



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Valid Ahmadi

TIETOKANNAN LUKEMINEN VIRTUAALIAVUSTAJALLA

Tekniikka
2019

TIIVISTELMÄ

Tekijä	Valid Ahmadi
Opinnäytetyön nimi	Tietokannan lukeminen virtuaaliavustajalla
Vuosi	2019
Kieli	suomi
Sivumäärä	52
Ohjaaja	Timo Kankaanpää

Tekoälystä ja virtuaaliavustajista ollaan puhuttu historian aikana enemmän kuin monesta muusta asiasta, ja nykyään virtuaaliavustajat ovat tulleet jäädäkseen ja tulee lisääntymään ajan mukaan.

Virtuaaliavustajat ovat vielä kehitysvaiheessa, jonka takia tässä opinnäytetyössä alussa perehdytään puheentunnistusmenetelmien historiaan, vertaillaan eri virtuaaliavustajia, kuinka muuttaneet ihmisten elämää ja virtuaaliavustajien tulevaisuuden mahdollisuuksia. Työssä tarkoituksena on osoittaa ja näyttää miten voi ohjelmoida virtuaaliavustajaa, kärkeä virtuaaliavustajaa hakemaan tietoja tietokannasta ja lukea ne käyttäjälle ääneen.

Työssä on otettu huomioon se, että lukijalla ei ole aikaisempaa kokemusta ja paljon tietoa virtuaaliavustajista näin ollen on selitetty askeleittain, miten työ on toteutettu. Työ onnistui hyvin ja tavoitteisiin päästiin.

ABSTRACT

Author	Valid Ahmadi
Title	Dataabse browser using a virtual assistant
Year	2019
Language	Finnish
Pages	52
Name of Supervisor	Timo Kankaanpää

It has been spoken about Artificial intelligence and virtual assistants during history more than many other things, and nowadays virtual assistants have come to stay and will increase over the time.

Virtual assistants are still in the development phase, which is why this thesis begins with a familiarization with the history of speech recognition methods, compares different virtual assistants, how they changed people's lives and the future of virtual assistants, the purpose of the work is to show how to program a virtual assistant, instruct the virtual assistant to retrieve information from the database and read it to the user aloud.

In this work I have taken into account the fact that the reader has no previous experience and much information about the virtual assistants therefore explained step by step, the work was successful and the goals were reached.

ALKUSANAT

Haluan kiittää yliopettaja Timo Kankaanpäättä, hän on tukenut minua aina kun olen tukea tarvinnut. Lisäksi haluan kiittää suuresti professori Petri Heloa mielenkiintoisesta projektista minkä sain häneltä sekä tuesta. Ilman häntä tätä projektia ei olisi.

Haluan vielä kiittää Vaasan ammattikorkeakoulun opettajia kuluneista kouluvuosista sekä suuret kiitokset mahtaville kavereille.

Lopuksi suuret kiitokset isälleni ja äidilleni, ilman heitä en olisi tässä.

Vaasa 19.12.2019

Valid Ahmadi

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVALUETTELO

TAULUKKOLUETTELO

1 JOHDANTO	10
2 PUHEENTUNNISTUS	11
2.1 Historia	11
2.2 Virtuaali/Ääniavustajat.....	13
2.3 Smart speakers.....	21
2.4 Ääniavustajien hyödyt	22
2.5 Ongelmat	23
2.6 Tulevaisuus.....	23
3 PÄÄTTÖTYÖSSÄ KÄYTETYT TEKNIIKAT	24
3.1 HTML5.....	24
3.2 Javascript.....	25
3.3 CSS.....	26
3.4 NodeJS	27
3.4.1 NodeJS Moduulit	28
3.5 Fulfillment	29
3.6 Webhook	29
3.7 Firebase Realtime Database	29
3.8 Dialogflow.....	29
3.8.1 Dialogflow agentin luonti	30
3.8.2 Intent luonti.....	34
3.8.3 Entity luonti	37
3.8.4 Fulfillment	38
3.9 Realtime Database luonti	41
3.9.1 Mahdollisia ongelmia	44
4 SOVELLUKSEN TOTEUTTAMINEN WEB-SIVUNA	44

5 OHJELMAN TESTAUS	49
6. YHTEENVETO.....	51
LÄHTEET	52

KUVALUETTELO

Kuva 1. Siri-käyttöliittymä.....	15
Kuva 2. Cortana-käyttöliittymä.....	16
Kuva 3. Cortana Halo pelisarjassa.....	17
Kuva 4. Alexa-käyttöliittymä.....	18
Kuva 5. Google Assistant-käyttöliittymä.....	19
Kuva 6. Bixby-käyttöliittymä.....	20
Kuva 7. Amazon Echo.....	21
Kuva 8. Google Home.....	21
Kuva 9. Agentin luonti Dialogflow:lla.....	22
Kuva 10. Valmiit Intentit.....	23
Kuva 11. Valmiita harjoitusfraaseja.....	24
Kuva 12. Google assistantin valinta.....	25
Kuva 13. Intentin luonti.....	26
Kuva 14. Projektin kaikki intentit.....	26
Kuva 15. Parametrit	27
Kuva 16. Factory intentin harjoitusfraasit.....	28
Kuva 17. Webhookin päälle laittaminen.....	29
Kuva 18. FactoryId entityn viittausarvojen ja synonyymien laitto.....	29
Kuva 19. FactoryContent entityn viittausarvojen ja synonyymien laitto.....	30
Kuva 20. Inline editorin päälle laittaminen.....	30
Kuva 21. Firebase SDK käyttöönotto.....	31
Kuva 22. Welcome ja fallback funktiot.....	32

Kuva 23. FactoryInside funktio.....	32
Kuva 24. Arvon tulostus	32
Kuva 25. Funktion suoritus.....	33
Kuva 26. Firebase Developer osiot	34
Kuva 27. Tietokannan turvallisuussäännöt	34
Kuva 28. Tietokanta (Database)	35
Kuva 29. Tietokannan turvallisuussäännön editointi	36
Kuva 30. Firebase Logs	36
Kuva 31. Factory 1 käyttöliittymä	37
Kuva 32. Html lähdekoodi head element	37
Kuva 33. Css Tooltip.....	38
Kuva 34. Css tausta ja mittarit	38
Kuva 35. Css landscape.....	39
Kuva 36. OEE mittari ja sen selitys	39
Kuva 37. Scriptviittaukset	39
Kuva 38. Javascript fetch	40
Kuva 39. Javascript OEE.....	40
Kuva 40. Google Assistant linkki testaukseen	41
Kuva 41. Google Assistant Simulator	41
Kuva 42. Invocation osio.	42

TAULUKKOLUETTELO

Taulukko 1. Älyavustajat.....	40
--------------------------------------	----

1 JOHDANTO

Monet lapset ovat varmasti haaveilleet tai nähneet elokuvissa esimerkiksi leijuvaan päähän, joka vastaa jokaiseen kysymykseen, nauraa, tekee asioita komentojen mukaan tai ovat halunneet tekoälyllä toimivan ystävän kuten elokuvassa A.I. vuodelta 2001.

Olemme aina halunneet voida kommunikoida laitteiden kanssa, tai ainakin saada laite vastaamaan meidän kysymyksiin äänen avulla.

Ei ole enää tieteisfiktiota tai fantasiaa, jotta voisi puhua virtuaaliavustajan kanssa, se mikä näimme elokuvissa, on tosi asiaa ja monilla tärkeimmillä yrityksillä on oma virtuaaliavustajansa.

Vuonna 2011 tapahtui jotain mullistavaa nimittäin Apple julkisti ensimmäisen puhetta tajuavan virtuaaliavustajan iphonelle. Sirin suosion myötä kilpailijat julkaisivat omat versiot virtuaaliavustajasta.

Äänentunnistus kehittyy koko ajan samalla kun sitä mahdollistava teknologia, komennot lisääntyvät. Tämän opinnäytetyön lukijalla ei ole luultavasti paljon tietoa virtuaaliavustajista, alussa perehdytään virtuaaliavustajan menneisyyteen, tekoälyyn, kuinka muuttanut ihmisten elämää. Lisäksi vertaillaan nykyteknologiayritysten eri virtuaaliavustajia, älykaiuttimia sekä pohditaan virtuaaliavustajan tulevaisuutta ja näytetään askeleittain kuinka saada virtuaaliavustaja hakemaan tietoja tietokannasta ja lukea sitä ääneen loppukäyttäjälle. Virtuaaliavustajana tässä työssä käytetään Google Assistantia

2 PUHEENTUNNISTUS

Puheentunnistus on menetelmä, jolla tietokone pystyy tunnistamaan ihmisen puheääntä. Sen avulla ihmisen ei tarvitse käyttää käsiään pelkästään vaan ääntään erilaisten laitteiden ohjaukseen, mikä helpottanee monien ihmisten elämää, kuten vammaisia, joilta puuttuu tietty vartalon osa joko sodan tai syntymän takia. Menetelmä on helppokäyttöinen sekä lisää tuottavuutta./1/

Puheentunnistuksella on myös kompastuskiviä, jotka hidastavat sen käyttöönottoa, esimerkiksi jos laitteisto ei täytä tiettyjä vaatimuksia, kuten pitää olla tarpeeksi tehokas mikrofoni, mikä kykenee poistamaan tarkasti ylimääräistä melua sekä tunnistamaan luonnollisen äänen./1/

Mikrofonin pitää kääntää äänivärähtelyt sähkösignaaliksi, jonka jälkeen muutetaan digitaaliseksi signaaliksi käyttämällä äänikorttia. Jotta tietokone pystyy tunnistamaan eri äänteitä, ohjelmisto analysoi digitaalisignaalia ja äänteet muunnetaan sanoiksi. Ohjelma keskittyy kokonaisuuteen, koska se ei pysty erottamaan sellaisia sanoja, jotka tarkoittavat eri asioita, mutta kuulostavat samanlaisilta. /2/

Kokonaisuuteen keskitytään trigram-analyysin avulla käyttäen tietokantaa, jossa tarkastellaan todennäköisyyksiä että tietyn sanamäärän jälkeen tulee tietty sana. Koska puheentunnistuksessa käytetään kielenä suurimmaksi osaksi englannin kieltä, edelliseen 'trigram analyysi' esimerkkinä , kun käyttäjä sanoo "where am" kolmas sana hyväksytään "I":nä eikä "eye":nä, mitkä englantia puhuttaessa kuullostavat ihmisille samanlaisilta. Koska erilaisia puheentunnistulaitteita alkaa koko ajan tullemaan markkinoilla lisää, laitteiden tehot lisääntyvät ja enemmän ihmisiä puhuvat puheentunnistukselle, tunnistus tarkentuu koko ajan ja koneoppiminen parantuu samalla./3/

2.1 Historia

Ennen 2000-lukua ei puhuttu Applen Siristä eikä teknologian kehitys ollut näin pitkällä. Kaikki alkoi 1952 luvulla Audrey:stä, jonka kehitti Bell Laboratories.

Audrey ymmärsi vain numeroita 0-9, kuten aina uudet teknologiat ovat suuria ja kömpelöitä, ja se oli sijoitettu telineeseen korkeudeltaan 2 metriä /4/.

1960 luvulla saatiin kehittyneempi versio IBM:ltä nimeltään Shoebox mikä pystyi käsittelemään 16 sanaa englannin kielellä ja 0-9 numeroita. Myöhemmässä vaiheessa kyseistä parannettiin hiukan uusilla sanoilla sekä lisättiin pientä yhteenlaskua, mikä pystyi laskemaan ja näyttämään vastauksella./5/

Vähän suurempiin sanavarastoihin päästiin Harpylla 1970-1976, joka kehitettiin Carnegie Mellonin yliopistossa, Harpy pystyi käsittelemään yli 1000 sanaa ja joitakin säkeitä./6/

1980 luvun puolessa välissä teknologia sai potkua eteenpäin, 'HMM' muodossa eli hidden Markov model, mikä pystyi hahmottamaan sanoja paljon paremmin ja tästä tuli perusta monille tuleville yrityksille./7/

1990 luvulla yritykset vuoron perään yrittivät omilla ideoillaan. Yksi niistä oli Dragon, jonka teknologia oli tarpeeksi kypsä ja hinta 9000 dollaria. Teknologian myötä tuli sovelluksia, esimerkiksi vastaavien lelujen muodossa. Monia vuosia ongelmana oli, että piti joka puheen välissä olla pieni keskeytys ja jatko. Tähän ongelmaan tuli ohjelmisto Dragonilta vuonna 1997, joka kuunteli jatkuvasti ilman pysähtymistä, tunnistaen 100 sanaa minuutissa sekä paljon halvemmalla hinnalla 150 dollaria./8/

Vuonna 2001 Microsoft lisäsi windows xp:hen puheentunnistusominaisuuden office xp:lle. Siinä pystyi puhumaan ja puhe muuttui tekstiksi, vaatimuksena oli Service Pack 1, mikrofoni sekä sovellus, mikä tukee puheen tunnistusta./9/

Viime vuosien aikana teknologia on kehittynyt ja puheentunnistusta on kehitetty lukemattomilla puhetiedoilla. Puhetiedot tulivat myöhemmin Googlen julkaistua appinsä iphonelle nimellä Google Voice Search app. Tämän myötä sovellus oli

miljoonien ihmisten käytettävissä, keräämällä tietoja miljoonista hauista. Näin ollen pystyi ennustamaan, mitä käyttäjä sanoo. Vuonna 2010 google lisäsi tunnustuksen sen android puhelimiin sekä Chrome selaimen. Tämä ominaisuus linkittää jokaisen henkilökohtaiseen google tiliin, näin ollen tallentaa äänen sekä tunnistaa ja rakentaa jokaiselle oman puhemallin, minkä tarkkuus on paljon parempi jokaisen kerran jälkeen./10/

Vuonna 2011 Apple julkaisi oman version nimeltä 'Siri'. Siri oli aikansa vallankumous, jotain uutta. Apple itse kutsui sitä virtuaaliseksi avustajaksi tai älykkääksi virtuaaliassistentiksi.

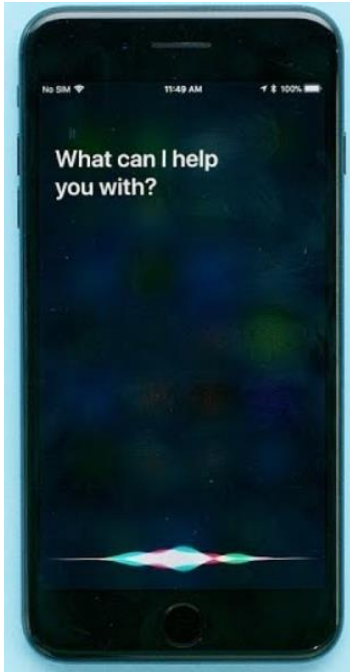
2.2 Virtuaali/Ääniavustajat

Vuoden 2011 loppupuolella tulleen Applen Sirin suosio yllätti suurinta osaa teknologia yrityksistä, kaikki halusivat mukaan jollakin omalla tavalla ja olla osallisena tähän mullistukseen. Taulukossa 1 on luettelo ääniavustajista.

Taulukko 1. Älyavustajat

Ääniavustajat	Julkaisu
Apple Siri	2011 Lokakuuta
Mixrosoft Cortana	2014 Huhtikuuta
Amazon Alexa	2014 Marraskuuta
Google Assistant	2016 Toukokuuta
Samsung Bixby	2017 Huhtikuuta

Siri



Kuva 1. Siri käyttöliittymä(/40/)

Sirille tärkeää oli luonnollisuus se, että ei tarvitse enää puhua kuin puhuisi robotille vaan nyt tuntuu kuin vastapuolena olisi ihminen. Yksi syy miksi Siriä sanottiin älykkääksi oli se, että Siri tajusi puheen kontekstin, esimerkiksi jos kysyt Siriltä, että muistuta minua, se kysyy takaisin että mitä haluat muistavasi ja myöhemmin perään kysyy vielä, että milloin haluat että muistuttaisin. Jos Sirille sanoo, että minulla on nälkä, Siri näyttää kaikkia ruokapaikkoja, jotka ovat lähetyvillä painamalla niitä tai sanomalla ääneen. Siri näyttää suoran reitin kyseiselle paikalle. Jos sanoo Sirille, että minulla on vessahätä, Siri näyttää lähimmät julkiset vessat ja samalla ohjaa sinne.

Siri myös tunnistaa ihmisiä jos he ovat kavereitasi, lähimmäisesi, kuten vaimo tai lapsesi, voit sanoa Sirille esimerkiksi, Siri muistuta minua soittamaan vaimolle kun olen kotona, kerrot Sirille kuka vaimo on ja myös missä asut, niin gpsn kautta Siri tarkistaa ja muistuttaa. Sen jälkeen kun kerran Sirille on kertonut keitä tietyt henkilöt

ovat, niin seuraavan kerran Siri muistaa heidät. Siri muistaa edellisiä asioita ja muistuttaa niistä, esimerkiksi jos sanoo Sirille että lisääsi kalenteriin tietyn asian, jos kyseisen päivän kalenteri on täynnä Siri sanoo, että on täynnä ja että haluaako poistaa jotain tai silti lisää./11/

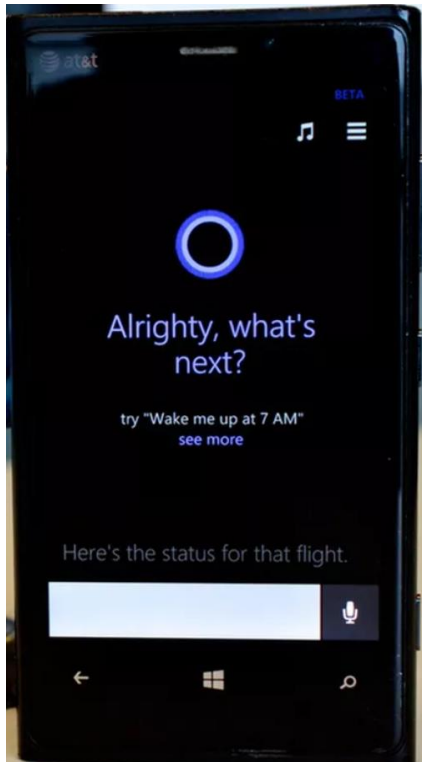
Sirillä pystyy ohjaamaan iphonia tuottamalla äänen, voi pyytää Siriä etsimään internetistä tietoa, lukemaan sähköposteja ääneen, soittamaan kavereille, kertomaan vitsejä, kysymään kartasta reittejä, vaihtamaan iphonen asetuksia, tekemään kalenteriin merkintöjä puolesta sekä paljon muuta. Näiden asioiden takia Apple kutsuu sitä ”henkilökohtaiseksi avustajaksi”. Ohjelma on asennettu valmiiksi jokaiseen iPhone 4s puhelimeen sekä uudempiin malleihin sekä myöhemmin Siri julkaistiin myös macOSille että watchOSille./12/

Google Now

Heinäkuussa 2012, Applen Sirin suosion myötä Google tuli mukaan omalla versiolla nimellä Google Now. Google Now oli rajoittuneempi kuin Siri monella eri tapaa esimerkiksi toisin kuin Apple Siri, Google Now ei pysty tuntemaan puhujaa paremmin joka kerran jälkeen, kun sille puhuu. Toinen ero näiden kahden välillä oli se, että Google Nowin kanssa ei pystynyt pitämään vuoropuhelua muodostamalla laitteen kanssa kaksisuuntaista keskustelua.

Tämän ongelman Google korjasi Google Assistantilla./13/

Cortana



Kuva 2. Cortana käyttöliittymä(/41/)

Ennen Googlen Assistantia vuonna 2013 kehittäjille tehty Lumia-puhelin oli ostettu eBaysta, josta appit oli merkattu z alkuisena kirjaimena "z":llä tarkoitetaan puhelinta, joka on tehty testiä varten, appien joukossa oli appi nimeltä "zCortana", joka oli Microsoftin vastaus Applen Sirille ja jota myöhemmin vuonna 2014 julkaistiin Windows Phone 8.1:lle nimellä Cortana./14/

Cortana on koodinimi, mikä tulee oikeastaan henkilöstä, jota on tehty viisaaksi Halo pelissä.. Alla kuva 3 Cortana hahmo Halo pelisarjassa. Cortana on keinotekoisesti tietokoneella tehty naishahmo, hahmoa esiintyy seuraavissa : Halo: Combat Evolved,

Halo 2, Halo 3, Halo: Reach, Halo 4, Halo 5: Guardians, hahmoa äänittää Halo peli sarjoissa että keinotekoisessa avustajassa Jen Taylor ./15/

Cortana toimii seuraavissa: Windows Phone, Windows, Microsoft Bands, Surface Headphones, Android, Xbox One, Skype, Ios, Windows Mixed Reality, Cyanogen OS.

Cortanalla pystyy tekemään melkein kaikkia asioita, mitä muillakin, kuten Siriällä.



Kuva 3. Cortana Halo pelisarjassa(/42/)

Amazon Alexa

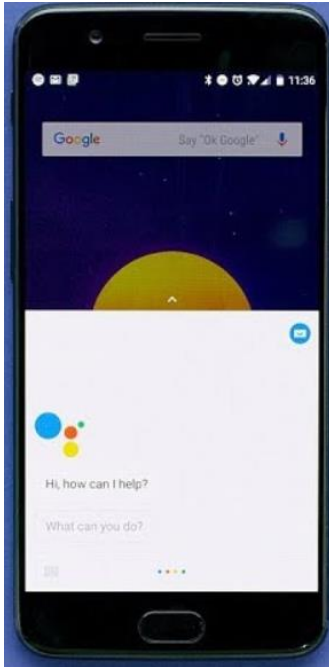


Kuva 3. Alexa käyttöliittymä(/40/)

Amazon toi oman version ääniavustajasta vuonna 2014. Sen mukana julkaistiin Amazon Echo-kaiuttimet, jotka omalla tavallaan mullistivat kaiutinmarkkinat, koska kyseessä oli älykaiutin, jossa oli ääniavustaja, josta puhutaan vähän myöhemmin lisää. Alexa toimii Fire Tabletissa, Fire TV:ssä sekä iotia tukevissa laitteissa./16/

Toisin kuin Google Assistantissa tai muissa ääniavustajissa, Amazonin Alexa toimii paremmin muissa kuin puhelimissa, Alexan saa ladattua älypuhelimiin, mutta kaikkia toimintoja ei voi käyttää./17/

Google Assistant



Kuva 4. Google Assistant käyttöliittymä(/40/)

Vuoden 2016 toukokuussa Google julkaisi Google Assistanin Googlen omassa Google I/O:ssa eli vuosittaisessa kehittäjätapauamisessa. Google Assistant paikkasi tärkeimpiä Google Nown puutteita, esimerkiksi kaksisuuntainen vuoropuhelun sekä oppii tuntemaan henkilöä, jonka kanssa puhuu./13/

Google Assistant pystyy hakemaan tietoja google-tilistä, varaamaan pöydän ravintolasta, esimerkiksi jos ravintola tukee ominaisuutta Google Assistantia voi pyytää varmaan pöydän tietystä ravintolasta ja tietynä aikana, avustaja soittaa ravintolaan jotta varmistaisi, että pyyntö voidaan toteuttaa. Kun pöytä on varattu, Google avustaja lisää sen kalenteriin ja muistuttaa siitä aikanaan./18/

Googlen Assistant toimii android puhelimissa ja Google Home älykaiuttimessa.

Samsung Bixby



Kuva 5. Bixby käyttöliittymä(/40/)

2017 Samsung julkaisi oman vastauksensa ääniavustajille Bixbyn muodossa. Bixby toimi alussa Galaxy S8, Note 8 puhelimissa ja bixbyä ei saa kuin vain Samsungilta. Tämän takia Bixbyn käyttäjiä ei ole niin paljon kuin muilla saman tapaisilla kuten Google Assistantilla. Tätä Samsung on yrittänyt muuttaa Bixby 2:lla, mikä avaa bixbyä kaikille laitteille. Bixby on heikompi kuin kilpailijansa ja hitaampi, Samsung on luvannut parantaa Bixbyä tulevaisuudessa. /19/

2.3 Smart speakers

Amazonin julkaistua älykaiuttimen vuonna 2014 ja sen suosion myötä valmistajat, kuten Google vuonna 2016 ja Apple vuonna 2018, julkaisivat omat kaiuttimensa nimeltä Google Home ja Apple HomePod. Älykaiuttimissa on samat ääniavustajat, mitä puhelimissakin on, kaiuttimia pidetään seuraavana isona asiana teknologian saralla.

Älykaiuttimilla saadaan paljon enemmän aikaan. Toimintoja, kuten tv kanavien ohjaaminen, youtubesta tai internetistä videoiden valitseminen, pysäyttäminen toistaminen, kelaaminen kaikki vain käskyllä. Päivän uutisia voidaan kertoa heti aamulla, kellon tai sään ilmoittaminen, tiedon haku internetistä, kodin laitteiden ohjaus, kuten valojen päälle tai pois laittaminen, valojen värin muuttaminen tai esimerkiksi tilaaminen pizzan pelin avulla./20/ Älykaiuttimiin on helppo siirtyä, kun kerran on ollut kokemusta ääniavustajista. Tässä opinnäytetyössä käytetään Google Home-älykaiutinta.

Amazon Echo



Kuva 7. Amazon Echo(/43/)

Google Home



Kuva 8. Google Home(/44/)

2.4 Ääniavustajien hyödyt

Ääniavustajat helpottavat ihmisten elämää huomattavasti, asiat hoituvat aiempaa nopeammin ja kätevämmiin. Kuvittele esimerkiksi istuvasi sohvalla, television kanavien vaihto, huoneitten valojen pois laittaminen ja monien muiden asioiden hoitaminen pelkästään istumalla sohvalla, puhelimen asetuksen muuttaminen ilman, että painaisi jokaista erikseen. Sokeat ihmiset voivat hakea tietoja internetistä pelkästään kysymällä ääniavustajalta, robotiikassa tai sairaat ihmiset, jotka eivät jostain syystä pääse liikkumaan. Ääniavustajat ovat jatkuvasti kuulolla, että ei tarvitsisi puhelimeen tai kaiuttimeen koskea, esim Google Homen saa päälle sanomalla "Hey Google." Tai Sirin "Hey Siri."./21/

2.5 Ongelmat

Ääniavustajat saavat aikaan, että ihmiset liikkuvat paljon vähemmän ja asiat hoituvat vähemmällä vaivalla. Esimerkiksi enää ei mennä huoneesta toiseen sulkemaan huoneen valoja tai pihan valoja, internetiin ei tarvitse lähteä sivusta toiseen, jotta löytäisi tietyn uutisen vaan kysymällä se löytyy heti. Ihmiset tulevat laiskemmiksi. Ongelmana on myös se, että ääniavustajat eivät tue vielä monia kieliä, yleinen kieli kaikilla on englanti. 27 prosenttia ihmisistä, jotka omistavat ääniavustajaa tukevan laitteen, käyttävät ääniavustajia ja kun ääniavustajat ovat jatkuvasti kuulolla, kun avustajalle antaa käskyn se lähtee kyseisen yrityksen palvelimiin johonkin päin maailmaa, jota siellä tulkitaan ja sen jälkeen vastaus tulee käyttäjälle./22/

Alle 50 prosenttia virtuaaliassistenttien käyttäjistä pelkäävät heidän yksityisyytensä puolesta ja, että heitä kuunnellaan jatkuvasti. Toukokuussa 2018 nainen valitti, että Amazonin Echo nauhoitti hänen miehensä kanssa käytyä yksityiskeskustelua. Yritykset kuuntelevat käyttäjiensä keskustelua, jotta voisivat parantaa laatua ja sitä, että ääniavustaja pystyisi tulevaisuudessa antamaan tarkempia ja parempia vastauksia./23/

2.6 Tulevaisuus

Ääniavustajien tulevaisuus näyttää päivä päivältä kirkkaammalta, koska puhuminen on helpompaa ja nopeampaa kuin kirjoittaminen, ääniavustajan tulevaisuus on sidoksissa esineiden internetin (IoT, esineet jotka ovat yhteydessä internettiin) kanssa, esimerkiksi autossa uuden viestin kuuntelemisesta, applikaatioiden ohjaamisesta tai musiikin vaihtamisessa, tai kotona pakastimen ohjauksesta television ohjaamiseen. Kaikki laitteet mitkä yhdistetään internetiin tavalla tai toisella tulevat tukemaan ääni ohjausta./24/

3 PÄÄTTÖTYÖSSÄ KÄYTETYT TEKNIIKAT

Tässä opinnäytetyössä ääniavustajana käytetään Googlen Google Assistantia

3.1 HTML5

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="utf-8">
6     <title>Google Assistant</title>
7     <style>
8     </style>
9 </head>
10
11 <body>
12     <header> </header>
13     <nav> </nav>
14     <article>
15         <section>
16         </section>
17     </article>
18     <aside> </aside>
19     <footer> </footer>
20
21 </body>
22
23 </html>
```

Esimerkkikoodi 1. Html5, jossa elementit ilman Javascriptiä ja tyylitiedostoa.

HTML5 tulee englannin kielestä HyperText Markup Language, se on viimeisin versio HTML:stä ja sisältää elementtejä ja tageja. Sillä tehdään web-sivuja, yhdistämällä tekniikoita, kuten CSS, Javascript, xml. HTML5 sisältää uusia

elementtejä, jotka parantaa ja helpottaa koodaamista, elementtejä, kuten : <article>, <header>, <section> sekä <footer> ja <aside>, myös videon ja äänen lisäys omaan kotisivuun sekä samassa sivussa hiirellä ottaminen ja pudottaminen.

Kuvassa, esimerkki HTML5sta./25/

3.2 Javascript

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <title>Google Assistant</title>
7  <style>
8  </style>
9  </head>
10
11 <body>
12     <header> </header>
13     <nav> </nav>
14     <article>
15         <section>
16         </section>
17     </article>
18     <aside> </aside>
19     <footer> </footer>
20
21     <script>
22         alert( 'Hei, tervetuloa' );
23     </script>
24 </body>
25
26 </html>
```

Esimerkkikoodi 2. Html5, jossa yhdistetty JavaScript.

JavaScript on ohjelmointikieli, jota käytetään, kun halutaan ohjata sivujen käyttäytymistä, esimerkiksi ponnahdusikkunan luonti, näytetään viestiä käyttäjälle, linkin painaminen, joka johtaa tiettyyn sivuun. Yleisimmin selainpuolella käytetään JavaScriptiä, JavaScript on funktionaalista ja oliomaista ohjelmointia. Nykyään JavaScriptillä pystyy tekemään selainpohjaisia applikaatioita sekä myös ohjelmoimaan server puolella (backend) node.js ympäristössä. Kuvassa JavaScript lisätään html:ään script elementillä `<script></script>` jotka laitetaan bodyn sisälle `<body> <script></script> </body>`, kuvan esimerkissä ilmestyy ponnahdusikkuna jossa lukee 'Hei, tervetuloa'./26/

3.3 CSS

CSS tulee englannin kielestä Cascading Style Sheets, CSS määrittelee miltä HTML tai HTML5 sivut näyttävät, taustaväri, fonttien koko, sivun pituus, leveys, kuvien kokojen muuttaminen ja paljon muuta. Tyylisivut tallennetaan .css muodossa, CSS on kätevä ja säästää paljon aikaa, esimerkiksi tallentamalla erilliseen tyyli tiedoston sitä voi käyttää niin monessa sivussa kuin haluaa.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5   body {
6     background-color: blue;
7   }
8
9   h1 {
10    color: white;
11    text-align: center;
12  }
13
14
15 </style>
16 </head>
17 <body>
18
19 <h1>Google Assistant</h1>
20
21 </body>
22 </html>
```

Esimerkkikoodi 3. css style elementin sisällä (engl. Internal CSS)

Kuvassa esimerkkikoodi 3 tulostaa sivulle valkoisen tekstin 'Google Assistant', joka on sijoitettu keskelle sivua ja taustaväriksi on laitettu sininen./27/

3.4 NodeJS

Vuonna 2009 toukokuun 27 päivänä, Ryan Dahl kehitti NodeJSn. NodeJS voidaan kutsua javascriptiksi, joka pyörii palvelinpuolella. Ennen NodeJS:aa jos halusi ohjelmoida palvelinpuolella, piti opiskella jotain muuta kuin javascriptiä,

esimerikiksi php:ta. Node.js:n tulon myötä ohjelmoijien urakka väheni, nyt ei tarvitse oppia uutta kieltä vain sekä palvelin että selaimenpuolella on sama kieli (javascript).

Node.js käyttää Googlen Chrome V8 moottoria javascriptin pyörittämiseen. NodeJS:lla pystyy tekemään skaalautuvia sekä reaaliaikaisia web-sovelluksia. Reaaliaikaisella tarkoitetaan esimerkiksi pelejä, jotka pelataan tietokoneen selaimessa. Tärkeänä asiana nodejs:ssa voi pitää sitä, että se käyttää myös asynkronisia sekä tapahtumapohjaisia ohjelmia./28/

```
var http = require('http');
http.createServer(function (req, res) {
  res.write('Hey Google!');
  res.end();
}).listen(8080);
```

Esimerkkikoodi 4. Node.js koodi 'Hey Google!'.

Kuvassa 4: http moduuli otetaan käyttöön require funktiolla, palvelimen puolelle tehdään oma objekti, kirjoitetaan vastaus selaimelle ' Hey Google' ja vastauksen lähetys pysäytetään.

3.4.1 NodeJS Moduulit

NodeJS moduulit ovat javascript-kirjastoja, tiedostoja, jotka sisältävät monia toimintoja, funktioita, joita voi käyttää Node.js:ssä. Moduulin voi lisätä require('') funktiolla, joka sisältää nimen./29/

```
var http = require('http');
```

3.5 Fulfillment

Kun dialogflowssa haluaa yhdistää sovelluksen muihin alustoihin, kuten Facebook Messenger, Slack tai Google Assistant tarvitaan käytettäväksi Fulfillment jotta voidaan yhdistää oman palvelun agenttiin. Jokaisessa intentissa voi erikseen laittaa Fulfillment päälle. Jos intentissa ei olla käytetty Fulfillmentia, vastaa Dialogflow valmiilla vastauksilla./30/

3.6 Webhook

Webhook, myös kutsuttu web callback, webhookilla pystyy hakemaan reaaliaikaista tietoa sekä toimittaa tietoja muille sovelluksille sinä aikana kun muutoksia tapahtuu eli tiedot saadaan heti./31/

3.7 Firebase Realtime Database

Firebase on reaaliaikainen tietokanta, jonka omistaa Google vuodesta 2014. Tietokanta on pilvipohjainen ja tiedot tallennetaan json muodossa. Kaikki tiedot menee pilveen, synkronoi tiedot kaikkien laitteiden välillä millisekunneissa, webistä mobiiliin./32/

3.8 Dialogflow

Vuonna 2016 Google osti yrityksen nimeltä API.AI, jonka Google nimesi Dialogflowksi seuraavan vuoden lokakuussa. Dialogflowlla tarkoituksena on tehdä luonnollinen keskusteleva käyttöliittymä verkkoselaimille, koska kyseessä on selain, niin se toimii puhelimissa, kaikissa selaimissa sekä IoT-laitteissa. Alustan pystyi erottamaan muista siitä, että se käyttää tekoälyä ja koneoppimistä hyödykseen ja näin ollen saa parempia tuloksia aikaan. Tämä helpottaa ohjelmoijaa,

koska asioista, kuten koneoppimisesta, tekoäly ja vaikeita kielenkäsittelyitä ei tarvitse miettiä, Dialogflow hoitaa niitä ohjelmoijan puolesta./33/

Dialogflown kannalta kaksi tärkeää asiaa, joilla pystyy ohjaamaan liittymää ovat : Intents ja Entities eli aikeet ja entiteetit. Aikeet kuvastavat sitä, että mihin käyttäjä tähtää tai mitä tavoittaa, esimerkiksi pysäytä musiikki tai soita äidille.

Aikeilla voi olla yksi tai kymmeniä entiteettejä, entiteetit ovat ne yksityiset kohdat aikeissa esimerkiksi äidille soittaminen tiettyyn aikaan tai tietty määrä joukkuetta. Ennen entiteetin nimeä tulee kirjoittaa merkki ”@”./34/

3.8.1 Dialogflow agentin luonti

Kun halutaan luoda kehittäjätili Dialogflow:hon tarvitaan ensiksi luoda Google tili (<https://accounts.google.com/signup/v2/webcreateaccount?flowName=GlifWebSignIn&flowEntry=SignUp>), koska kyseessä on googlen laitteet mikäli haluaa google assistantin toimivan kaikissa googlen palveluissa kuten puhelimissa ja älykaiuttimissa tarvitaan niihin kirjautua samalla google tilillä. Jos omistaa google tilin ennestään voi käyttää sitä. Tämän jälkeen Dialogflow sivulla kirjaututaan google tilillä sisään.

factory CREATE

DEFAULT LANGUAGE ⓘ

English – en ▼
Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+3:00) Europe/Moscow ▼
Date and time requests are resolved using this timezone.

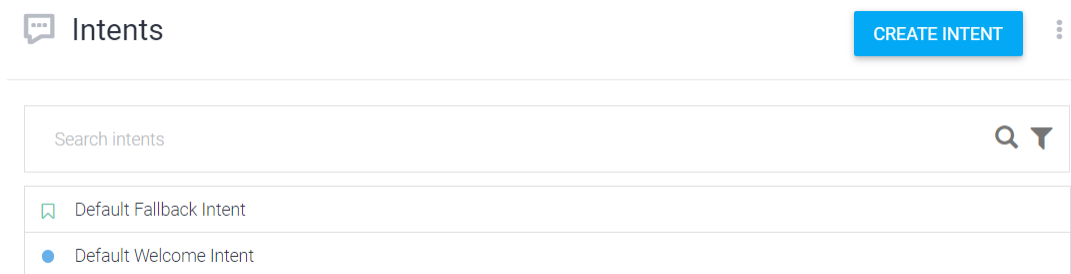
GOOGLE PROJECT

Create a new Google project ▼
Enables Cloud functions, Actions on Google and permissions management.

Kuva 9. Agentin luonti Dialogflow:lla

Kuten kuvasta näkyy, annetaan google agentille nimi. Tässä tapauksessa on annettu factory ja valitaan oikealta oikea aikavyöhyke ja painetaan CREATE-nappulaa.

Avautuu uusi ikkuna, josta näkyy, että meille on valmiiksi tehty kaksi Intentiä Default Welcome Intent ja Default Fallback Intent



Kuva 10. Valmiit Intentit

Kun henkilö puhuu assistentille asioita, joita ei ole Intentissä ja se ei tajua sitä niin silloin kutsutaan automaattisesti Default Fallback Intent. Vastauksena tulee kuten olettaa esimerkiksi : anteeksi, voitko toistaa tai en tajunnu voitko sanoa vielä kerran ? näitä vastauksia on valmiina monia, Dialogflow looppaa jokaista kerran läpi siten, että ei tule sama vastaus sen jälkeen valitsee satunnaisesti.

Kun käyttäjä alkaa puhumaan assistentille, automaattisesti kutsutaan aina Default Welcome Intent. Default Welcome Intent sisältää harjoitusfraaseja (engl. Training phrases), joilla voidaan opettaa avustajaa että kun tietyllä tavalla kutsutaan sitä niin se katsoo jos sitä kutsunta tapaa on fraaseissa. Avustaja vastaa muuten kutsutaan Default Fallback Intentiä.

Training phrases ⓘ

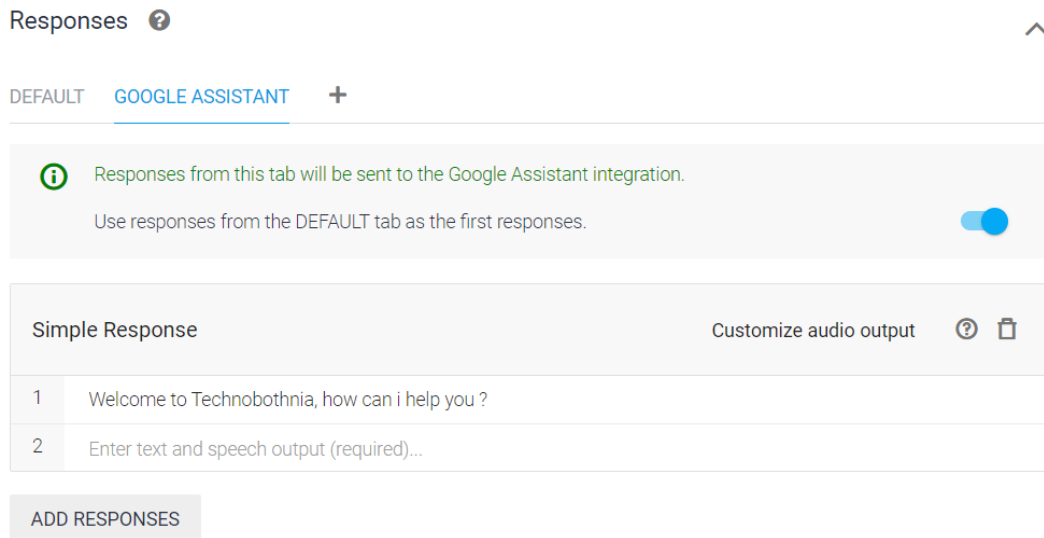
Search training phrases 🔍 ^

” Add user expression
” just going to say hi
” heya
” hello hi
” howdy
” hey there
” hi there
” greetings

Kuva 11. Valmiita harjoitusfraaseja

Kuvassa valmiita harjoitus fraaseja, joita avustaja kuuntelee ja reagoi sen mukaan.

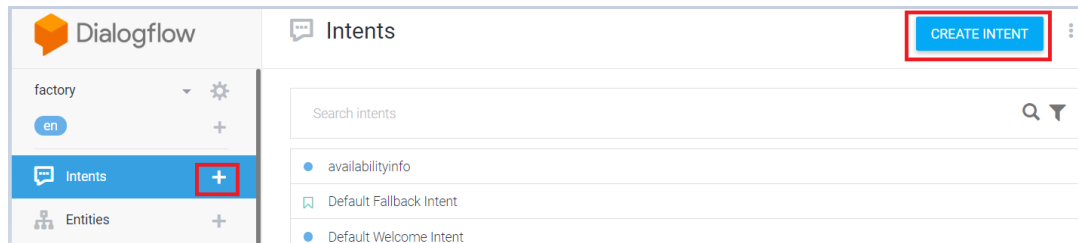
Kun fraasi ja se mitä käyttäjä sanoo täsmäävät, reaktio tulee vastauksista (engl. Responses) , Vastauksissa tukea voi lisätä esim. seuraaviin: Viber, Google Assistant Facebook messenger, Skype, Kik



Kuva 12. Google assistantin valinta

Tässä työssä käytettiin Google Home kaiutinta, joka toimii Google Assistant avustajalla. Näin ollen valitsin Google Assistant ja vastaa 'Welcome to Technobothnia, how can i help you?'. Kun painaa 'ADD RESPONSE' vastaus voidaan saada monilla eri tavoilla. Esimerkiksi jos haluaa saada kuvan vastauksena 'Media Content', listana 'List', linkkinä 'Link Out Suggestion', taulukkona 'Table Card', korttina 'Basic Card'. Tässä työssä käytetyssä 'simple response', on monia tapoja, jotka eivät toimi oikein Google Home kaiuttimella, koska ei ole näyttöä. Puhelimissa, joissa on Google Assistant avustaja onnistuu eri tavoilla.

3.8.2 Intent luonti



Kuva 13. Intentin luonti

Seuraavaksi luodaan intent, minkä luonti onnistuu kahdella tavalla painamalla '+' tai 'CREATE INTENT' oikealla ylhäällä nurkassa. Intentille annetaan nimi, tässä tapauksessa kuten kuvassa näkyy 'availabilityinfo', tässä intentissä lisätään fraasi esim. 'What is availability' ja vastauksena että mitä se tarkoittaa.

● availabilityinfo
🔖 Default Fallback Intent
● Default Welcome Intent
● factory
● factoryinfo
● factorynumbers
● oeecalculatation
● OEEInfo
● oeenumbersb
● oeenumbersg
● oeenumberstbest
● performanceinfo
● qualityinfo

Kuva 14. Projektin kaikki intentit

Projektissa on 3 kappaletta tehdasta 'Factory 1', 'Factory 2' ja 'Factory 3' jokainen factory sisältää : OEE, Availability, Performance, Quality. Kuten kuvassa näkyy meillä on 13 intentiä. Info loppuiset intentit kertovat, miten saadaan kyseiset intentit pyörimään ja vastaamaan, että mitä kyseinen intent tarkoittaa. 'factorynumbers' intent vastaa tehtaiden määrästä, 'oeecalulation' vastaa siitä, että kuina OEE on laskettu, 'oenumbersb' kertoo, että mitkä ovat huonot OEE-arvot, 'oenumbersg' hyvät OEE-arvot, 'oenumbersbest' Parhaat OEE-arvot, 'factory' intentit:issä on 'Action and parameters'-osio, mikä toimii yhdistävänä tekijänä intentien ja entitien välillä./35/

Action and parameters

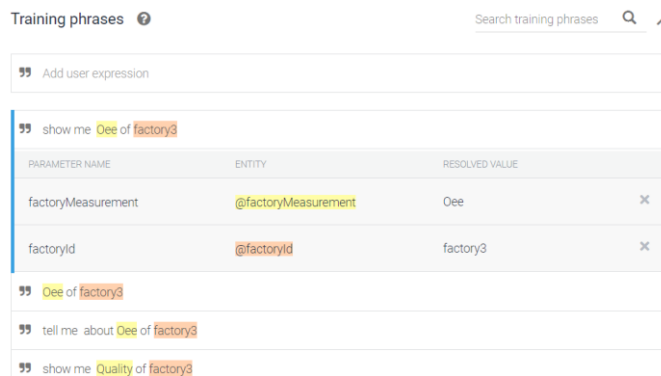
REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	factoryId	@factoryId	\$factoryId	<input type="checkbox"/>	What is the fac...
<input checked="" type="checkbox"/>	factoryContent	@factoryContent	\$factoryContent	<input type="checkbox"/>	Which measure me...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	–

[+ New parameter](#)

Kuva 15. Parametrit

Kun intentiteetti on sama kuin mitä käyttäjä lausuu, Dialogflow käy käyttäjän lauseen läpi ja poimii tiettyjä arvoja lauseesta ja tarjoaa niitä parametreina. Parametreille pitää antaa nimi, tässä tapauksessa 'factoryId', 'factoryContent'. 'REQUIRED'-osio, mikä on sinisellä värillä tarkoittaa, että intenti ei toimi ilman vastaavaa parametria. Tähän lisätään entity, mikä on tehty ennestään että vastaisi tätä parametria, jotta voisimme käyttää sitä, entityn lisäyksestä (3.7.3 Entity luonti). Lopuksi on 'PROMPTS', mikä tarkoittaa sitä, että kun käyttäjä lausuu jotain, joka ei sisällä kyseisiä parametreja, agentti kysyy käyttäjältä

lisätietoja. Esim. Projektissa meillä on 3 tehdasta, jos käyttäjä ei sanoa suoraan 'tell me about factory 3 quality', vaan sanoo vain 'factory 3', agentti kysyy promptin välityksellä ' Which measurement? Please choose one: Quality, Performance, Availability, OEE.'/36/



Kuva 16. Factory intentin harjoitus fraasit

Factory intents sisältää harjoitusfraaseja, jossa on kysely arvoja eri tehtailta eri synonyymeilla, jotta käyttäjän ei tarvitse kysyä vain tietyllä tavalla. Esimerkiksi. 'Show me Oee of factory3', 'Oee of factory3' tai 'tell me about Oee of factory3' kaikki kysyvät samaa mutta eri tavoilla. Mitä enemmän käyttäjä kysyy, sitä parempia ja tarkempia vastauksia saa, koska Dialogflow käyttää Googlen koneoppimista hyväkseen.

'Oee' ja 'factory3' ovat saanneet värin keltainen ja punainen, mikä johtuu siitä, että kyseiset fraasit sisältävät parametrejä, jotka täsmäävät äskeisiin parametreihin, jotka luotiin näin ollen ne on tunnistettu.

Jotta saataisiin jokaisen tehtaan arvot tietokannasta, webhook pitää laittaa päälle tälle intentille Fulfillment kohdalta.

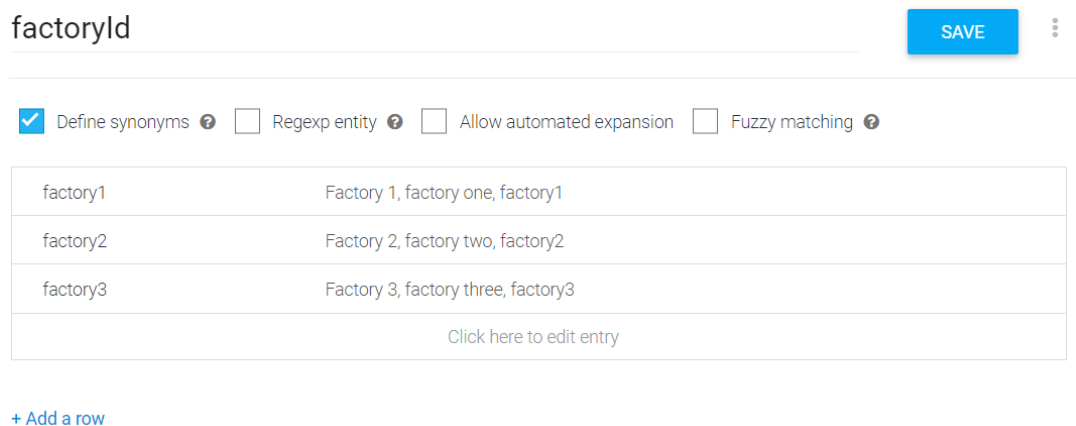


Kuva 17. Webhookin päälle laittaminen

Rullataan alas sivulla ja Fulfillment osiossa painetaan 'Enable webhook call for this intent' kuten kuvassa.

3.8.3 Entity luonti

Entityn luonti onnistuu samalla tavalla kuin intentin luonti eli kahdella tavalla painamalla '+' tai 'CREATE ENTITY' mikä on oikealla ylhäällä nurkassa, luodaan entity 'factoryId' nimellä.



Kuva 18. FactoryId entityn viittaus arvojen ja synonyymien laitto

Painetaan 'Define synonyms', jotta entityä voidaan kutsua eri synonyymeilla niin kone tajuaa paremmin, että mitä haetaan. Sen jälkeen klikataan 'Click here to edit

entry' johon 'Enter reference value' kohdalle lisätään factoryt, ja 'Enter synonym' kohdalle synonyymit kuten kuvassa.

factoryContent SAVE ⋮

Define synonyms ? Regexp entity ? Allow automated expansion Fuzzy matching ?

Availability	Availability, availability
OEE	OEE, oee
Performance	Performance, performance
Quality	Quality, quality
Click here to edit entry	

Kuva 19. FactoryContent entityn viittaus arvojen ja synonyymien laittaminen

Kuvassa on sama toistettu 'factoryContent' entitylle, 'reference value' ovat 'Availabilty', 'OEE', 'Performance', 'Quality'.

3.8.4 Fulfillment

Koska sovellus halutaan yhdistää Google Assistantiin omalla koodilla, painetaan fulfillment osiota, mikä on sivun vasemmalla puolella 'Entities'-osion alla,

⚡ Fulfillment

Webhook DISABLED 🔌

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

Inline Editor (Powered by Cloud Functions for Firebase) ENABLED 🔵

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

index.js package.json 📄

Kuva 20. Inline editorin päälle laittaminen

Fulfillment osiossa on 'Inline Editor', jossa voidaan ohjelmoida omaa ohjelmaa toimivaksi tässä projektissa, 'Inline editorissa' toimii pelkästään node.js, painetaan 'Enable', jotta voidaan ohjelmoida.

```

index.js package.json
1 'use strict';
2 const functions = require('firebase-functions');
3 const {WebhookClient} = require('dialogflow-fulfillment');
4 const {Card, Suggestion} = require('dialogflow-fulfillment');
5     var admin = require('firebase-admin');
6
7     admin.initializeApp({
8     credential: admin.credential.cert({
9     projectId: '...',
10    clientEmail: 'firebase-adminsdk-...@...iam.gserviceaccount.com',
11    privateKey: '-----BEGIN PRIVATE KEY-----\n...',
12    }),
13    databaseURL: 'https://factory-...firebaseio.com'
14 });

```

Kuva 21. Firebase SDK käyttöönotto

Kuvassa koodi suoritetaan "strict mode"-tilassa, otetaan moduulit käyttöön require funktiolla, jotta tiliä voidaan vahvistaa ja todistaa että on oikea tili kyseessä. Jotta päästäisiin käyttämään Firebase-palveluja, tarvitaan tehdä oma yksityinen avain tiedosto, jonka jokainen tili saa vain kerran eikä sitä saa hukata, eikä antaa muille, avain tiedosto tulee JSON-muodossa. 'Firebase Admin SDK' ladataan koneelle ja se sisältää 'privateKey' ja 'clientEmail'. 'databaseURL'. Kuvassa peitetyt kohdat ovat jokaiselle yksityisiä ja sen takia peitetty. 'databaseURL' kohdalle laitetaan, tietokannan URL minkä saimme kohdassa (3.7.1)

```

index.js  package.json
15 process.env.DEBUG = 'dialogflow:debug'; |
16 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
17   const agent = new WebhookClient({ request, response });
18   console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
19   console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
20
21   function welcome(agent) {
22     agent.add(`Welcome to my agent!`);
23   }
24
25   function fallback(agent) {
26     agent.add(`I didn't understand`);
27     agent.add(`I'm sorry, can you try again?`);
28   }

```

Kuva 22. Welcome ja fallback funktiot

Teemme mahdolliseksi virheenkorjauksen, funktion siirtoa varten oleva nimi pitää olla 'dialogflowFirebaseFulfillment', joka sisältää aloitus ja fallback funktiot

```

index.js  package.json
29 function factoryInside (agent) {
30   const factoryId = agent.parameters.factoryId;
31   const factoryContent = agent.parameters.factoryContent;
32   var tietokanta = admin.database();
33   var osoitus = tietokanta.ref(`values/${factoryId}/${factoryContent}`);
34   return osoitus.once("value")
35     .then( snappi => {
36       var seuraus = snappi.val();
37       agent.add(` ${factoryId}'s ${factoryContent} is ${seuraus}`);
38     })
39     .catch( virheilmoitus => {
40       agent.add('oops, got problem');
41       console.error( virheilmoitus );

```

Kuva 23. FactoryInside funktio

Kuvassa, Entityt 'factoryId' ja 'factoryContent' liitetty samannimiisiin constantteihin, 'osoitus' variaabelilla viitataan tietokantaan.'catch' virheen käsittely.

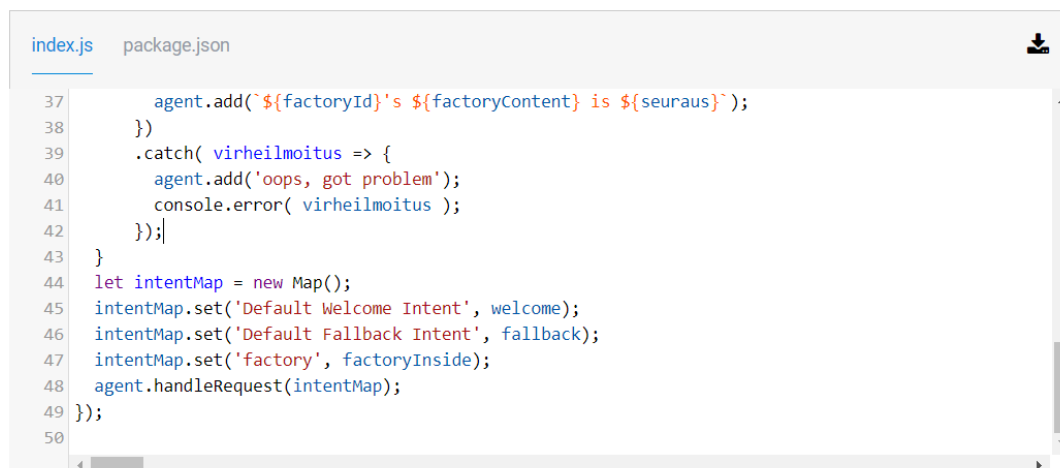
```

37 agent.add(` ${factoryId}'s ${factoryContent} is ${seuraus}`);

```


Kuva 24. Arvon tulostus

Kuvassa, kyseiset factoryt mitä käyttäjä kysyy ja sen arvojen tulostus.



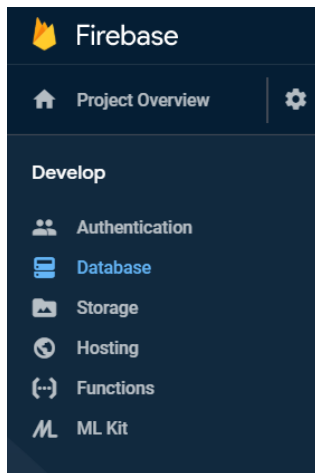
```
index.js package.json
37     agent.add(`${factoryId}'s ${factoryContent} is ${seuraus}`);
38   })
39   .catch( virheilmoitus => {
40     agent.add('oops, got problem');
41     console.error( virheilmoitus );
42   });
43 }
44 let intentMap = new Map();
45 intentMap.set('Default Welcome Intent', welcome);
46 intentMap.set('Default Fallback Intent', fallback);
47 intentMap.set('factory', factoryInside);
48 agent.handleRequest(intentMap);
49 });
50
```

Kuva 25. Funktion suoritus

Kuvassa funktioiden welcome, fallback, factoryInside suoritus.

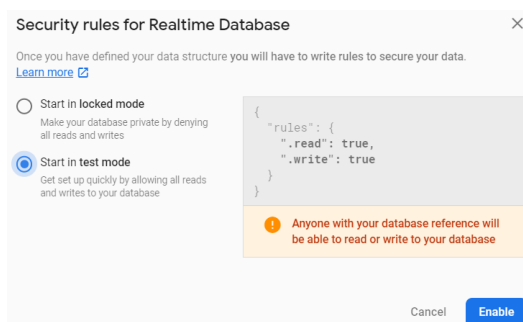
3.9 Realtime Database luonti

Luonti onnistuu menemällä Firebase omalle kotisivulle, josta painetaan 'GO TO CONSOLE' joka menee automaattisesti sisään Google tunnuksella, koska aikaisemmin kirjauduttiin Dialogflowin google tunnuksilla. Valitaan oma projekti kohdasta 'All Firebase projects' nimeltä 'factory'.



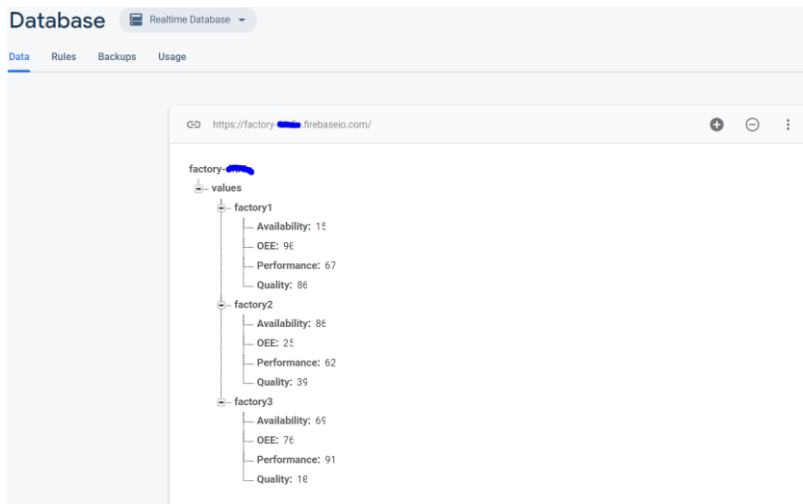
Kuva 26. Firebase Developer osiot

Develop osiosta painetaan 'Database', skrollataan alas ja 'Realtime Database' kohdalla painetaan 'Create database'.



Kuva 27. Tietokannan turvallisuussäännöt

Seuraavaksi turvallisuussäännöt avautuvat tietokannalle. Valitsemme 'start in test mode', mikä antaa luvan lukemiselle ja kirjoittamiselle tietokantaan, hyväksytään painamalla 'Enable'.

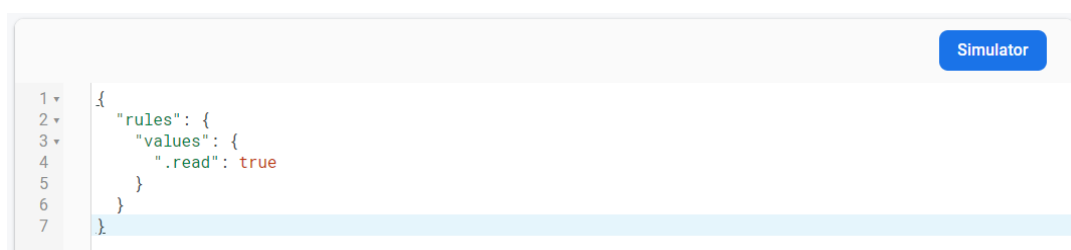


Kuva 28. Tietokanta (Database)

Database ikkunassa on 'Data' jossa tärkein itse tieto tulee tänne, 'Rules' turvallisuussäännöt voi muuttaa täältä. 'Backups' on varmuuskopiointi sen tallentaminen ja palauttaminen tapahtuu tässä osiossa. 'Usage' ilmoittaa että monta henkilöä ovat samaan aikaan yhteydessä kyseisessä tietokannassa, kuinka paljon tietoa on tallennettu, sekä antaa tiedon yleisestä suorituksesta tietokannassa./37/

'Data osiossa' painetaan '+' mistä voi lisätä nimen ja arvon, projektin tapauksessa lisätään 'values' ja kolme 'factory' sekä niiden sisältö ja arvot kuvan mukaan hyväksymistä varten painetaan 'Add'. Realtime Database antaa linkin kyseiselle tietokannalla, jota referoimalla pääsee käsiksi tietokantaan, koska linkit ovat jokaiselle yksityisiä, on linkki peitetty.

Painetaan 'Rules'-osiota ja vaihdetaan turvallisuussääntöjä.



Kuva 29. Tietokannan turvallisuussäännön editointi

Halutaan lisätä käyttäjälle oikeus lukea polkua /values/ .

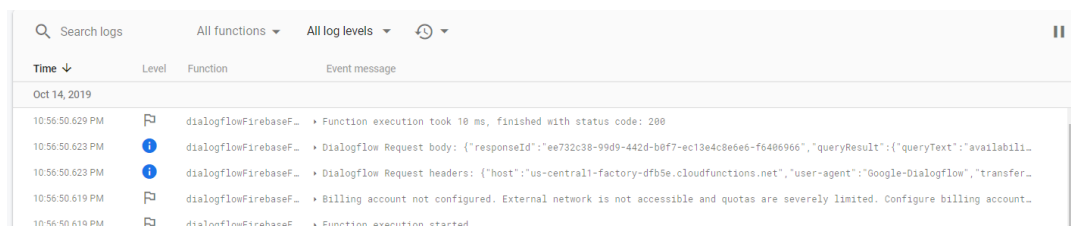
Sääntötyyppejä ovat : .read, .write, .validate, .indexOn

'validate' Kertoo miltä oikein asetettu tai järjestetty arvo näyttää

'indexOn' liittää lapsen indexille jotta voi nopeuttaa kyselyjä ja järjestää haku halutulla tavalla. /38/

3.9.1 Mahdollisia ongelmia

Jos tulee ongelmia tietokannan puolella, Develop osion Functions kohdalla on 'Logs'-osio kuvassa, josta näkee ongelman syy ja päivämäärä.

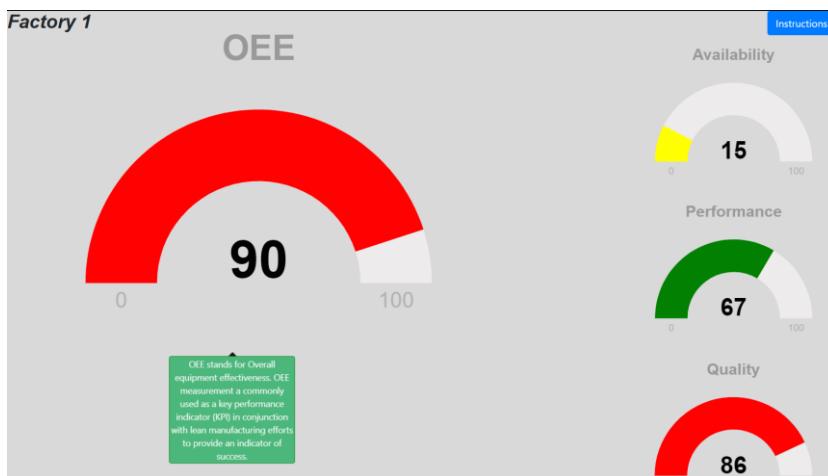


Kuva 30. Firebase Logs

4 SOVELLUKSEN TOTEUTTAMINEN WEB-SIVUNA

Sivun luonnissa käytetään Javascript pluginia(laajennus) nimeltä 'JustGage', JustGage:lla voi tuottaa animaatio mittareita, sekä vektorin piirtoa. Se toimii kaikilla selaimilla.

Mittaria muutetaan hieman, lisäämällä animaatiota, vihjelaatikkoa ja haetaan meidän tietokannasta arvot. /39/



Kuva 31. Factory 1 käyttöliittymä

Projektin käyttöliittymä kuvassa 31. Työ sisältää 3 kappaletta tehdasta (Factory 1, Factory 2, Factory 3), jokainen tehdas näyttää samalta mutta eri arvoilla, käytetty bootstrap frameworkia (ohjelmistokehys), jotta mukautuisi mobiili näytöille sopivaksi.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Factory 1</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <link rel="stylesheet" href="animations.css" type="text/css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
```

Kuva 32. Html lähdekoodi head element

Kuvassa 'bootstrap.min.css' on css 'instruction' -osiolle. 'animations.css' on animaatiota varten, kun sivulle tulee animoidaan kyseiset mittarit eri tavoilla, 'jquery.min.js' on itse mittaria varten. Ilman tätä mittaria ei näy, 'popper.min.js' tarvitaan tooltipiä varten, kun hiiren kursorin vie mittarin päälle, ilmestyy vihreä tekstikenttä, mikä selittää, että mitä kyseinen meinaa.

```

13  /* Tooltip */
14  .tooltip > .tooltip-inner {
15    background-color: #3CB371;
16    color: white;
17    border: 1px solid green;
18    padding: 1px;
19    font-size: 14px;
20  }

```

Kuva 33. Css Tooltip

Selitettävä tekstikenttä, johon laitettu teksti, tausta väri sekä tekstin koko.

```

21  .oee {
22    float: left;
23    height: 550px;
24    width: 100%;
25    display: inline-block;
26  }
27  .gauge {
28    height: 250px;
29    display: inline-block;
30    width: 100%;
31  }
32  p {font: italic bold 28px arial,serif ;
33    margin-top: 0px;
34    float: left;
35    font-size: 30px;}
36
37  body {margin: 100px auto;
38    text-align: center;
39    background-color: rgb(217, 217, 217);
40    margin-top: 0px;}

```

Kuva 34. Css tausta ja mittarit

Kuvassa 'oee'-mittarin koko, leveys, sijainti, 'gauge' muut mittarit niiden korkeus ja leveys, 'p' factory tekstin fontti, sijainti ja koko, 'body' sivun sisältö, sijainti ja taustaväri.

```

42  .ohjeet{text-align: right;}
43
44  @media only screen and (orientation: landscape) {
45
46    .gauge {
47      width: 30%;}
48
49    .oee {width: 50%;
50      margin-left: -156px;
51      margin-top: -47px;}
52  }

```

Kuva 35. Css landscape

Kuvassa 'ohjeet' on 'instruction'-osion sijaintia varten sekä sen sisällön tekstien sijainti, kun puhelinta kääntää mittarien koko ja sijainti muuttuu.

```
<div id="gg1" class="oeo animated growIn" class="test"
href="#" data-toggle="tooltip" data-placement="bottom"
title="OEE stands for Overall equipment effectiveness.
OEE measurement a commonly used as a key performance
indicator (KPI) in conjunction with lean manufacturing
efforts to provide an indicator of success.">
</div>
```

Kuva 36. OEE mittari ja sen selitys

'gg1' on 'oeo' mittari, jossa on tooltip sekä sen teksti, mikä ilmestyy alapuolelle, että animaatio mikä saa osion pomppimaan 'animated growIn', samalla tavalla meillä on 'gg2', 'gg3', 'gg4' ja niiden tekstit, että eri animaatioit.

```
<script src="css3-animate-it.js"></script>
<script src="raphael-2.1.4.min.js"></script>
<script src="justgage.js"></script>
```

Kuva 37. Scriptviittaukset

Animaatio, justgage pitää ladata niiden sivuilta koneelle ja viitata niihin.

Seuraavaksi mennään javascript-sion puolelle.

```
$(document).ready(function() {
  $('[data-toggle="tooltip"]').tooltip();
});
document.addEventListener("DOMContentLoaded", async function(event) {
  var response = await fetch("https://factory-█.firebaseio.com/values.json");
  var arr = await response.json();
  var valuesparsed = arr['factory1'];
```

Kuva 38. Javascript fetch

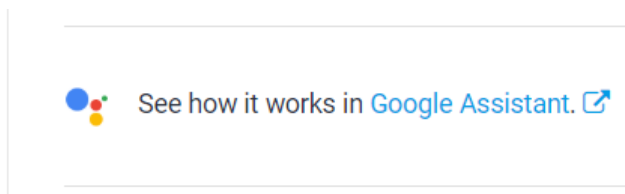
Otetaan tooltip käyttöön bootstrap pluginilla, jotta voimme hakea tiedot, käytämme fetch APIa. URL-sivu, minkä saimme, kun teimme tietokantaa käytämme tässä sitä mutta lopussa lisätään 'values.json' jotta saamme json siitä, factory 1 tiedot haetaan.

```
var gg1 = new JustGage({
  id: 'gg1',
  counter: true,
  hideInnerShadow: true,
  value: valuesparsed['OEE'],
  title: 'OEE ',
  gaugeWidthScale: 1.1,
  min: 0,
  max: 100,
  customSectors : [{"lo":0,"hi":40,"color":"yellow"},
                   {"lo":41,"hi":60,"color":"#f9c802"},
                   {"lo":61,"hi":85,"color":"green"},
                   {"lo":85,"hi":100,"color":"red"}],
  levelColorsGradient: false
});
```

Kuva 39. Javascript OEE

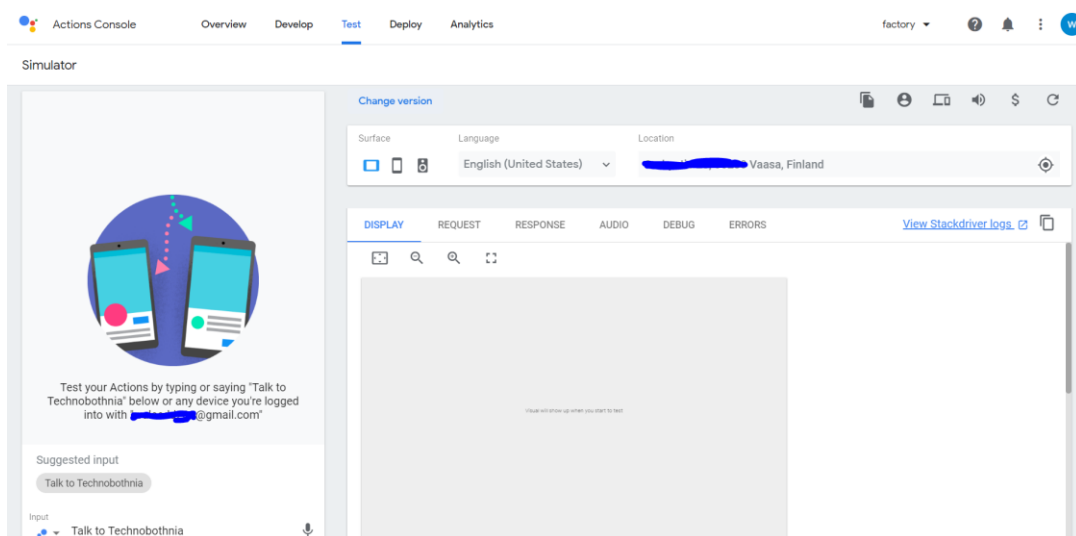
'gg1' eli 'OEE'-mittari, laitetaan maximi ja minimi, mikä meidän tapauksessa on 0 ja 100 sekä leveys asteikko että nimi. Muut mittarit tehdään samalla tavalla.

5 OHJELMAN TESTAUS



Kuva 40. Google Assistant linkki testaukseen

Dialogflown sivun oikealla puolella, kun painaa 'See how it works in Google Assistant' päästään 'Actions console' sivulle.

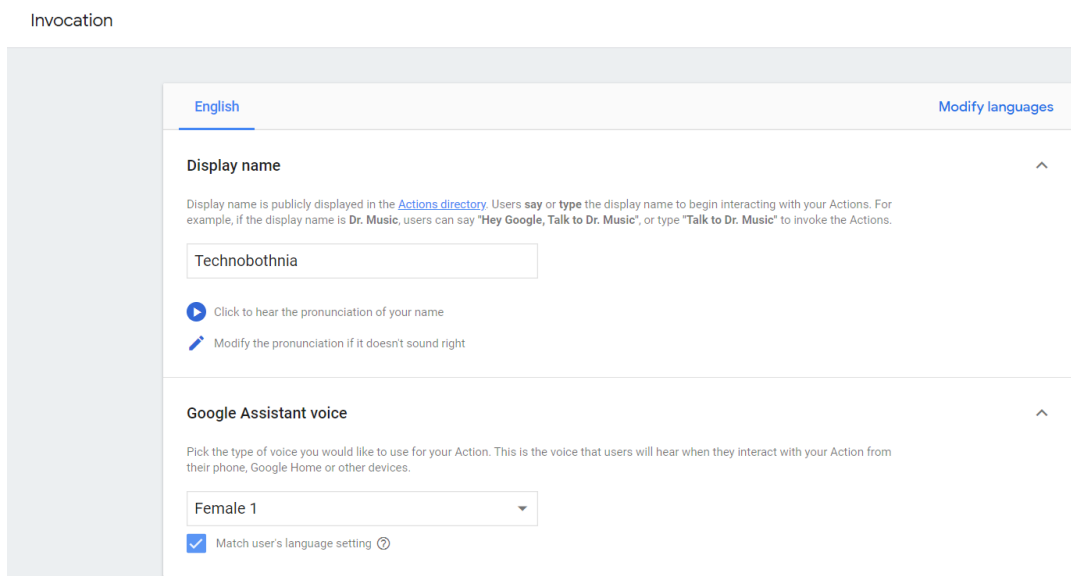


Kuva 41. Google Assistant Simulator

Tässä sivussa voidaan testata Google Assistantia, vasemmalla alalaidassa painamalla 'Talk to my test app' meidän tapauksessa 'Talk to Technobothnia'. 'Surface' kohdalla voi valita sekä laitteen, että miltä näyttää esim puhelimesta, 'Location' paikkakunta kannattaa laittaa tarkasti, jotta saa tarkemmin

tiedot, 'Display' kohdalla näytetään keskustelu, 'Response' näyttää json muodossa vastauksen, sekä 'error' kohdalla näkee ongelmat mainitsin tärkeimmät.

Ylhäällä kuten näkyy ollaan 'Testi'-osiossa painetaan 'Develop'-osiota.



Kuva 42. Invocation osio

Jotta ohjelmaa pääsee käyttämään ja sen voi aktivoida, pitää Google Assistantille laittaa kutsuma nimi, esim meidän tapauksessa 'Talk to Technobothnia', kutsumanimen voi valita minkä tahansa ja sitä pitää kirjoittaa 'Display name' kohdalle, jos ei laita mitään oletuksena on 'Talk to my test app', eli assistentti kutsutaan 'Hey google, Talk to Technobothnia'. 'Google Assistant voice' kohdalla voi valita assistentille äänen, jossa on kahden miehen että kahden naisen äänet. Kaikki laitteet, jossa on google assistant ja yhdistettynä samaan google tiliin kuin projekti voi kutsua avustajaa 'Hey google, Talk to Technobothnia' kutsulla, Google Assistant on tällä hetkellä saatavilla tuetuissa autoissa, kelloissa, televisioissa, google home kaiuttimessa, puhelimissa sekä monia älykodin laitteita pystyy ohjaamaan avustajalla.

6. YHTEENVETO

Projekti oli kiinnostava ja opetti paljon asioita, esimerkiksi Node.js koodaus, Firebase Realtime Database sekä asioita, mitä koulussa opin sain niitä taitoja kehitettyä paremmaksi.

Projekti onnistui hyvin ja tavoitteisiin päästiin, mutta aina voisi keksiä uutta. Ohjelmaa on helppo laajentaa jatkossa lisäämällä käskyjä ja näille haluttuja vastauksia. Itse olen tyytyväinen, että tavoitteisiin päästiin ja lopputulos oli odotetunlainen, toivottavasti työstä hyötyvät muutkin.

LÄHTEET

/1/ What Is Speech Recognition?

Viitattu. <https://www.lifewire.com/what-is-speech-recognition-3426721>

/2/ What Are the Benefits of Speech Recognition Technology?

Viitattu. <https://signalprocessingsociety.org/publications-resources/blog/what-are-benefits-speech-recognition-technology>

/3/ Speech recognition.

Viitattu . <https://www.britannica.com/technology/speech-recognition>

/4/ A BRIEF HISTORY OF VOICE ASSISTANTS

Viitattu . <https://www.theverge.com/ad/17855294/a-brief-history-of-voice-assistants>

/5/ Voice Assistant Timeline: A Short History of the Voice Revolution

Viitattu. <https://voicebot.ai/2017/07/14/timeline-voice-assistants-short-history-voice-revolution/>

/6/ A short history of speech recognition

Viitattu. <https://sonix.ai/history-of-speech-recognition>

/7/ The History of Voice Recognition Technology

Viitattu. <http://blog.condecsoftware.com/a-history-of-voice-recognition-technology>

/8/ A Brief History of ASR: Automatic Speech Recognition

Viitattu. <https://medium.com/descript/a-brief-history-of-asr-automatic-speech-recognition-b8f338d4c0e5>

/9/ SolutionBase: Using speech recognition in Windows XP

Viitattu. <https://www.techrepublic.com/article/solutionbase-using-speech-recognition-in-windows-xp/>

/10/ Google Voice Search gets personalized recognition

Viitattu. <https://www.techspot.com/news/41589-google-voice-search-gets-personalized-recognition.html>

/11/ Apple Siri: the next big revolution in how we interact with gadgets?

Viitattu. https://www.washingtonpost.com/business/technology/apple-Siri-the-next-big-revolution-in-how-we-interact-with-gadgets/2011/10/04/gIQAnq8eOL_story.html?noredirect=on

/12/ Mikä On Siri? Ja Onko Siri Kilpailija?

Viitattu. <https://fin.climathonboston.com/what-is-Siri-962>

/13/ The difference between Google Now and Google Assistant

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

/14/ Leaked Windows Phone 8.x dev build screenshots reveal notification center and more

Viitattu. <https://www.windowscentral.com/leaked-screenshots-windows-phone-developer-build-show-notification-center-and-more>

/15/ The heart of Halo: Actress talks a decade spent playing Cortana

<https://www.nbcnews.com/technology/heart-halo-actress-talks-decade-spent-playing-cortana-1B6732259>

/16/ Amazon Alexa: Cheat sheet

Viitattu. <https://www.techrepublic.com/article/amazon-alexa-the-smart-persons-guide/>

/17/ Google Assistant vs Alexa: What's the best digital assistant in 2019?

Viitattu. <https://www.trustedreviews.com/opinion/google-assistant-vs-alexa-3370865>

/18/ Book a table with the Google Assistant across the country on more devices

Viitattu. <https://www.blog.google/products/assistant/book-table-google-assistant-across-country-more-devices/>

/19/ What is Bixby? Samsung's smart assistant explained

Viitattu. <https://www.pocket-lint.com/phones/news/samsung/140128-what-is-bixby-samsungs-assistant-explained-and-how-to-use-it>

/20/ Älykaiuttimet valloittavat maailmaa, mutta Suomessa ei puheohjauksella saa taksia, leffalippuja tai paljon muutakaan

Viitattu. <https://yle.fi/uutiset/3-10490314>

/21/ The best smart speakers 2019: which one should you buy?

Viitattu. <https://www.techradar.com/news/best-smart-speakers>

/22/ Are Virtual Assistants Worth The Privacy Issues That Come With Them?

Viitattu. <https://afrotech.com/are-virtual-assistants-worth-the-privacy-issues-that-come-with-them>

/23/ Here's Amazon's explanation for the Alexa eavesdropping scandal

Viitattu. <https://www.vox.com/2018/5/24/17391480/amazon-alexa-woman-secret-recording-echo-explanation>

/24/ The rise of voice control in the internet of things

Viitattu. <https://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/The-rise-of-voice-control-in-the-internet-of-things>

/25/ html5 overview

Viitattu. https://www.tutorialspoint.com/html5/html5_overview.htm

/26/ Javascript

Viitattu. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

/27/ Mikä on CSS? Jaa

Viitattu. <https://peda.net/reisjarvi/perusopetus/kisatienkoulu/oppiaineet/tietotekniikka/internet/vat3-1/css/css-luonnos>

/28/ Node.js

Viitattu. <https://www.fi.w3ki.com/nodejs/index.html>

/29/ Node.js Module

Viitattu. <https://www.tutorialsteacher.com/nodejs/nodejs-modules>

/30/ Fulfillment

Viitattu. <https://cloud.google.com/dialogflow/docs/fulfillment-overview>

/31/ What's a Webhook?

Viitattu. <https://sendgrid.com/blog/whats-webhook/>

/32/ Firebase Realtime Database

Viitattu. <https://firebase.google.com/docs/database>

/33/ Voice user interface

Viitattu. <https://learning.oreilly.com/library/view/voice-user-interface/9781788473354/05981d22-b35e-43c8-af15-d016035127e5.xhtml>

/34/ Dialogflow documentation

Viitattu. <https://cloud.google.com/dialogflow/docs/>

/35/ User Inputs with Dialogflow Actions & Parameters

Viitattu. <https://mindsit.net/user-inputs-with-dialogflow-actions-parameters>

/36/ Parameters

Viitattu. <https://cloud.google.com/dialogflow/docs/intents-actions-parameters#params>

/37/ Monitor Database Performance

Viitattu. <https://firebase.google.com/docs/database/usage/monitor-performance>

/38/ Understand Firebase Realtime Database Rules

Viitattu. <https://firebase.google.com/docs/database/security>

/39/ What is JustGage

Viitattu. <https://justgage.com/>

/40/ Phones

Viitattu. <https://i.ytimg.com/vi/BkpAro4zIwU/maxresdefault.jpg>

/41/ Cortana windows phone

Viitattu. [https://cdn.vox-cdn.com/thumbor/-iQ-27pDCtKIzH0EKRxISO_tztk=/0x0:1020x676/1200x800/filters:focal\(429x257:591x419\)/cdn.vox-cdn.com/uploads/chorus_image/image/62960905/cortana-windows-phone-theverge-8_1020.0.jpg](https://cdn.vox-cdn.com/thumbor/-iQ-27pDCtKIzH0EKRxISO_tztk=/0x0:1020x676/1200x800/filters:focal(429x257:591x419)/cdn.vox-cdn.com/uploads/chorus_image/image/62960905/cortana-windows-phone-theverge-8_1020.0.jpg)

/42/ Cortana

Viitattu. [https://cdn.vox-cdn.com/thumbor/sGOHaPE9nawkWQvrSbSKG1j8i4I=/152x0:1757x1070/1400x1400/filters:focal\(152x0:1757x1070\):format\(jpeg\)/cdn.vox-cdn.com/uploads/chorus_image/image/45507198/maxresdefault.0.0.jpg](https://cdn.vox-cdn.com/thumbor/sGOHaPE9nawkWQvrSbSKG1j8i4I=/152x0:1757x1070/1400x1400/filters:focal(152x0:1757x1070):format(jpeg)/cdn.vox-cdn.com/uploads/chorus_image/image/45507198/maxresdefault.0.0.jpg)

/43/ Amazon Echo

Viitattu. https://en.wikipedia.org/wiki/Amazon_Echo#/media/File:Amazon_Echo.jpg

/44/ Amazon Echo vs Google Home

Viitattu. <https://assets.pcmag.com/media/images/573034-amazon-echo-vs-google-home.jpg?thumb=y&width=810&height=456>