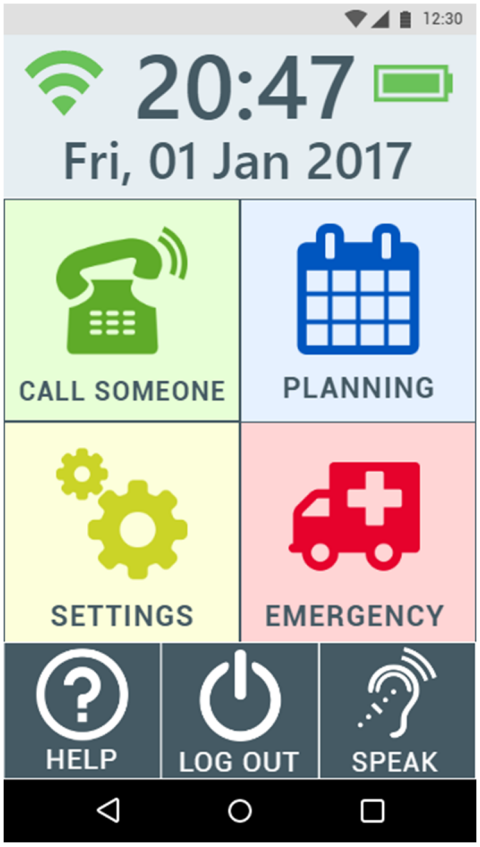
Help pop-up scherm



Emergency pop-up scherm



Main scherm



Dit is het main scherm van het applicatie geworden. Er werd geen ruimte tussen de button gelaten zodat men gemakkelijker op buttons kan drukken. We vinden hier de kleuren van het bedrijf terug in het lettertype. Elk app onderdeel heeft een verschillend kleur gekregen. Zo is het call functie groen. Het planning blauw de setting geel en de emergency in het rood. Dankzij het kleuren verdeling kunnen oudere mensen zonder het lezen weten waar ze terecht gekomen zijn in het applicatie.

De hoofding is ook verander en wifi en batterij hebben verschillende statussen. Op het volgende afbeelding zien we de verschillenden iconen van batterij en wifi.

Planning main

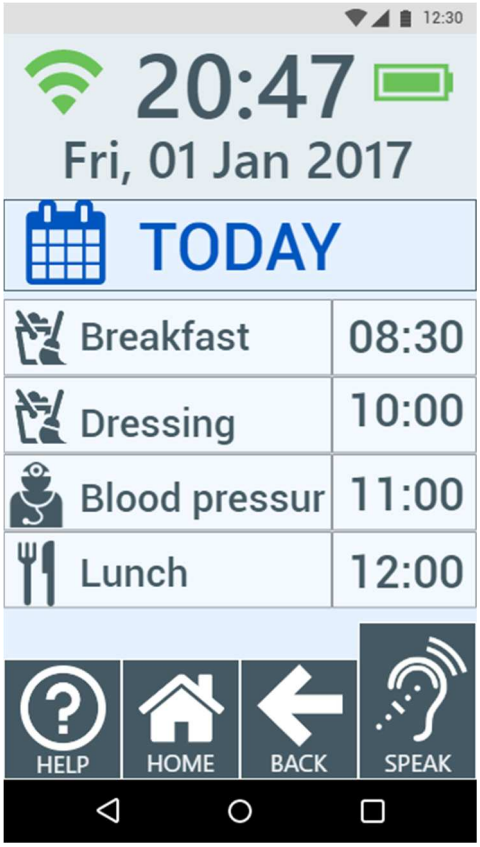
Het blauwe kleur is hier de thema van het scherm. Grote iconen werden gedesigneerd en elke keer wordt er op het icoon geschreven wat het doet. Wat de functionaliteiten betreft is er niets veranderd en is het concept dezelfde als het mockup gebleven.

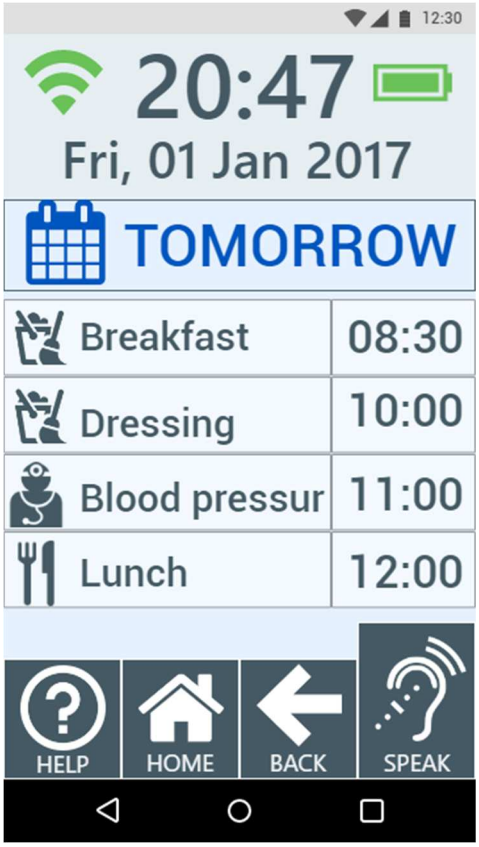
Show planning

Show planning – Today

Show planning – Tomorrow

We blijven in het blauwe thema. Hier blijft het functionaliteit in de logica van de mockups. Bij elk activiteit wordt er een icoon toegevoegd. Dit icoon wordt gegeven als een activiteit in een bepaald activiteit-categorie valt.



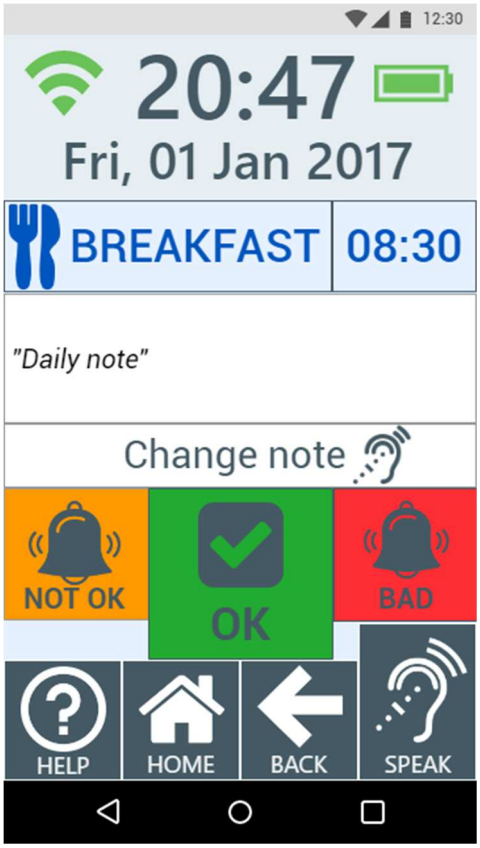


Iconen van activiteiten.

In het volgende tabel vindt u de verschillende icon en hun categorie.

	Normal visit icon
	Housekeeping icon
	External appointment icon
	Internal appointment icon
	Medical intake icon
	Medical visite icon
	Physical activity icon
	Mental activity icon
	Nutrition intake icon
	Default icon

Take a note



Nadat er op een activiteit word geklikt komen we op dit scherm terecht. We krijgen hier het activiteit te zien en alle andere knoppen dat in het mockup besproken werden.

Bijlage X: Device

Door: Koumeyl Belkhidar

Device

Voorwoord

Voor het functioneel analyse is het belangrijk om te zien welke apparaten bruikbaar zijn voor ons project. Aangezien ons project voor een specifiek publiek ontwikkeld wordt is het belangrijk om te weten welke device aan de eisen van het publiek kan voldoen.

Rasperbery PI

Een Raspberry Pi is een computer gebouwd op een enkele printplaat (singelboardcomputer). Het is dus heel compact en wordt verkocht aan een minimale prijs. Het laatste versie, de Raspberry PI 3 Model B, is een 1.2 GHz Quad-Core met 1GB RAM geheugen. Het heeft een netwerkaansluiting en ondersteund WIFI en bluetooth 4.1. Voor het opslaggeheugen beschikt het over een microSD-poort. Het apparaat beschikt ook over 4 USB-poorten en een HDMI-poort. Wat de OS betreft zijn er gecostomizeerde versies vans linux beschikbaar voor rasbery

Prijs van complete starter pack met screen **94,78 €.**

Flirc een universele infrared-reciever usb. Kan gebruikt worden om een afstandsbedieningen te koppelen aan een Rasberry Pi.

Prijs: **32€**

Android

NFC

NFC is een contactloze communicatiemethode. Het kan gebruikt worden voor authenticatie in ons project. Android apparaten kunnen met NFC werken. Een mogelijkheid zou zijn om door een NFC kaart een gebruiker te kunnen authentificeren.

Prijs: **8,70 €**

Qmote

Qmote is een bluetooth afstandsbediening voor smartphones en home automation. Het is met een app geleverd dat het mogelijkheid geeft om het afstandbediening te controleren en te configureren volgens de eisen van het gebruiker. Het levert een lijst van functionaliteiten. Het kan een app openen, volume verminderen, een locatie doorsturen enz....

Prijs: **32$**

Google home

Google Home is a voice-activated speaker powered by the Google Assistant. Ask it questions. Tell it to do things. It’s your own Google, always ready to help. Just start with, "Ok Google".

203,00 €

Andere voorbeelden Amazon Echo.

Bijlage VIII: Google Voice Action

Door: Koumeyl Belkhidar

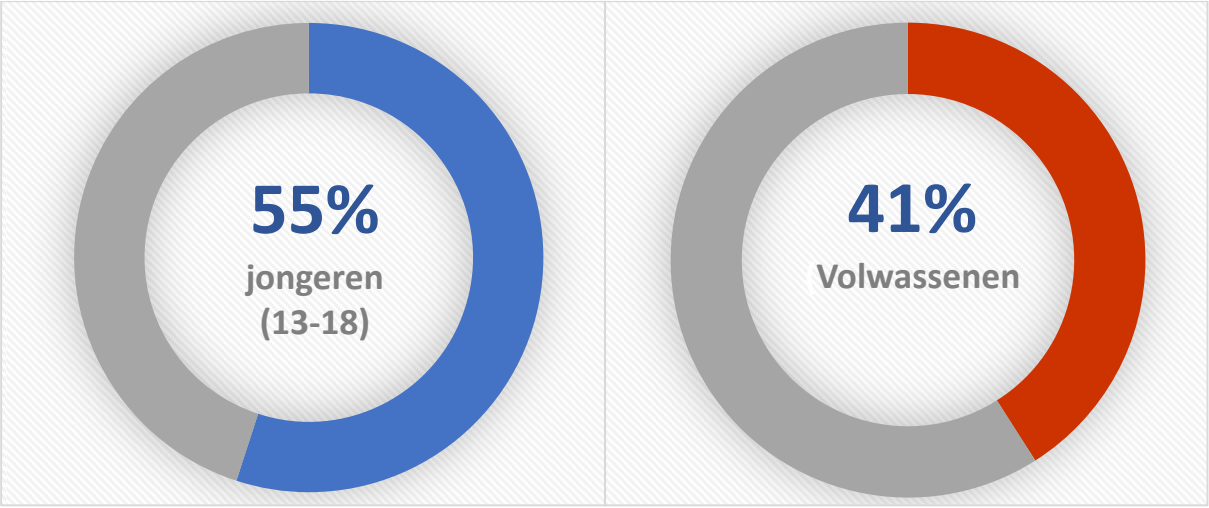
Google Voice Action

Voorwoord

Voor ons functioneel analyse hebben we beslist om ons te verdiepen op de verschillende Voice Technologieën. Google Voice Action is een bekende spraaktechnologie. In het volgende analyse gaan we de verschillende aspecten hiervan bekijken en zien of het bruikbaar is voor ons project.

Introductie

Google is een bedrijf dat mensen met informatie verbind. Wereldwijd wordt er per maand meer dan 100 miljard opzoekingen op google gedaan. Meer en meer van opzoekingen worden op andere device gedaan dan op computer. In 10 landen (onder andere Japan en de Verenigde Staten) worden de opzoekingen in meerderheid op een mobile device gemaakt. Jongeren gebruiken meer Voice Search (technologie om met de stem opzoekingen te maken) dan ouderen.



Voeren dagelijks meer dan 1 keer vocaal opzoekingen uit.

Waarom gebruiken mensen "Spraakgestuurd zoeken"?

Meestal is het gebruik hiervan gemeenschappelijk. Ouderen en jongeren gebruiken het voor situaties zoals iemand opbellen of routes opvragen. Maar het gebruik kan ook verschillen. Jongeren kunnen Voice search veel gebruiken voor hun huiswerk waar daarentegen zouden ouderen het meer gebruiken om sms-berichten te sturen.

Wanneer wordt Spraakgestuurd zoeken" gebruikt?

Voor jongeren als voor volwassenen wordt Voice search meestal gebruikt met vrienden of voor de televisie. Sommige gebruiken het tijdens het uitoefenen van sport of bij het koken.

Wat zouden mensen met "Spraakgestuurd zoeken" willen doen?

Volgens de statistieken van Google zouden mensen graag pizza's kunnen bestellen (45% jongeren en 36% volwassenen). Mensen zouden ook graag de afstandsbediening kunnen vinden met Spraaktechnologie (34% jongeren en 44% volwassenen). Ten laatste zouden mensen ook hun sleutels willen opzoeken met spraaktechnologie (34% jongeren en 44% volwassenen).

Google hoopt dat ze een tool zouden kunnen tonnen dat kan gebruikt worden bij het ontwikkelen van applicaties.

Gebruikte technologieën

We gaan nu de werking van Google Voice Action uitleggen en de technologieën die eraan gekoppeld zijn.

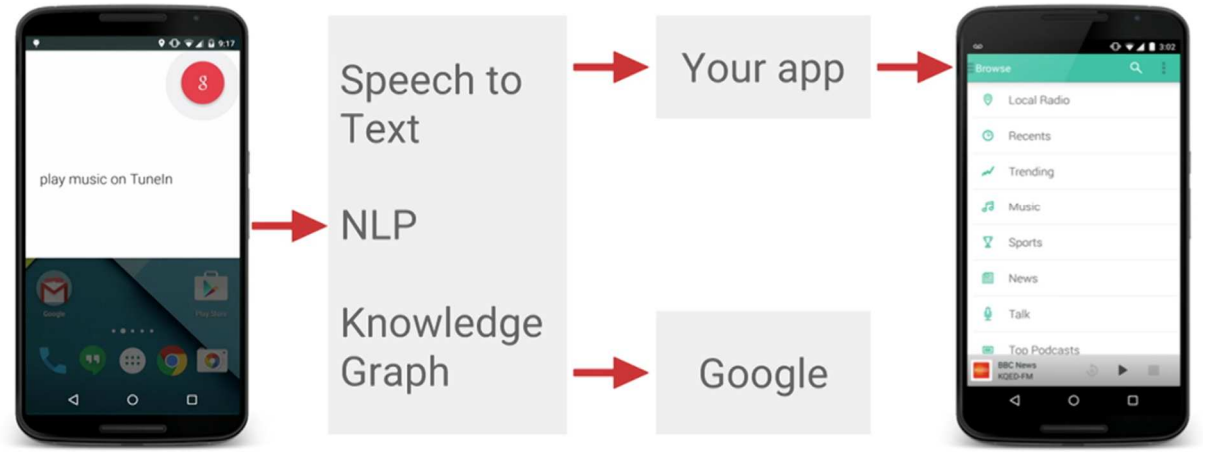
Knowledge Graph

Knowledge Graph is een technologie dat google doorheen de jaren heeft gebuild zodat het gebruikt kan worden bij het ontwikkelen van technologieën. Het wordt bij verschillende projecten bij google gebruikt en spraaktechnologie is een van de begunstigde hiervan. Knowledge Graph is een technologie dat het informatie over iets en de relaties die eraan gekoppeld zijn bijhoud. Dankzij Knowledge Graph is het kans dat een opzoeking niet slaagt van 20 % naar 8% verminderd in de laatste jaren. Spraaktechnologie bij google ondersteund 20 talen en het is ook meer toegankelijk geworden. Geen training meer nodig zoals het vroeger het geval was. Google vindt nu dat hun technologie goed werkt en dat er dus bruikbare apps mee gemaakt kunnen worden.

Google Voice Action

Google Voice Action ook wel Google Voice search genoemd is een technologie dat beschikbaar is op Android. Het is volledig gratis ook bij development. Het werd sind's het Android versie 4.1+ (Jelly Bean) geïntegreerd in het Google Now applicatie.

Werking Spraakgestuurd zoeken.



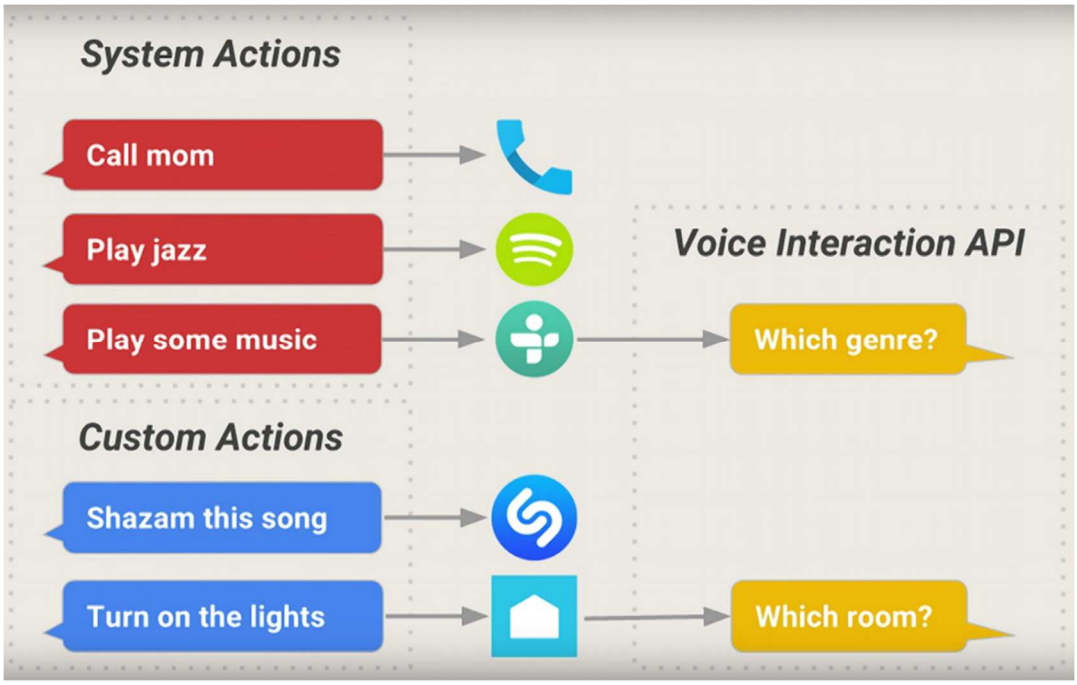
Eerst spreekt de gebruiker door het google-microfoon. Daarna wordt wat het omgevormd naar tekst. Vervolgens wordt wat de user heeft gevraagd geïnterpreteerd. Daarna wordt er via de Knowledge Graph een oplossing gevonden en word het resultaat op google getoond.

We gaan vervolgens bekijken hoe u google zou informeren over de mogelijkheden dat uw app geeft zodat gebruikers uw app kunnen gebruiken met het spraaktechnologie van google.

Android Speech Recognition vs. interpretation of speech.

Android had al een API dat het herkennen van stem implementeerde. Dit API gaf terug wat er gezegd werd in het Android apparaat. Het werkt als een transcriptie. Het gaf gewoon de woorden dat er gezegd werden. Dit was het eerste API maar doorheen de tijd introduceerde Google andere API's.

Voice actions API's



System Voice Actions

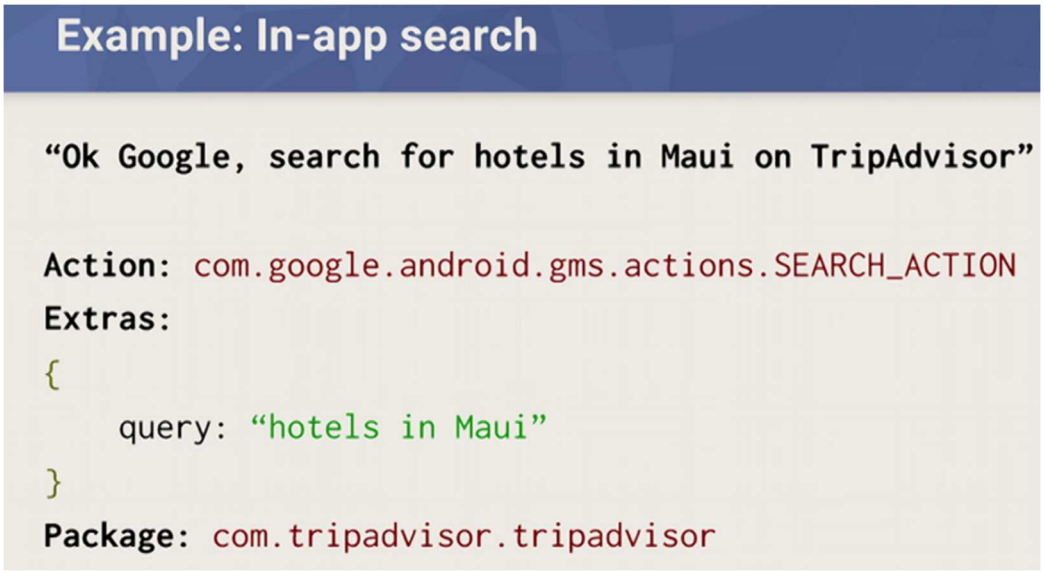
System Voice Actions zijn standaard acties dat mensen met hun Android apparaat willen doen. Google identificeert de acties en gaat dan definiëren welke app dit actie gaat kunnen uitvoeren.

Implementatie voorbeeld "In-App search":

Example: In-app search

```
<activity android:name=".SearchableActivity">
    <intent-filter>
        <activity android:name="com.google.android.gms.actions.SEARCH_ACTION"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

Dit is het gemakkelijkste te implementeren in een Android app. Die lijnen moeten bijgevoegd worden in het activatie intern filter van het app. Het gebruikt het Android search widget. Dankzij die enkele lijnen zegt uw applicatie aan google dat het opzoekingen kan uitvoeren



Dit is een voorbeeld van hoe Google Voice Actions het gaat interpreteren. Nadat Voice Action het weet dat het gebruiker iets gevraagd heeft op "Tripadvisor". Word er een query doorgestuurd aan het app en daar word dus de opzoeking uitgevoerd. Google Voice Actions ondersteund verschillende basic system acties.

Hier is een lijst van verschillende acties dat mogelijk zijn.

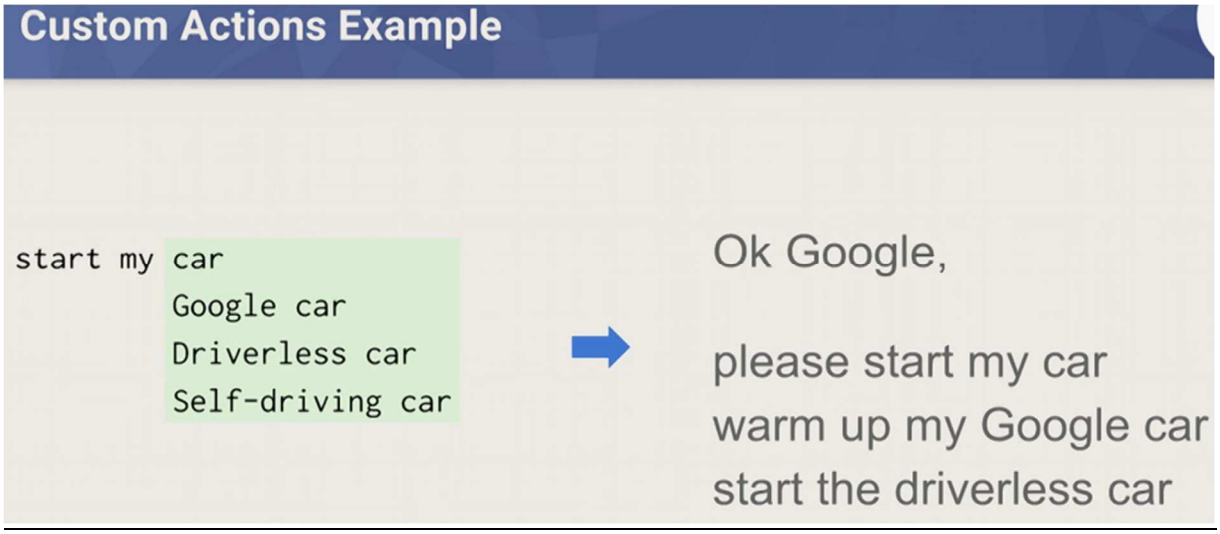
- Alarm actions
- Communication actions
- Fintess Actions
- Local actions
- Open actions
- Productivity actions
- Search actions

voor meer informatie en om een gedetailleerde weergave te krijgen van al die acties zie het Google developers website (https://developers.google.com/voice-actions/system/).

Custom Voice Actions

Soms kan het zijn dat er een specifiek terminologie in een Android app bestaat dat Google niet kent. Het is mogelijk om met Google dan te gaan definiëren wat die terminologieën zijn. In feite Google Voice Actions aanpassen volgens het gebruik dat uw app aanbiedt.

Voorbeeld:



Hier zien we een concreet voorbeeld in een app kan heet woord "car" op verschillende manieren gezegd worden. Die verschillende Actions dat uw app kan doen kunt uw gewoon gaan definiëren in Google Voice Actions.

Meer dan een 1000 apps hebben die soorten acties geïmplementeerd. Veel applicaties met messaging systemen hebben het geïmplanteerd (Zoals: WatsApp, Viber enz....).

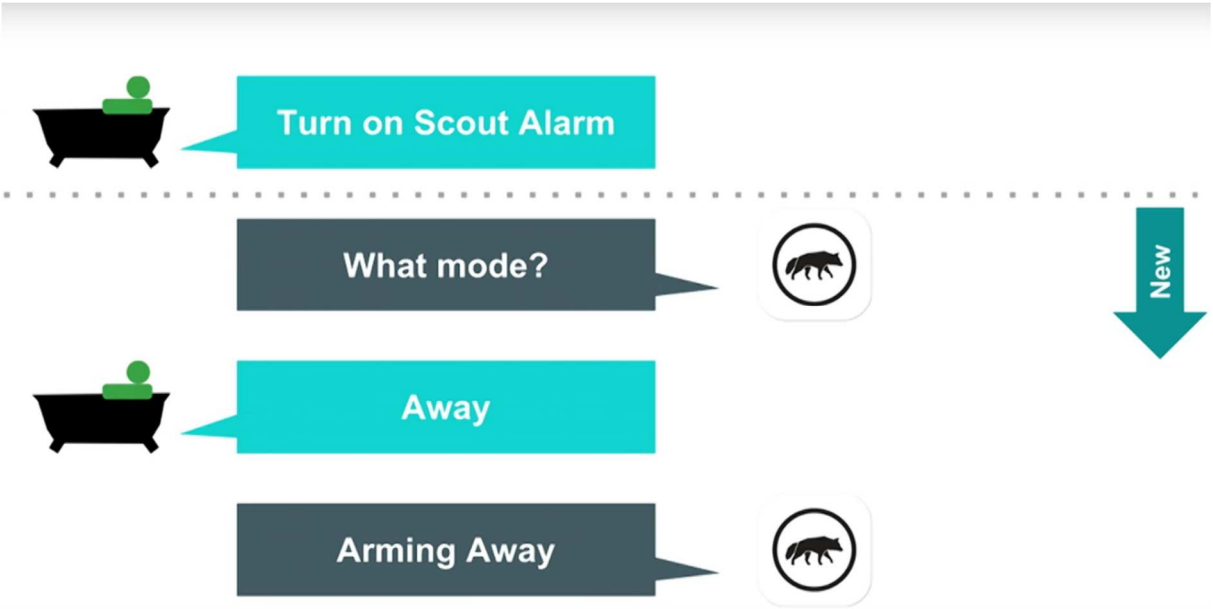
Edit: Volgens de documentatie van Google worden de aanvragen voor Custom Voice actions niet meer geaccepteerd.

Bron: https://developers.google.com/voice-actions/custom-actions

Voice Interaction API

In Android M had google een API dat Voice Iteraction API als naam had. Alles wat er nu besproken werd gaat u gewoon brengen naar uw applicatie. Vanaf dit punt stopt het spraaktechnologie belevenis vanuit uw perspectief. De user spreekt een actie dat uw app moet uitvoeren en het app voert het uit. Als het actie nog andere stappen nodig heeft gaat u gewoon de touchscreen weergeven. Maar met Voice Interaction API gaat het app vragen kunnen stellen aan de gebruiker over wat het app verder zou moeten doen.

Voorbeeld:



Gebruiker wilt Alarm aanzetten het alarm applicatie kan dan verder vragen stellen. Zoals welke modus wilt u? Het app gaat dan ook een confirmatie geven van de modus.

Implementatie voorbeeld:

Handling a Voice Interaction

```
<activity android:name=".MyVoiceActivity">
    <intent-filter>
        <action android:name="org.example.MY_ACTION"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.VOICE"/>
    </intent-filter>
</activity>
```

Aan de hand van dit code weet uw applicatie dat het gaat Voice Interaction implemeteren.

Handling a Voice Interaction

```
class MyVoiceActivity extends Activity {
  @Override
  public void onResume() {
    if (isVoiceInteractionRoot()) {
      startMyVoiceInteraction();
    }
    startMyTouchInteraction();
  }
}
```

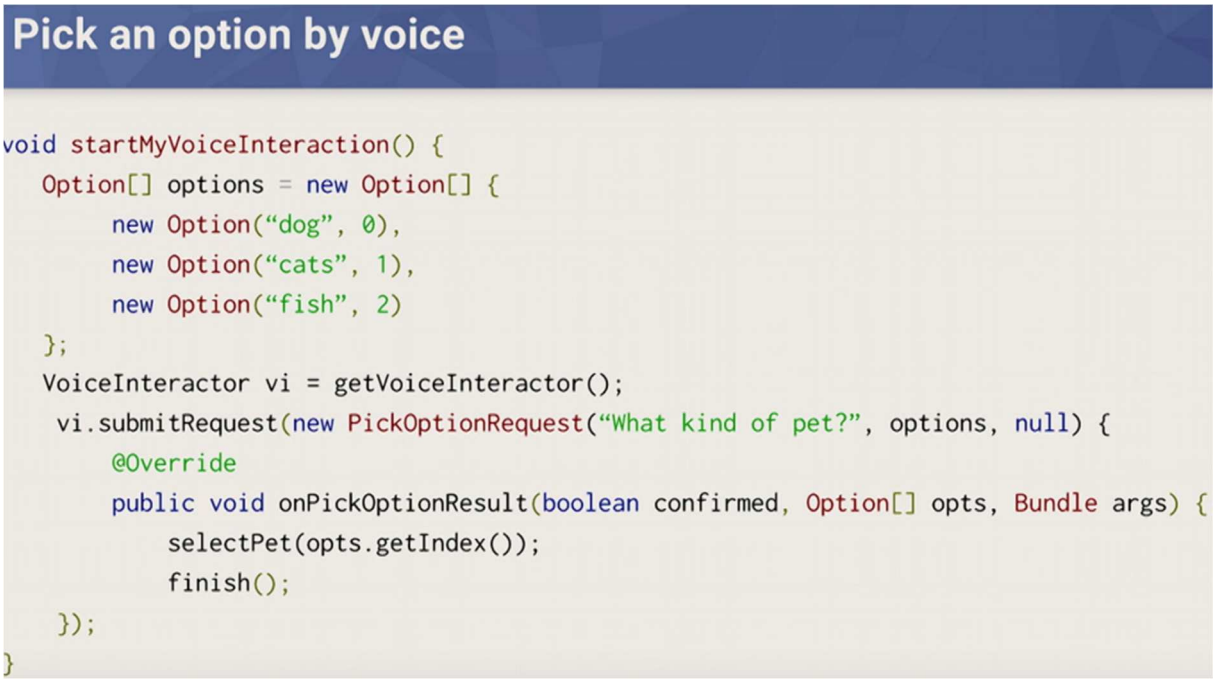
Daarna word er in een standaard Android activity gedefinieerd waar we gewoon gaan vragen of het een spraak interactie is of niet. Als het niet het geval is dan gewoon het touchscreen gebruiken.

Confirmation

```
void startMyVoiceInteraction() {
    getVoiceInteractor().submitRequest(new ConfirmationRequest("Are you sure?", null) {
        @Override
        public void onConfirmationResult(boolean confirmed, Bundle result) {
            if (confirmed) {
                doAction();
            }
            finish();
        }
    });
}
```

Hier wordt er een standaard confirmatie gedefinieerd bent u zeker om een bepaalde actie de voeren? ja of nee. Dit zou gebruikt kunnen worden zodat het applicatie nog kan vragen dat ze een bepaalde actie kan uitvoeren. In geval dat het applicatie het misschien niet goed begrepen heeft.

Andere voorbeeld dat een app zou kunnen vragen is "kies een optie".



Wat zou u willen kiezen. Dan gaan we een optie definiëren. Synoniemen zouden ook gedefinieerd kunnen worden. Maar in geval het spraaktechnologie het niet goed verstaat gaat het technologie vragen om het terug te zeggen. Het is ook daarna belangrijk om de gebruiker te informeren over wat er uitgevoerd werd. Nog belangrijker als het applicatie het action niet uitgevoerd heeft moet het ook uiteraard gezegd worden aan de gebruiker en misschien ook waarom. Dit allemaal kan bijgevoegd worden. Voor meer informatie over het werking en implementatie van het technologie in Android zie Google devlopper website (https://developers.google.com/voice-actions/).

Conclusie en advies voor het project.

Pro's	Contra's
+ Gratis	**- Niet mutliplatform**
+ Goed gedocumenteerd	**- Geen custom acties mogelijk**
+ 20 Talen ondersteund	
+ Spraaktechnologie is stabiel	
+ Weinig error's	

Google Voice actions is een stabiel technologie. Het wordt gebruikt door veel bekende app's (TripAdvisor, Shazam, Spotify). Het ondersteund verschillende talen en is volledig gratis. Daarentegen is Google Voice Action toch gelimiteerd. Het werkt alleen met Android en laat ons dus niet de kans om met verschillende programmeertalen of besturingssystemen te werken. Het geeft ook niet de mogelijkheid om custom acties te schrijven wat in ons project uiteraard een probleem is. Het project heeft specifieke eisen dat de System actions niet kunnen ondersteunen.

Bijlage XI: Implement

Door: Koumeyl Belkhidar

Implement

Introductie

Na het design fase komt het implement fase. Dit fase bevat het ontwikkelen en verwerken van een werkend prototype. De verschillende technologieën dat gebruikt werden om het prototype functioneel te maken gaat hier ook besproken worden.

Gebruikte technologieën

Android

Er werd beslist na analyse dat er met Android gewerkt ging worden. Dit omdat het technologie opensource was en goed omging met andere technologieën dat gebruikt werden bij het ontwikkelen van het project. Het initieel app was ook een smartphone applicatie het was dus ook een reden waarom we met Android gingen werken. Api.ai is ook compatibel met Android. Een android device heeft een camera en micro waarmee we videocalls kunnen maken. Dit was ook een vereiste van het bedrijf.

Sinch

Sinch[6] is een oplossing waarmee we videocalls via android kunnen maken. Het SDK (Sofware devlopment kit) biedt het mogelijkheid om app tot app telefoongespreken te maken. Het biedt echter ook app tot telefoon services en om een dubbelauthenticatiesysteem via sms te implementeren.

Twilio

Twilio[7] is een bedrijf dat zoals Sinch een Voice & video solution bieden. Waar Twilio een meerwaarde heeft is dat ze ook de mogelijkheid geven om in een applicatie een sms-systeem te implementeer. Dit gaan we nodig hebben om het emergency systeem te ontwikkelen (zie emergency screen).

Er werd ook gebruikt gemaakt van API.AI dat een gedetailleerde analyse heeft gekregen (zie bijlage: technische analyse.)

NFC

NFC is een contactloze communicatiemethode dat een extensie is van radiofrequecy. Veel smartphones hebben het basis geïntegreerd. In het prototype wordt het gebruikt bij het authenticatie (zie loginscherm).

---

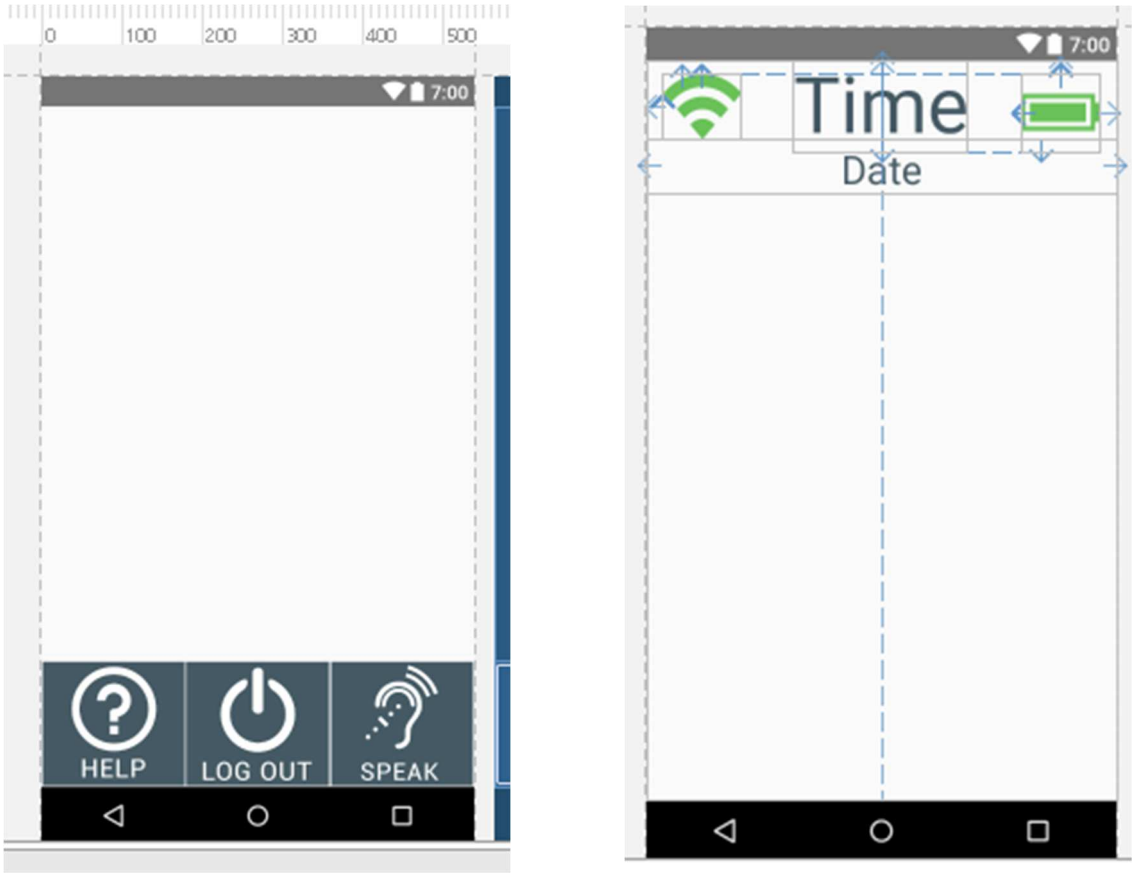
[6] https://www.sinch.com/
[7] https://www.twilio.com/

Implementatie werkend prototype

Hier wordt er stap per stap uitgelegd hoe het prototype ontwikkeld werd. Het werd als volgt aangepakt de screens werden 1 per 1 geïntegreerd in het programma. De knoppen en layout werden toegevoegd aan de drawable's van het project. Er werden ook verschillende designs aangemaakt tijdens de ontwikkeling volgens de eisen van het development.

1. MainActivity

Het eerste Activity dat gemaakt werd is het MainActivity. Het MainActivitie bestaat uit een hoofding en een footer. Dit komt verder in het applicatie op bijna elke screen aan bod dus was het belangrijk om het goed te implementeren. Er werd dus rap beslist om het layout van de header en de footer apart aan te maken en daarna te includen in de Activities waar ze aan bod komen. Header wordt op een assychroon manier elk second geupdate. Zodat we de status van de batterij en wifi, de tijd en de datum juist hebben.



Afbeelding 1.1: Footer en header design in android.

```
        <Button
            android:id="@+id/buttonEmergency"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/buttonSettings"
            android:layout_alignBottom="@+id/buttonSettings"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:background="@drawable/emergency" />
    </RelativeLayout>

    <include layout="@layout/footer" />
</RelativeLayout>
```

Afbeelding 1.2: Include van header in het mainscreen.

Afbeelding 1.3: MainActivity

Op het scherm hiernaast krijgen zien we de MainActivity waar we 4 grote knoppen te zien krijgen. De reden waarom het applicatie zo werd ontworpen word besproken in het design bijlage. Het volgende onderdeel dat geimplementeerd werd in het applicatie is het login systeem. We kunen op het LoginActivity terecht komen als we op log-out button drukken.

2.LoginActivity

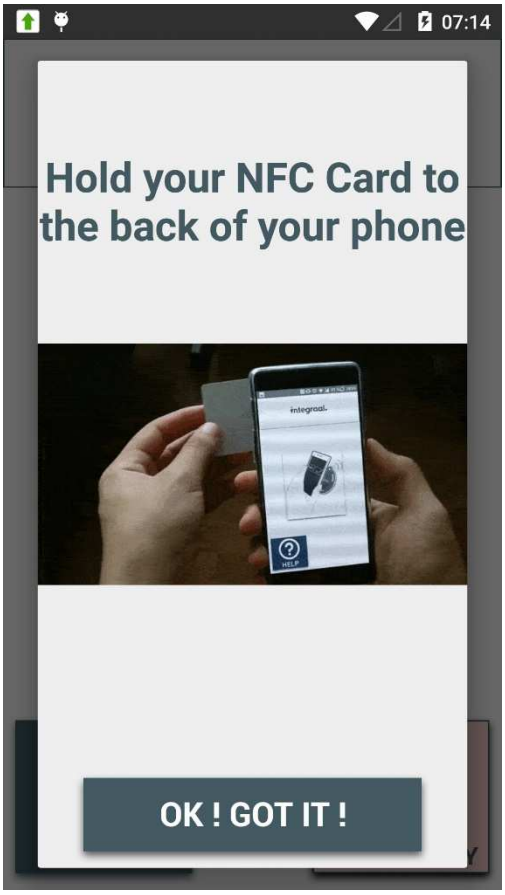
Dit is het LoginActivity dat bestaat uit twee buttons. Op dit scherm kan het gebruiker zich authenticeren via een NFC-kaart dat een unieke code bevat. Het NFC-kaart is niet genoeg want het systeem gaat ook nakijken of het Mac-Adress van het toestel in het systeem geregistreerd is. Uiteindelijk wordt er een laatste check aangemaakt om te zien of het mac-adress en het NFC-kaart op het tag wel juist zijn. Een kaart behoort alleen maar aan één toestel en als het user het kaart verliest moet hij het systeem administrator contacteren.

De user beschikt hier ook over een Helpknop. Dit knopt opent een scherm waarop er via video getoond word hoen een gebruiker moet inloggen (zie afbeelding 2.2).

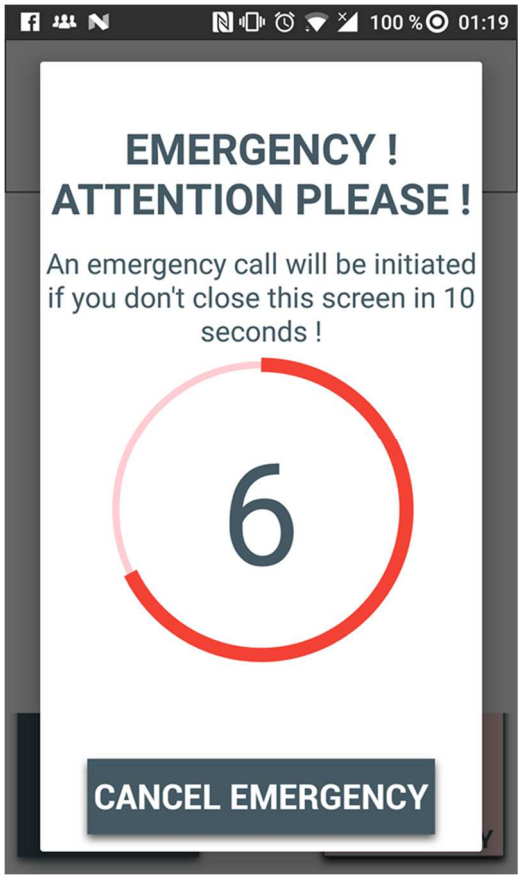
Naast het helpbutton zien we ook het emergency button. Wanneer er op het button gedrukt word. Krijgt het gebruiker een AlertDialog te zien waarop gewaarschuwd word met een spinner dat een alarmbericht doorgestuurd zal worden na 10 seconden.

Afbeelding 2.1: LoginActivity

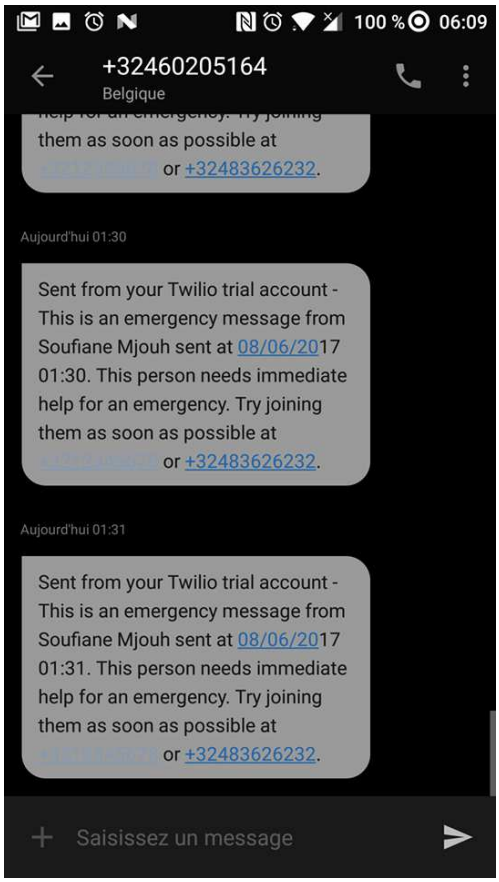
Het sms wordt via het Twilio systeem doorgestuurd dankzij hun sdk dat in het project geimplementeerd werd. Het sms bericht wordt aan alle personnen dat ingesteld werden om het bericht te krijgen doorgestuurd. Het kunnen mantelzorgers zijn of verpleegsters bijvoorbeeld.



Afbeelding 2.2: Hulpscreen met video van alertdialog hoe een gebruiker moet inloggen.



Afbeelding 2.3: Emergency met het countdown spinner.



Afbeelding 2.4: sms-berichten dat aankomen op het gsm van het mantelzorger.

3. Planning

Als in de MainActivity op planning drukken komen we op het planning-menu terecht.

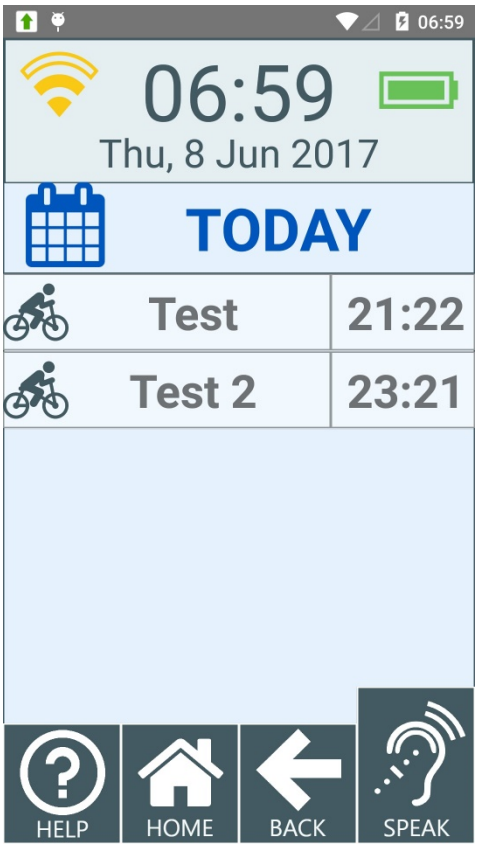
Afbeeldingen3.1: PlanningActivity (links) en ShowPlanningActivity (Rechts)

Op het PlanningActivity kan de gebruiker kiezen om een planning te raadplegen of om een activiteit toe te voegen aan zijn planning.

Als het gebruiker op het "Show Planning"-knop drukt krijgt hij de ShowPlanningActivity te zien waar hij kan kan kiezen op welke dag hij zijn planning wilt raadplegen. Hier geven we de parameters door aan het volgende activiteit. En aan de hand van een if structuur krijgen we de juiste datum tevoorschijn.

Als het gebruiker op het choosedate knop drukt krijgt hij het volgende scherm te zien. Om dit scherm kan hij het dag, het maand en het jaar kiezen door op het plus of min knop te drukken. Hij kan ook het datum mondeling uitspreken door op het oortje te drukken. Nadat het datum uitgesproken wordt krijgt de gebruiker het te zien in het textvieuw na het knop. Als het datum correct is kan hij het datum doorsturen door op select date te drukken. Het cancelknop dient om alles te annuleren.

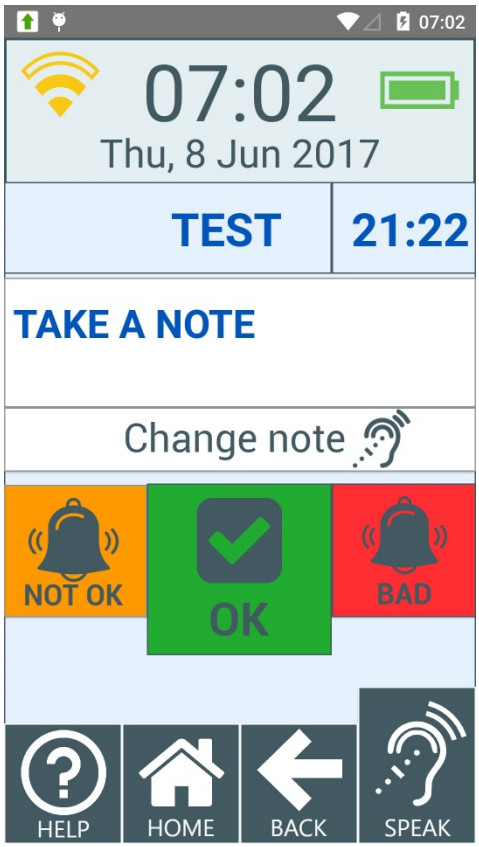
Afbeelding 3.2 choose date



Afbeelding 3.3 PlanningShowDay

Als een gebruiker het datum kiest komt hij op het PlanningShowDay terecht. De titel van het activity veranderd in kwestie van de gevraagde datum. Het kan Today, Tomorrow, Yesterday, of een datum bevatten.

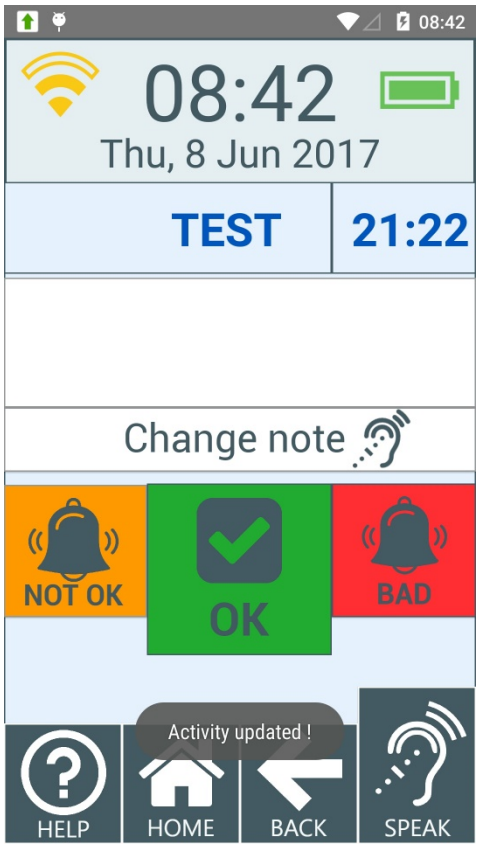
De gebruiker krijgt op dit activity een lijst van alle activiteiten dat op die dag gepland zijn. Naast het activiteit krijgen we ook het uur te zien waarop het gepland is en een icoon. Het icoon geeft aan welk soort activiteit het is. Er zijn negen soorten activiteiten in het systeem (meer detail in het design bijlage). Als we op een activteit klikken kommen we op het volgende Activity terecht.



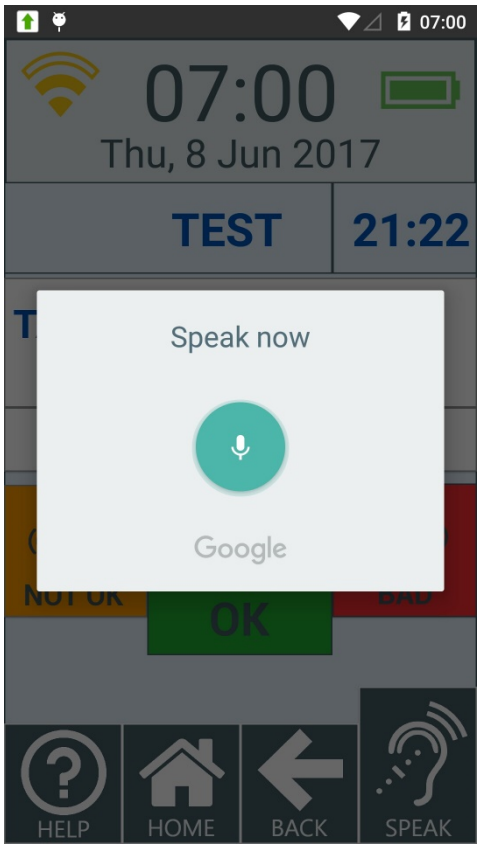
Afbeelding 3.4 PlanningDetailActivity

Op het PlanningDetailActivity krijgen we meer detail over het geselecteerde activiteit. We kunnen hier een notitie nemen over het activiteit het kan vocaal of schrijvend gebeuren. Een gebruiker kan ook zijn mening uiten via de 2 alarm knoppen en het ok knop.

Oranje wil zeggen dat het slecht verlopen is. Rood dat er een zwaar probleem is gebeurd. Groen dat het goed verlopen werd. Als men op een knop drukken krijgen we een toast te zien dat ons zegt dat het activiteit geupdate werd. Het toost word ook vocaal uitsproken zoals alle toast's van het programma



Afbeelding 3.5: Activity met toast



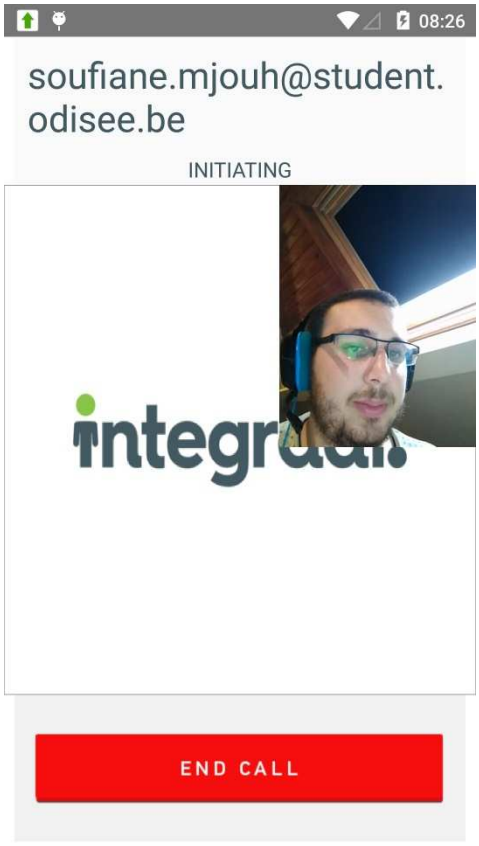
Afbeelding 3.6: Note word ingesproken

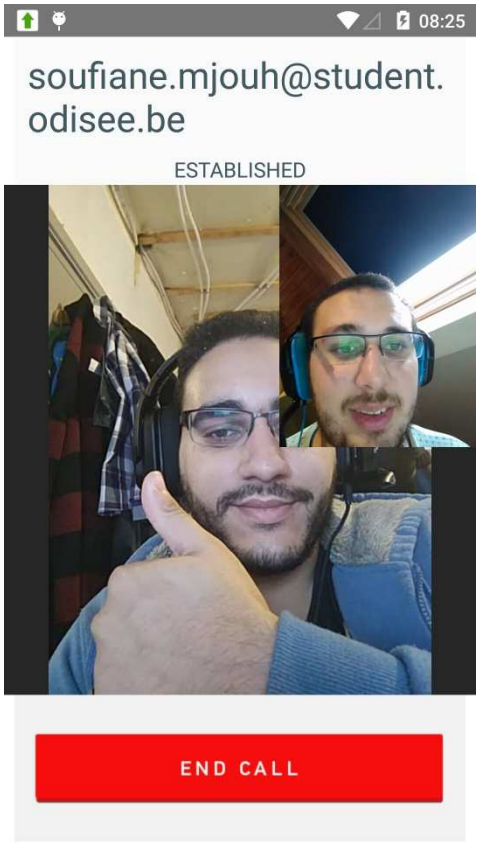
CallSomoeneActivity

Afbeelding 4.1: CallSomoeneActivity

Afbeelding 4.1: CallSomoeneActivity met offline persoon.

Als men op het "MainActivity" op de "Callsomeone" knop drukken komen we terecht op het het CallSomoeneActivity. Op dit scherm krijgen we een lijst van personen te zien. Die personen zijn de contacten aan wie we kunnen bellen en dat op het systeem geregistreerd zijn. Als een persoon offline staat wordt het knop vergrijst. Als het gebruiker erop drukt krijgt hij een toast te zien dat hem zegt dat het gebruiker momenteel offline is. Het toast wordt ook vocaal gezegd door de applicatie. Als het gebruiker online staat kan men een videocall met het gebruiker starten.

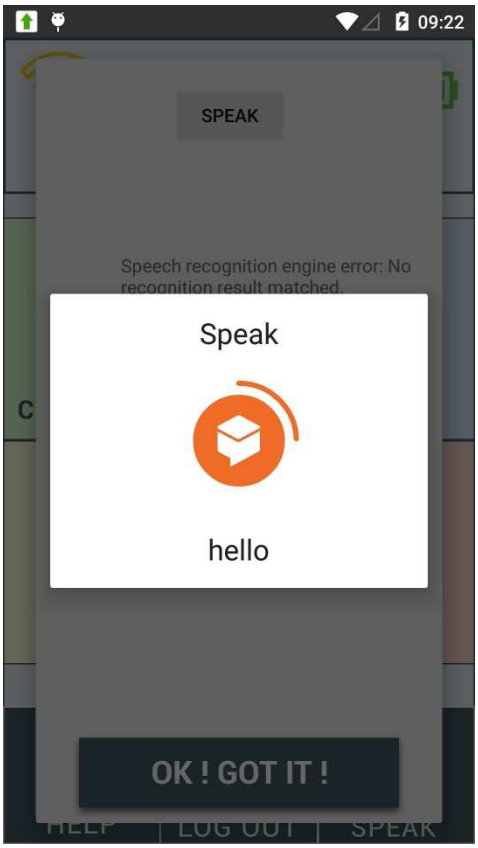




Afbeelding 4.2 video call wordt gestart *Afbeelding 4.3 gebruiker antwoord op de call*

Het call systeem werkt met het sinch sdk. Het sdk is goed gedocumenteerd en biedt veel mogelijkheden.

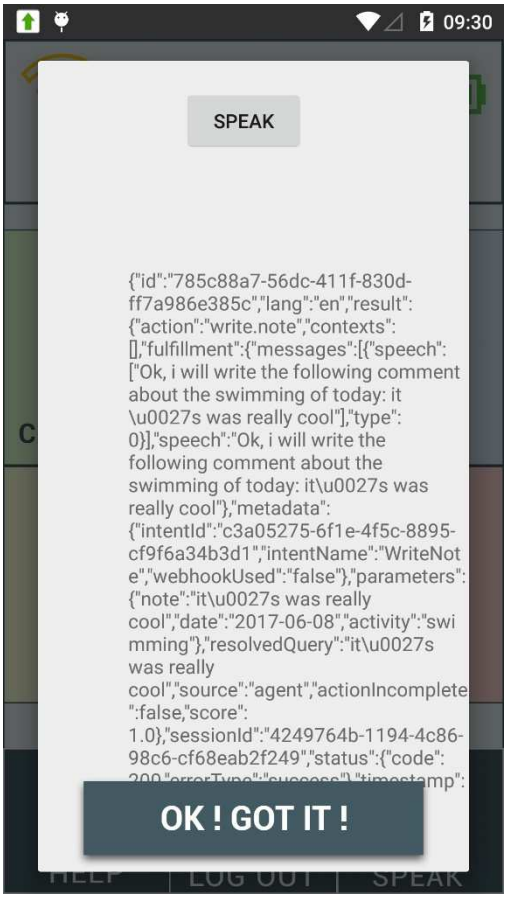
Speakbutton



Op bijna elk scherm van het applicatie krijgen we een speakbutton te zien. Het speakbutton geeft de mogelijkheid om een notitie vocaal op te kunnen slaan door een conversatie met een ai-agent te doen. Als men vraagt om een notitie te nemen voort het ai-agent een conversatie met de gebruiker uit waar ze vraagt wat ze nodig heeft om de taak uit te kunnen voeren. Na het informatie krijgen we een json file te zien waar we al het informatie dat we nodig hebben krijgen om het in de back-end te stockere.

Afbeelding: SpeakButton

Conversatie met api.ai-agent

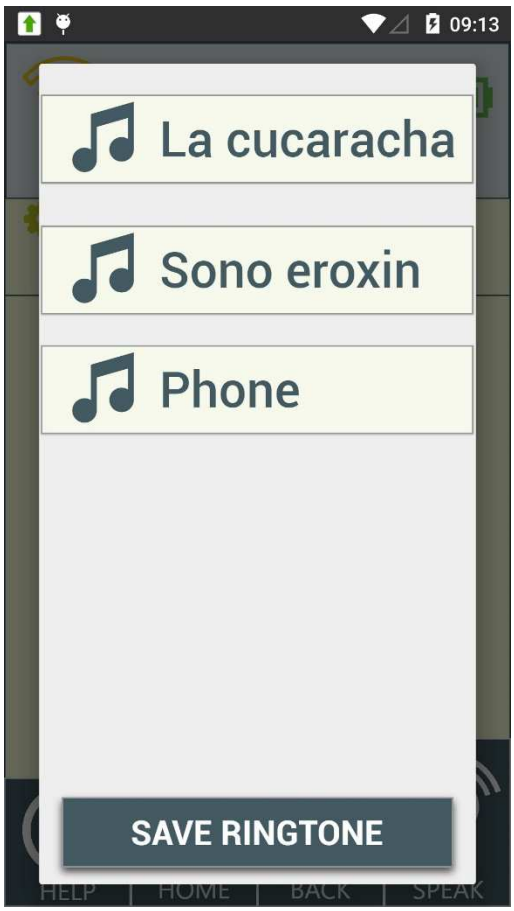


Op het afbeelding heernaast kan er een antwoord geraadpleegd worden. We krijgen hier alle gegevens in verband met het informatie het actie dat uitgevoerd word (namenlijk write.note) en de verschillende parameters. We krijgen hier de datum, het activiteit en het note dat we daarna in ons database kunnen opslaan.

Settings

Afbeelding 5.1: SettingsActivity

Het laatste onderdeel van het applicatie is settings. Bij settings kunnen we ons ringtone veranderen. Andere features kunnen in het definitief app bijkomen zoals he veranderen van zijn profiel-foto enz. Als men op het change Ringtone button drukken krijgen we een alertdialog te zien dat ons de mogelijkheid geeft om van ringtone te veranderen.



Conclusie

De verschillende features dat er in de functionele analyse beschreven werden en de directieve van het bedrijf werden zo goed mogelijk gerespecteerd. We hebben aan de eisen van het bedrijf kunnen antwoorden met het dit prototype en kunnen aantonen dat een app maken voor oudere mensen mogelijk is. Het prototype heeft alle gevraagde features en zelfs features dat initieel niet gevraagd werden zoals het emergency systeem.