



SAVONIA

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

LUONNOLLISEN KIELEN CHATTIBOTTI TERVEYDEN- HUOLLOSSA

TEKIJÄ: Yunus Emre Turunç

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä(t) Yunus Emre Turunç	
Työn nimi Luonnollisen kielen chattibotti terveydenhuollossa	
Päiväys	14.05.2019
Sivumäärä/Liitteet	33
Ohjaaja(t) Keijo Kuosmanen ja Jussi Koistinen	
Toimeksiantaja/Yhteistyökumppani(t) Vireum Oy	
Tiivistelmä	
<p>Tämän opinnäytetyön tarkoituksena oli luoda keskusteleva chattibotti (VireBot) hyödyntäen Googlen Speech-to-Text ja Text-to-Speech toimintoja. Yrityksillä on käytössä kirjoitusta analysoivia chattibotteja, mutta puhetta tuottavia ja tulkkaavia chattibotteja ei ole vielä laajasti käytössä.</p> <p>Opinnäytetyö aloitettiin suunnittelulla. Ohjelmointikieliä tutkittiin etukäteen, jotta löydettäisiin sopivat tälle työlle. Täysin uusia kieliä olivat mm. Angular JS, Ionic Framework ja Java, joita tutkittiin ja opeteltiin etukäteen. Ensimmäisessä vaiheessa VireBot tulee tiedustelemaan terveydentilaan liittyviä kysymyksiä suoraan potilaalta käyttöliittymänä joko Vireumin Android-mobiilisovellus tai Vireum Web-käyttöliittymä. Terveydenhuollon ammattilainen tuottaa VireBottiin kysymyksiä ja lähettää suoraan potilaiden tabletteihin. Kysymyksiä voisi ammattilainen itse määrittää käyttöliittymästä tai käyttää suoraan valmista kyselylomaketta, mikä löytyy yhden napin takaa ja valmista kyselyä voi myös vapaasti haluamansa mukaan muokata (lisätä tai poistaa kysymyksiä). Chattibotin tarkoitus oli toteuttaa erillisenä ohjelmistomodulina, jotta se voidaan integroida asiakkaiden omiin palveluihin muissa projekteissa.</p> <p>Tämän opinnäytetyön tuloksena on toimiva keskusteleva chattibotti. Aihe hallittiin alusta loppuun sekä kaikki asetetut tavoitteet saavutettiin. Lopuksi toimiva chattibotti ominaisuus integroitiin Vireum-home sovellukseen, mikä on potilaiden käytössä tällä hetkellä.</p>	
Avainsanat	
TTS (Text-to-Speech), STT (Speech-to-Text)	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Yunus Emre Turunç			
Title of Thesis Talking Chatbot in Health Care			
Date	14 May 2019	Pages/Appendices	33
Supervisor(s) Mr. Keijo Kuosmanen, Senior Lecturer and Mr. Jussi Koistinen, Senior Lecturer			
Client Organisation /Partners Vireum Oy			
<p>Abstract</p> <p>The purpose of this thesis was to create a conversational chatbot (VireBot) utilizing Google's Speech-to-text and Text-to-Speech functions. Businesses use chatbots that analyze text, but there are not widely used talking chatbots and interpreter bots yet.</p> <p>The thesis started with planning. Programming languages were researched in advance to find suitable ones for this thesis. Completely new languages for the author AngularJS, Ionic Framework and Java were chosen, and they were studied first. In the first version, a VireBot will inquire directly about the patient's health status as a user interface in either the Vireum Android mobile application or the Vireum Web interface. The questions will be generated by a healthcare professional and be sent directly to patients' tablets. The questions could be determined by the professional from the user interface or directly from the ready questionnaire, which will be found behind a single button and it can be freely modified as desired (add or remove questions). The implemented chatbot is intended to be implemented as a standalone software module so that it can be integrated with the customer's own services.</p> <p>As a result of this thesis, a functional conversational chatbot was developed. The subject was managed from the beginning to the end, and the targets set in the schedule were reached. A working chatbot feature was integrated to the Vireum home application which is in patients' use now.</p>			
Keywords			
TTS (Text-to-Speech), STT (Speech-to-Text)			

ESIPUHE

Haluan kiittää ohjaajani Keijo Kuosmasta vinkeistä opinnäytetyö aiheeni liittyen sekä koko Savonia AMK:n organisaatiota monipuolisesta opetuksesta.

Kiitokset myös toimeksiantaja Vireum Oy:n toimitusjohtaja Antti Väänäselle tosi mielenkiintoisesta aiheesta ja tarjoamastaan mahdollisuuksista opinnäytetyön aikana.

Eriyiset kiitokset läheiselleni, jotka tukivat ja motivoivat minua opinnäytetyön ajan.

Kuopiossa 14.05.2019

Yunus Emre Turunç

SISÄLTÖ

1	JOHDANTO	7
1.1	Vireum Oy	7
1.2	Opinnäytetyön tavoitteet.....	7
1.3	Chattibotti	8
1.3.1	Miksi yritykset tarvitsevat chattibottia?	8
2	TYÖSSÄ KÄYTETYT TYÖKALUT JA TEKNIIKAT	9
2.1	Java	9
2.2	Javascript, HTML ja CSS.....	11
2.3	Angular	11
2.4	Ionic framework.....	12
2.4.1	Ionic framework versiot.....	13
2.4.2	Ionic Framework – Asennus	14
2.5	PostgreSQL.....	14
2.5.1	PgAdmin	15
2.6	GitLab.....	15
3	TOIMINNOT	16
3.1	Firebase.....	16
3.1.1	Mikä on device token?	16
3.2	Text-to-Speech	18
3.2.1	Miten Text-to-Speech käytetään?.....	18
3.2.2	Miten Text-to-Speech toimii?.....	18
3.3	Speech-to-text	20
3.3.1	Miten Speech-to-Text toimii?.....	21
4	KÄYTTÖLIITTYMÄ	23
4.1	Sivupalkki.....	23
4.1.1	Asiakasprofiili	24
4.2	Kyselyn luonti	25
4.3	Kysely näkymä Web-käyttöliittymässä.....	26
4.4	Kyselyn vastaanotto.....	26
4.4.1	Kyselyn ilmoittaminen.....	27
4.4.2	Miten chattibotti ominaisuus käytännössä toimii?	28

4.4.3	Yhteenveto kyselystä.....	29
5	YHTEENVETO.....	30
6	LÄHDELUETTELO.....	31

1 JOHDANTO

Vireum on terveydenhuollon etävastaanotto palvelu, joka tarjoaa monimuotoisen yhteydenpidon ja tiedon välittämisen potilaiden ja ammattilaisorganisaation välillä. Terveydenhuollon palveluja tarjoavia yrityksiä on aika monta ja idea syntyi siitä, että Vireum halusi jotenkin erota muista saman alan yrityksistä. Uusi ominaisuus puhuva chattibotti kuulosti tosi mielenkiintoiselta aiheelta ja teki Vireumista entistä helppo käyttäiseksi. Nykypäivänä varsinkin iäkkäät käyttäjät arvostavat, että käyttämänsä sovellus on mahdollisimman yksinkertainen ja helppokäyttöinen.

Terveydenhuollon puolella ammattilaisten on erityisen tärkeä tietää mahdollisimman paljon tietoa potilaiden terveydentilastaan mutta terveydenhuollon ammattilaisilla ei välttämättä ole aina aikaa varata kyselyihin. Mitä jos pystyisi suoraan potilaille lähettämään kyselyn tietokoneesta ja potilaat saisivat vastata silloin, kun heillä sopii parhaiten sekä potilailta saatu vastaukset näkyisivät suoraan ammattilaiselle? Tällöin ei tarvitse erikseen varata aikaa, mikä sopii sekä ammattilaiselle, että potilaalle. Kaiken lisäksi potilaiden ei tarvitse kirjoittaa mitään, kun pystyy helposti puhumalla vastamaan kysymyksiin.

1.1 Vireum Oy

Vireum Oy on 2017 perustettu terveydenhuollon palveluja tarjoava IT-alan startup yritys. Vireum Oy:n ykkösprojektina on Vireum sovellus, joka tarjoaa mm.

- Videopuhelut
- Chat-palvelu
- Ajanvarauskalenteri
- Muistutukset
- Dokumentit
- Hoito-ohjeet
- Potilaskyselyt

Vireum-palvelu sisältää useita ominaisuuksia. Vireum sovelluksesta on Vireum-home versio, joka on toteutettu Ionic frameworkilla, jossa tullaan alun perin käyttämään chattibotti ominaisuutta. Vireum-home sovellus on tarkoitettu ihmisille, jotka tarvitsevat etävastaanotto palvelua. Vireum sovelluksessa voidaan soittaa videopuheluita, lähettää viestejä, ilmoituksia sekä kyselyitä. (Vireum, 2017)

1.2 Opinnäytetyön tavoitteet

Opinnäytetyön tavoitteena on tutkia ja saada suomen kieltä puhuva ja ymmärtävä chattibotti, mikä helpottaa käyttäjien työtä ja säästää aikaa. Opinnäytetyö aiheen suurin haaste oli suomen kieli. Viikon suunnittelun ja tutkimuksen jälkeen päädyin siihen tulokseen, että luonnollisen kielen chattibotti on mahdollista toteuttaa myös suomenkielellä. Google tarjoaa Text-to-Speech ja Speech-to-Text toimintoja 30+ eri kielissä, mikä mahdollisti työn toteuttamisen.

Työn tavoitteita olivat seuraavat:

- Helppokäyttöinen
- Hyvä UI/UX-suunnittelu
- Kielet (Suomi, Englanti)
- Erillinen ohjelmisto moduuli, mikä mahdollistaa ominaisuuden integrointia muihin projekteihin
- Toimiva ominaisuus

Lopputuloksena päästiin asetettuihin tavoitteisiin. Alusta lähtien tiesin, mitä piti tehdä. Ennen, kun aloitin opinnäytetyötä tekemään, olin jo tutkinut tarpeeksi aihetta ja ohjelmointikieliä etukäteen, joita tulen työn aikana käyttämään. Tämä helpotti minun työtä huomattavasti ja kiinnostus aihetta kohtaan kasvoi. Parasta oli se, että itse sain tehdä UI-suunnittelut kokonaan.

1.3 Chattibotti

Chattibotti on avustaja tietokoneohjelma, joka pystyy käymään keskustelua ihmisen kanssa. Botteja suunnitellaan tekemään töitä ihmisen puolesta. Tätä voidaan kutsua myös työn automatisoinniksi. Chattibotit parantavat asiakaskokemusta, sillä ihmiset säästävät rahaa ja aikaa. Tutkimuksien mukaan 2020 mennessä yli 85% asiakkaiden vuorovaikutusta hoidetaan ilman ihmistä. Botteja voidaan hyödyntää moneen tarkoitukseen, kuten tietojen keräämiseen ja kokousten järjestämiseen. Tekoäly on nykyään kova sana ja sitä kehitetään jatkuvasti. Tekoälyn avulla botitkin pystyvät käyttäytymään paljon järkevämmiin ja tarjoamaan parempaa palvelua. (Anadea, 2018)

1.3.1 Miksi yritykset tarvitsevat chattibottia?

Botit pystyvät esimerkiksi käsittelemään useita pyyntöjä valtavalla nopeudella, mitä ihminen ei pystyisi. Tällöin yritys säästää rahaa ja vähemmässä ajassa saa tehtyä enemmän töitä. Kuluttajat hyötyvät myös chattiboteista, kun se helpottaa myös loppu käyttäjien työtä. Esimerkiksi opinnäytetyössä toteuttamani VireBotin avulla käyttäjät pystyvät vastaamaan kyselyihin ilman manuaalista kirjoittamista, mikä voisi olla hankala joillekin käyttäjille. Yrittäjät ovat heränneet tähän muutokselliseen kehitykseen, joten chatti –palvelujen määrä on kasvanut huomattavasti viime aikoina. (Anadea, 2018)

2 TYÖSSÄ KÄYTETYT TYÖKALUT JA TEKNIIKAT

Luonnollisen kielen chattibotin tarkoitus oli integroida olemassa olevaan sovellukseen uutena ominaisuutena, joten en voinut itse päättää kaikki tekniikat osaamiseni perusteella. Jotkut tekniikat olivat entistä tuttuja mutta monet taas täysin uutta. Esimerkiksi AngularJS, Ionic framework ja PostgreSQL olivat tekniikoita, joihin tutustuin ensimmäistä kertaa opinnäytetyössä. Netistä löytyi hyviä esimerkkejä, joiden avulla pääsin suhteellisen nopeasti jyvälle.

Työn toteutuksessa työkaluja sain ihan itse päättää. Tekstieditoriksi käytin backend puolella NetBeans: iä ja muissa tehtävissä alustariippumaton Atomia, joka on avoimeen lähdekoodiin tarkoitettu editori.

2.1 Java

Oliopohjainen ohjelmointikieli Java on erittäin suosittu ohjelmointityökalu ja yksi vanhimmista kielistä, joka on käytetty opinnäytetyössä. Java ohjelmointikieli julkaistiin Yhdysvalloissa 1995, joka on yhteensopiva käytetyimpien käyttöjärjestelmien (Windows, Linux, macOS) kanssa. Javan syntaksi muistuttaa muita oliopohjaisia kieliä, kuten C ja C++. (Koulutus.fi, 2018)

Javasta julkaistiin 11 versiota ensimmäistä julkaisu päivästä nykypäivään. Muutamat ovat:

- JDK 1.0 (Tammikuu 1996, vanhin versio)
- J2SE 1.2 (Joulukuu 1998)
- Java SE 6 (Joulukuu 2006)
- Java SE 11 (Syyskuu 2018, uusin versio) (Wikipedia, Wikipedia, 2019)

```

public Result addQuestion( @JSR303Validation Question q,
                          Validation validation) {

    jsonlogger.logSerialized(q);

    if (validation.hasBeanViolations()) {
        return Results.badRequest();
    }
    try {
        User user = User.find.where().eq("id", q.user_id).findUnique();
        if (user != null) {
            q.setUser(user);
        }else{
            return Results.badRequest();
        }
    }catch (Exception e) {
        logger.error(e.toString());
    }

    try {
        q.save();
        ArrayNode node = mapper.createArrayNode();
        List<PushToken> t = q.getUser().getPushTokens();
        if(!t.isEmpty()){
            logger.debug("sending questions, have " + t.size() + " tokens");
            t.stream().forEach(token -> node.add(token.getDeviceToken()));
            gateway.sendMessage(q, node);
        }else{
            logger.debug("no device tokens for user "+ q.getUser().getId());
        }
    }

    return Results.json().render(q);
}catch (Exception e) {
    logger.error(e.toString());
    return Results.badRequest();
}
}

```

Kuva 1. Esimerkki Java koodi, joka tallentaa kysymykset ja laitetunniste tietokantaan

Opinnäytetyössä Java ohjelmointi kieliä käytin backend puolella mm. kyselyn tallentamista ja poistamista varten tietokanta tauluun. Java on vanha oliopohjainen ohjelmointikieli mutta edelleen yksi käytetyimmistä ohjelmointikielistä nykyajan yrityksissä.

2.2 Javascript, HTML ja CSS

JavaScript on Netspace:n kehittämä dynaaminen web-ohjelmointikieli. JavaScript julkaistiin myös 1995. JavaScript on yksi kolmesta kielestä, joita kaikkien web-kehittäjien on osattava. Web-sivut eivät ole ainoa paikka, jossa JavaScriptiä käytetään. Monet työpöytä- ja palvelinohjelmat käyttävät myös JavaScriptiä, joista Node.js on tunnetuin. Jotkut tietokannat, kuten MongoDB ja CouchDB, käyttävät myös JavaScriptiä ohjelmointikielenä. (W3schools, 2019)

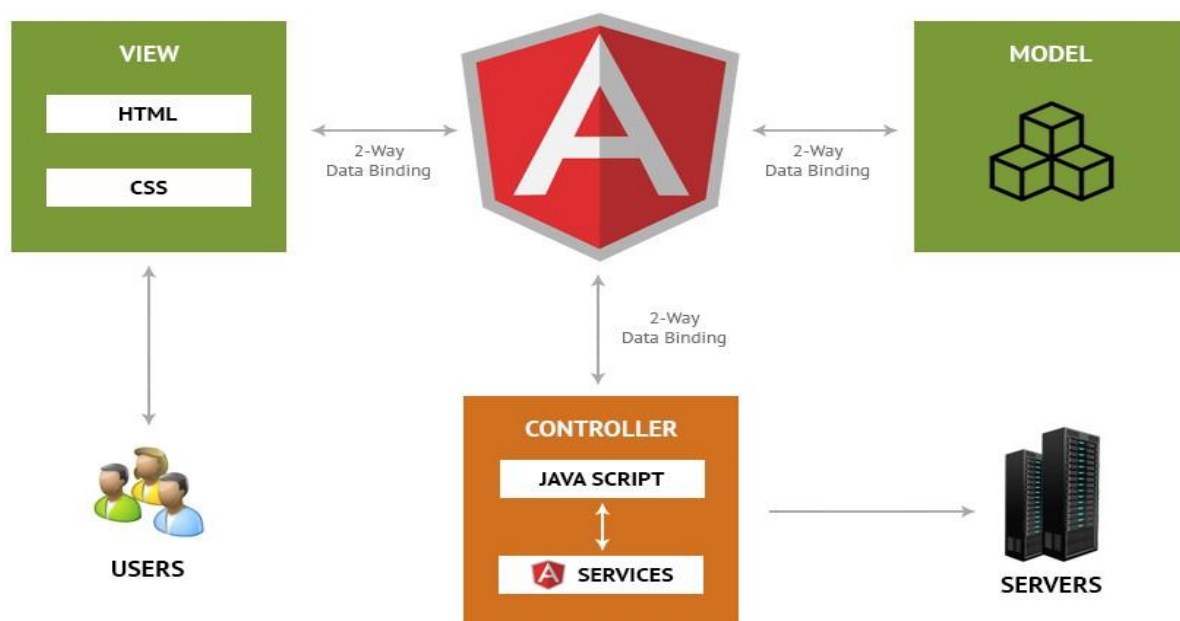
HTML (Hypertext Markup Language) on tietokonekieli, joka on suunniteltu verkkosivustojen luomiseksi, sillä voidaan määrittää verkkosivujen runkoa ja sisältöjä. Se on tosi yksinkertainen ja suhteellisen helppo oppia. (Source, 2012)

CSS (Cascading Style Sheets) on tekniikka, joka määrittelee HTML-kielellä luotu sivun ulkoasun. CSS tarjoaa laajan valikoiman työkaluja ja se on rakenteeltaan tosi yksinkertainen. Kaikki nämä tekniikat kulkevat ns. käsi kädessä. Pelkästään näitä kolmea tekniikoita käyttämälläkin voidaan saada hienon näköisiä netti sivuja ja sovelluksia aikaiseksi. (Tutorialspoint, 2019)

2.3 Angular

Angular on erittäin tehokas JavaScript-ohjelmistokehys, jonka ylläpitää Google Inc. Ensimmäinen versio julkaistiin 2010 mutta kehitys versio vasta 2015. Angular on alusta, jonka avulla on helppo rakentaa sovelluksia verkossa. Angular antaa kehittäjille mahdollisuuden rakentaa eläviä sovelluksia Webissä, mobiililaitteissa sekä työpöydällä. (Wikipedia, 2019)

Angularista on 2 eri versiota, AngularJS ja Angular 2. Vaikka AngularJS:n kehittäjät kirjoittivat Angular 2:en, silti on tosi paljon eroa kahden version välillä. Voidaan olettaa, että uudemmassa versiossa on joitakin etuja verrattuna vanhempaan versioon. Kyllä se vain näin on. Ensinnäkin AngularJS perustuu JavaScriptiin ja Angular 2 taas TypeScriptiin. Suurin etu uudessa versiossa on modulaarisuus. Paljon ydintoimintoja siirrettiin eri moduuleihin ja se aiheutti ytimen, dynaamisen latauksen, asynkronisen mallin kokoamisen ja lisätuen reaktiiviseen ohjelmointiin. AngularJS:llä on monia direktiivejä, ja jokainen kehittäjä voi myös määrittää uuden direktiivin. Angular 2:ssa on myös direktiivejä, mutta niitä käytetään hieman eri tavalla. (House, 2017)



Kuva 2. Angular Model View Controller -arkkitehtuuri (TECHNOLOGIES, 2019)

Angular toteuttaa MVC (Model View Controller) -arkkitehtuurin web-sovelluksen kehittämiseksi. MVC-arkkitehtuurin käyttö auttaa tietojen hakemisessa ja lisäämisessä käyttöliittymästä riippumatta. Controller reagoi käyttäjän syöttöön ja toimii siltana näkymän ja mallin välisen yhteyden muodostamisessa. (TECHNOLOGIES, 2019)

2.4 Ionic framework

Ionic Framework on täysin maksuton avoimen lähdekoodin kehys, jonka on kehittänyt Max Lynch, Ben Sperry ja Adam Bradley vuonna 2013. Ionic frameworkkiä käytetään hybridi sovelluksien rakentamiseen. Ionic tarjoaa työkaluja ja palveluja hybridi-mobiilisovellusten kehittämiseen käyttäen mobiili ystävällisiä Web-tekniikoita, kuten HTML5 ja CSS. Sovelluksia voidaan rakentaa näiden verkko-tekniikoiden avulla ja jakaa sen jälkeen paikallisten sovellusmyymälöiden kautta, jotka ovat asennettavissa laitteille hyödyntäen Apache Cordovaa. (Ganatra, 2015)

Ionic frameworkin hyvä puoli on se, että sillä pystyy kehittämään hybridi sovelluksia. Web-sovelluskehittäjien ei erikseen tarvitse opetella mobiili ohjelmoinnin kieliä, kuten Kotlin (Android) ja Swift (iOS), kun Ionic mahdollistaa kehittämään Web-tekniikoilla hybridi sovelluksia. Hybridi sovellus on siis se, että yksi koodi pohja toimii iOS- ja Android käyttöjärjestelmissä sekä selaimessa. Yksinkertaistaa koodin kirjoittamista ja säästää aikaa. Ionic on suunniteltu toimimaan sujuvasti kaikissa nykypäivän mobiililaitteissa ja selaimissa. Ionic on yksi käytetyimmistä natiivi kehysistä tällä hetkellä, joka on kehitetty Apache Cordovan ja Angularin päälle. Ionic frameworkillä on myös tosi kaunis oletus käyttöliittymä, jonka voidaan helposti muokata yksityiskohtaisesti. (Ami, 2018)

2.4.1 Ionic framework versiot

Ottamalla mobiilisovellus kehittäjien palautteita, Ionic-tiimi päätti rakentaa oman kehiksen nykykaisilla web-standardeilla, joka keskittyy suorituskykyyn.

- Versio 1
 - Ionic frameworkin alfa versio julkaistiin marraskuussa 2013
 - Beta-versio julkaistiin maaliskuussa 2014
 - Versio 1.0 tuli vasta toukokuussa 2015

Ionic frameworkin ensimmäisessä versiossa tuetut järjestelmät olivat Android 4.1 ja uudemmat, iOS 7 ja uudemmat sekä Black Berry 10 sovellukset. Jos jostain syystä halutaan edelleen tehdä projekteja Ionic 1:llä (ei suositella), projektin pystyy luomaan `"-type ionic1"` lipun avulla. (Ganatra, 2015)

- Versio 2

Ionic:sta uusi versio eli version 2 julkaistiin 2016, jonka mukaan tuli tosi paljon uusia ominaisuuksia verrattuna ensimmäiseen versioon. Ionic 2:ssa vaihtui kieli (Angular 2) ja arkkitehtuuri. Tässä versiossa on esimerkiksi paljon uusia komponentteja, jopa vanhemmat komponentit ovat optimoitu, joita on tosi helppo käyttää sovelluksissa. Muita uutuuksia ovat mm. Uusi navigointi järjestelmä, tehokas lomake kontrolli komponentti ja uusi virheenkäsittely menetelmä. (Wickramasinghe, 2017)

- Versio 3

Ionic tiimiläiset työskentelevät kovasti ja Ionic framework kasvaa tosi kovaa tahtia. Jo vuosi sen jälkeen, kun Ionic 2 tuli markkinoille, julkaistiin toukokuussa 2017 Ionic 3. Tässä versiossa Ionic framework päivitettiin yhteensopivaksi Angular 4.0 ja TypeScript 2.2: n kanssa, joka toi mukanaan joi-takin uusia ominaisuuksia. TypeScript mahdollistaa nopeamman käynnistyksen (build) sekä tyy-pin tarkistamisen sovelluksessa. Tämän avulla pystytään rakentamaan kevyempiä ja nopeampia sovel-luksia. (Carney, 2017)

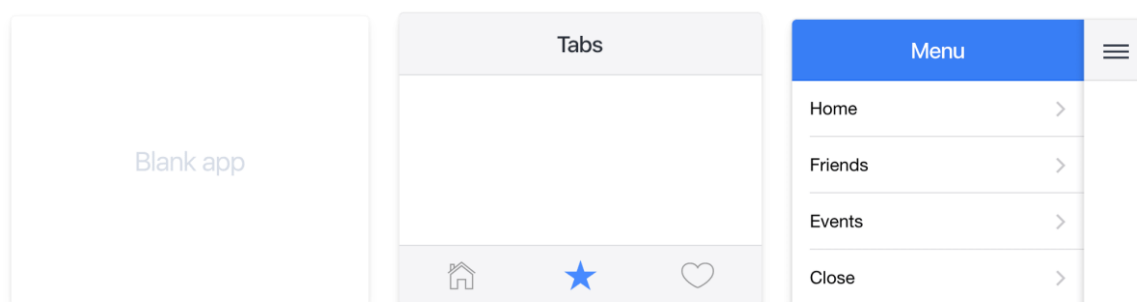
- Versio 4

Opinnäytetyön kirjoittamishetkellä tuorein versio Ionic:sta on 4, joka julkaistiin 2018. Beta-versio antoi kehittäjille ensimmäisen vaikutelman suorituskyvyn parantamisesta. Ionic 4 on edelleen Angu-larin johtava mobiiliratkaisu, joka on ottanut uusia Angular työkaluja ja ominaisuuksia käyttöön. Io-nic 4 tuo merkittäviä suorituskykyä, uusia dokumentaatiota, tehokkaita teemoja ja paljon muuta pie-niä parannuksia. Vaikka onkin paljon eroa ja parannuksia vanhoihin versioihin verrattuna, Ionic 4:een siirtyminen on yksinkertaista ja helppoa. (Bradley, 2018)

2.4.2 Ionic Framework – Asennus

Ionic suosittelee projekteja luomaan Ionic CLI:ä käyttäen, joka tarjoaa laajan valikoiman työkaluja ja apuvälineitä. Ennen, kun asennetaan Ionic CLI:ä pitäisi Node.js ja npm olla asennettuina koneella. Node.js voidaan tarkistaa komennolla "node -v", jos tuloksena saadaan esim. "v11.0.0" tarkoittaa, että node.js on asennettuina koneella ja versio numero on 11.0.0. Npm tarkistetaan komennolla "npm -v" ja tuloksena pitäisi saada esim. "6.4.1". (Ionic, 2019)

- 1) Node.js & npm asennus
- 2) npm install -g ionic
- 3) ionic start myFirstApp blank / tabs / sidemenu
- 4) cd myFirstApp
- 5) ionic serve



Kuva 3. Esimerkki kuvat Ionic projekti vaihtoehtoista (Blank, Tabs, Sidemenu) (Ionic, 2019)

"npm install -g ionic" komennolla asennetaan ionic: iä globaalisesti (-g lippu) koneelle. "ionic start myFirstApp blank" komento luo tyhjän projektin, jonka nimi on "myFirstApp". Vaihtoehtoja on tyhjän lisäksi kaksi. Voidaan luoda "tabs" projektin, joka tekee valmiiksi (ainakin) kolme eri välilehteä projektiin. Sitten on vielä "sidemenu" vaihtoehto, se luo siis projektin, jossa on valmiiksi sivupalkkia, jota voi helposti muokata omanlaiseksi. "cd myFirstApp" komennolla siirrytään (cd = change directory) juuri luotuun projekti kansioon ja "ionic serve" ajaa projektin paikallisesti portissa 8080 (localhost:8080).

2.5 PostgreSQL

PostgreSQL on tehokas avoimen lähdekoodin tietokantamoottori, joka käyttää ja laajentaa SQL-kieltä. Turvallinen ja helppokäyttöinen tietokantajärjestelmä toimii kaikilla tunnetuilla käyttöjärjestelmillä, kuten Windows, Linux ja macOS. PostgreSQL:ssä on monia ominaisuuksia mm. Tietotyypit, luotettavuus, turvallisuus, laajennettavuus ja tekstihaku. Mahdollistaa hallitsemaan tietoja ja tiedostoja koosta riippumatta. PostgreSQL antaa kehittäjien määrittää omia tietotyyppisiä ja kirjoittaa koodia eri ohjelmointi kielistä, joka ei ole mahdollista monessa muissa tietokantajärjestelmissä. (PostgreSQL, 2019)

2.5.1 PgAdmin

PostgreSQL:än suosituin työkalu on pgAdmin, joka on julkaistu PostgreSQL -lisenssillä. Graafisella käyttöliittymällä varustettu pgAdmin -ohjelmistoa kehitetään jatkuvasti. Uusin versio on opinnäytetyön kirjoittamishetkellä v.4.3, joka on julkaistu helmikuussa 2019. PgAdmin kehitystiimin mukaan uusimmassa versiossa on tehty kovaa työtä, sillä yli 100 bugi korjausta tehty ja lisätty uusia ominaisuuksia verrattuna versioon 3. (pgAdmin, 2019)

2.6 GitLab

”GitLab on verkkopalvelu, joka tarjoaa Git-versiohallinnan, wikin, bugienseurannan, kehitystoiveet ja tehtävienhallintatoimintoja. GitLabin lähdekoodi on kirjoitettu Ruby-ohjelmointikielellä, ja sen kehityksestä vastaa GitLab Inc. Myöhemmin joitain osia ohjelmistosta on uudelleenkirjoitettu Go-kielellä. GitLabia käyttävät useat organisaatiot, kuten IBM, Sony, Nasa, Alibaba, SanDisk ja Intel.” (Wikipedia, 2018)

GitLabin ominaisuuksia ovat mm.

- Isojen ja pienien projektien tallentaminen
- Avoin lähdekoodi kirjasto
- Hosting palvelut
- Vian seuranta
- Tiedoston muokkaaminen mahdollisuus suoraan Web-käyttöliittymästä

GitLab eroaa kilpailijoistaan GitHubista ja Bitbucketista, jotka velottavat tietovarastoista tai lisäkäyttäjiltä. GitLab on tosi kätevä erityisesti, jos tehdään ryhmätöitä. Projektin koodia ylläpidetään GitLabissa ja jokainen ryhmän jäsen, joilla on oikeus (koodin jakamiseen), voi tallentaa omat muutokset GitLab projektiin. Muut ryhmän jäsenet pystyvät ottamaan omalla koneella saman koodi muutoksen projekti kansiossa komennolla ”git pull”. GitLabissa säilyy sekä vanhat, että uudet versiot koodista, joten tarvittaessa pystytään helposti palaamaan edelliseen versioon. (Yeremeeva, 2017)

Alla on muutama hyödyllinen git komento:

- `git clone`
- `git pull`
- `git status`
- `git checkout`
- `git branch`
- `git add .`
- `git commit -m`
- `git push`

3 TOIMINNOT

3.1 Firebase

Firebase on Inc on James Tamplin ja Andrew Lee:n perustama vuonna 2011 yritys. Firebase on mobiili ja web-sovellusten kehitysympäristö. Firebasen ensimmäinen tuote oli Firebase Realtime -tietokanta. API, joka synkronoi sovellustiedot iOS-, Android- ja Web-laitteista ja tallentaa niitä Firebase-pilviin. Tämä tuote mahdollistaa ja auttaa ohjelmistokehittäjiä rakentamaan reaaliaikaisia yhteistyösovelluksia. (Wikipedia, 2019)

2014 lokakuussa Google Inc osti Firebasen ja heti seuraavana vuonna osti hosting palveluja tarjoava Divshot nimisen yrityksen ja yhdisti sen Firebase-tiimiin. Firebase-hosting on Firebasen staattinen web-hosting palvelu, jota osittain tukevat Divshotin teknologiat. Nykypäivänä Divshot-arkkitehti ja web-hosting tuotteet ovat lopettaneet toimintansa. (Wikipedia, 2019)

Firebasella on yhteensä noin 15 toimintoa, käytetyimmät ovat mm.

- Analytiikka
- Hosting-palvelut
- Reaaliaikainen tietokanta ja tietovarasto
- **Firestore pilvi-viestipalvelu (FCM)**
- Authentication as a Service (Konkka, 2017)

Minua kiinnosti opinnäytetyön aikana FCM eli Firebase Cloud Messaging toiminto. Firebase:ä hyödynsin ammattilaiskäyttäjän kyselyn lähettämistä varten, suoraan käyttöliittymästä potilaan tablettiin laitetunnisteen avulla.

3.1.1 Mikä on device token?

Device token on laitetunniste, jonka avulla viestit ja ilmoitukset menevät laitteille, jolle se on tarkoitettu. Laitetunniste on merkkijono, joissa on 64 heksadesimaalista symbolia. Esim. laitetunniste "a2df25c1608agusEmrEda160497tMmZdfde210717fF7cc993eb6y6i3ewrt08la". (Polkhovskiy, 2019)


```

firebaseCloudMessaging() {
  console.log("Firebase instance:", { firebase: this.firebase });
  this.firebase
    .getToken()
    .then(token => {
      console.log(`The token is ${token}`);
      this.updateTokenToBackend(token);
    })
    .catch(error => console.error("Error getting token", error));
  this.firebase.onTokenRefresh().subscribe((token: string) => {
    console.log(`Got a new token ${token}`);
    this.updateTokenToBackend(token);
  });
  this.firebase.onNotificationOpen().subscribe(
    data => {
      if (data.PushMessageId) {
        console.log("notification FCM", data);
        this.notification.next(data);
      } else {
        console.log("Questionnaire FCM", data);
        this.questionnaire.next(data);
      }
    },
    function(error) {
      console.error(error);
    }
  );
}

```

Kuva 4. Koodi esimerkki firebase funktiosta

Ylhäällä on näyttökuvaa funktiosta, joka tulostaa laitetunniste konsoliin. Jos jostain syystä laitetunniste muuttuu, sille generoidaan uusi ja tulostetaan se. Tässä funktiossa kutsutaan myös "updateTokenToBackend" funktiota, joka näkyy alhaalla.

```

updateTokenToBackend(token) {
  this.http
    .post(this.resolver.resolve("ADD_DEVICE_TOKEN"), { token: token })
    .subscribe(
      result => {
        console.log("result of sending token to be:", result);
      },
      error => {
        console.log("error", error);
      }
    );
}

```

Kuva 5. Koodi pätkä laitetunnisteen tietokantaan lähettämisestä

FirebaseCloudMessaging -funktiossa saatu laitetunniste lähetetään backendiin ja tallennetaan tietokantaan. Jos tällä käyttäjällä on aiemmin jo laitetunniste tietokannassa, mikä ei ole sama kuin nyt saatu, vanha korvataan uudella ja jatkossa ilmoitukset lähetetään uuden tunnisten perusteella.

3.2 Text-to-Speech

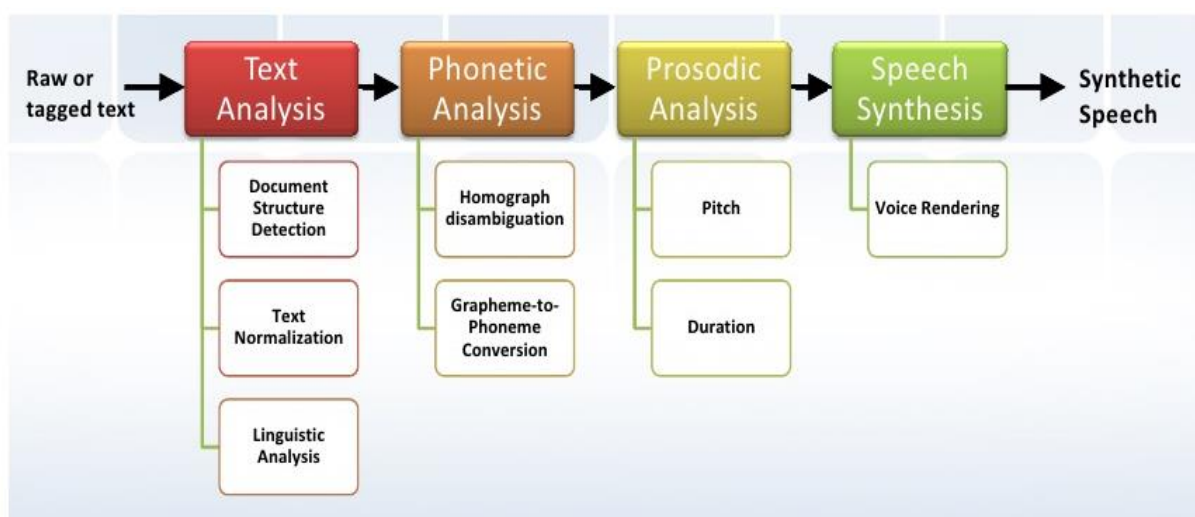
Text-to-Speech (TTS) on aputeknologia, joka lukee ääneen digitaalisen tekstin tietokoneissa, älypuhelimissa ja tableteissa. Nykypäivänä lähes jokaiselle digitaaliselle laitteelle on saatavilla TTS-työkaluja. Napsauttamalla painiketta tai sormea, TTS voi muuntaa tekstin ääniksi noin 45+ eri kielellä. (Understood, 2018)

3.2.1 Miten Text-to-Speech käytetään?

Kaikentyyppisiä tekstitiedostoja voidaan lukea ääneen, mukaan lukien Word, Pages-asiakirjat ja online-sivut. TTS:n ääni generoidaan tietokoneella, ja lukunopeutta voidaan yleensä nostaa tai hidastaa. Äänenlaatu vaihtelee, mutta monet äänet ovat inhimillisiä. Maailman puhutuimmat kielet kuten Englanti, Kiina, Venäjä, Saksa ja Ranska eroavat hieman muista. Näissä on enemmän kuin kaksi ääni vaihtoehtoja. Esim. Englannin kielessä on tarjolla noin 5+ eri ääni vaihtoehtoja (nainen, mies, eri aksentti). Tekstin lukunopeutta ja haluamaamme kielen voidaan helposti asettaa koodissa kielen koodien avulla (en-US, fi-FI, tr-TR). (Understood, 2018)

On jopa tietokoneella tuotettuja ääniä, jotka kuulostavat lapsilta. Tällaisia ääniä käytetään yleensä peleissä, jotka on tarkoitettu lapsille. Monet TTS-työkalut korostavat sanoja, kun niitä luetaan ääneen. Näin lapset voivat nähdä tekstiä ja kuulla sen samanaikaisesti. Joissakin TTS-työkaluissa on myös optinen merkintunnistus (OCR). OCR mahdollistaa TTS-työkalujen lukemisen ääneen kuvista. Lapsi voi esimerkiksi ottaa valokuvan katukyltistä ja saada merkin sanat muuttumaan ääniksi. (Understood, 2018)

3.2.2 Miten Text-to-Speech toimii?



Kuva 6. TTS:n rakenne

1) Text Analysis

Ääneksi muunnettava teksti käydään läpi ja analysoidaan ensin.

- DST on rakenne, joka perustuu tekstin eri visuaalisten ominaisuuksien analyysiin

- Tekstin normalisointi (Text normalization) on prosessi, jossa tunnistetaan idiomaattiset aiheet, lyhenteet ja numerot. Tekstin analyysissä muunnetaan ne täydelliseksi tekstiksi tarpeen mukaan (Microsoft, 2018)
- Kielellistä analyysiä (Linguistic analysis) käytetään ääntämisen päättämiseksi. Käytetään myös äänen akustiset esitykset aaltomuotojen tuottamiseksi (Colotte, 2017)

2) Foneettinen analyysi (Phonetic analysis)

Foneettinen analyysi perustuu perinteiseen puheäänien luokitteluun käyttäen kansainvälistä foneettista aakkosia (IPA). IPA on järjestelmä, jota käytetään yleisesti tutkimukseen ja kieltenopetukseen. (Mopapa, 2017)

- Homograph disambiguation
- Grapheme-to-phoneme conversion

3) Prosodia analyysi (Prosodic Analysis)

Prosodia on tärkeä puhe, joka auttaa ylläpitämään ilmentymiskykyä ja ymmärrettävyys puheesynteesijärjestelmissä. Prosodiset komponentit ovat piki, kesto, aksentti ja muotoilu.

- Pitch (Piki)
- Duration (Kesto)

4) Speech synthesis

Text-to-Speech (TTS) -järjestelmä muuntaa normaalin kielen tekstin puheeksi.

- Voice rendering eli äänen renderöinti.

```
readQuestion() {
    this.tts.speak({
      locale: 'fi-FI',
      text: this.ques.questions[this.j].message,
      rate: 1
    })
    .then(() => console.log('Success'))
    .catch((reason: any) => console.log(reason));
}
```

Kuva 7. Koodi esimerkki kysymyksen äänen lukemisesta

Ylhäällä on näyttökuvaa readQuestion-funktiosta, jossa näkyy, miten kysymyksen äänen lukeminen tapahtuu. Speak-funktiossa asetetaan kieli, luettava teksti sekä lukunopeus. Asetetaan kieleksi suomi mutta Googlen Text-to-Speech toiminto ymmärtää suomen kielen lisäksi myös englantia ja ruotsia automaattisesti. Seuravaksi luettava teksti, minun tapauksessa kysymys, eli ques taulukosta questions objektin j kohdan viesti. "j" muuttuja asetetaan ensin nolllaksi ja kasvatetaan sen mukaan,

kun siirrytään kysymyksissä eteenpäin. "rate" eli lukunopeus voidaan määrittää 0-1 välistä. 1 on oletus arvo ja ihan normaali ihmisen puhenopeus, joten en muuttanut sitä. Jos tulee semmoisia tilanteita, että halutaankin vähentää lukunopeutta, se onnistuu helposti. Esim. "rate: 0.88"

3.3 Speech-to-text

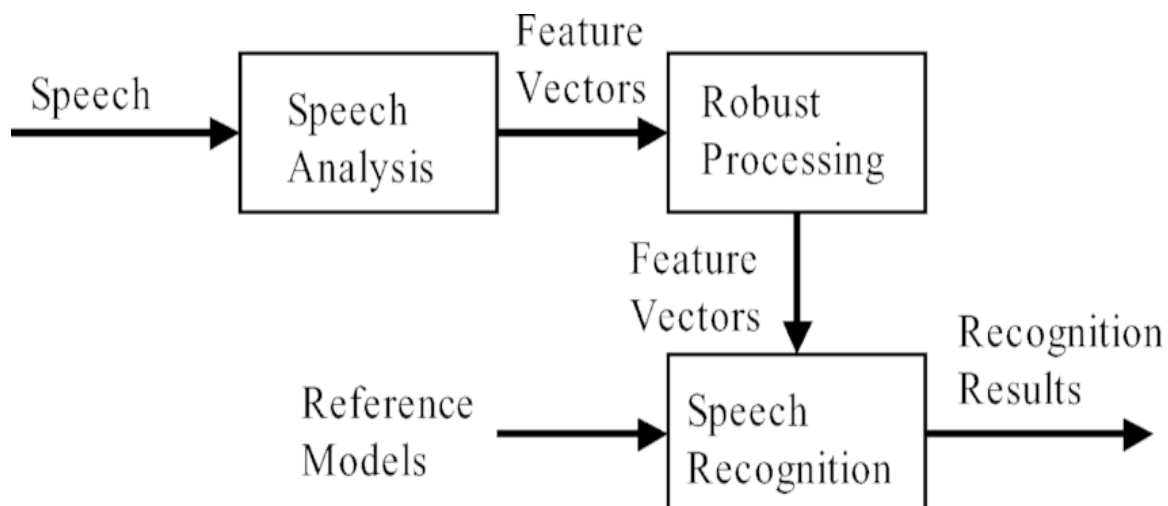
Speech-to-text (Speech recognition) ohjelmisto on puheen tunnistustekniikka, joka ottaa tehokkaasti äänisisällön ja muuntaa sen tekstinkäsittelyohjelmassa tai muussa näyttöpaikassa kirjoitetuiksi teksteiksi. Tämän tyyppinen puheentunnistusohjelmisto on erittäin arvokas kaikille, jotka tarvitsevat paljon kirjoitettua sisältöä ilman paljon manuaalista kirjoittamista. Se on hyödyllistä myös vammaisille ja iäkkäille, joille näppäimistön käyttö on vaikeata. (Techopedia, 2018)

Speech-to-Text ohjelmisto myydään eri alustoilla erillisenä sovelluksena mutta uusimmissa käyttöjärjestelmä versioissa joillekin laitteille se on valmiiksi sisään rakennettu ominaisuus. Henkilökohtaiset avustajat (personal assistants) ovat olleet jonkin aikaa markkinoilla. Tunnetuimmat ovat Applen Siri, Google Assistant, Amazon Alexa, Samsungin Bixby ja Microsoftin Cortana. Nämä avustajat osaavat kuunnella ja vastata Speech-to-Text ja Text-to-Speech -ohjelmistojen avulla. Jotkut käyttävät myös tekoälyä hyväksi, jolloin osaavat antaa paljon järkevämpiä neuvoja. (Hindy, 2019)

Teknisen toiminnon kannalta monet puhekieliset ohjelmat rikkovat puhesanan äänen lyhyiksi sanoiksi ja yhdistävät nämä sanat yksinkertaisiin foneemeihin tai ääntämisyksiköihin. Sitten monimutkaiset algoritmit lajittelevat tulokset ja lopuksi näyttävät kokonaisina lauseina. Speech-to-Text -ohjelmisto on parantunut melko tarkasti ja kehittynyt yleisesti toimivaksi, jotta sillä olisi suurempi rooli nykyaikaisessa viestinnässä digitaalisilla alustoilla. Googlen puheen tunnistus toiminnossa on aika laaja kieli valikoima. Googlen Speech recognizer API on ilmainen ja osaa tunnistaa 100+ eri kieltä.

Opinnäytetyössä Speech-to-Text toimintoa käytetty kuuntelemaan potilaiden vastauksia. Vireum on tarkoitettu enemmän iäkkäille ihmisille, jotka eivät välttämättä osaa manuaalisesti kirjoittamalla vastaamaan kysymyksiin. Äänen tunnistus helpottaa huomattavasti Vireumin käyttöä. Sitten, kun kysymys on luettu ääneen, pystyy puhumalla vastaamaan jokaiseen kysymykseen näppärästi, jonka jälkeen ymmärretty lause näytetään näytössä.

3.3.1 Miten Speech-to-Text toimii?



Kuva 8. Kaavio, jossa näkyy, miten puheen tunnistus toimii. (Miyanaga, 2015)

Ylhäällä on kaavio, jossa näkyy vaiheet, miten puheen tekstiksi muuntaminen toimii. Jos mennään syvemmälle puheen muuntaminen tekstiksi tietokoneessa (mikä tahansa laitteessa) on käytävä läpi useita monimutkaisia vaiheita. Kun puhutaan, syntyy ilmaa värähtelyjä. Analoginen-digitaalimuunnin (ADC) muuntaa tämän analogisen aallon digitaalisiksi tiedoiksi, joita tietokone pystyy ymmärtämään. Järjestelmä suodattaa digitoidun äänen poistamaan kohinaa ja joskus erottamaan sen eri taajuusalueille (taajuus on ääniaaltojen aallonpituus, jonka ihmiset kuuntelevat eroavaisuuksina). Se myös normalisoi äänen tai säätää sen vakiotasolle. (Grabianowski, 2006)

Seuravaksi signaali jaetaan pieniin segmentteihin, jonka jälkeen ohjelma sovittaa nämä segmentit tunnetuille foneemeille sopivalla kielellä. Viimeinen vaihe tuntuu yksinkertaiselta mutta ei se oikeasti ole. Ohjelma tutkii foneemeja niiden ympärillä olevien muiden foneemien yhteydessä. Se kulkee kontekstuaalisen foneemipiirin läpi monimutkaisen tilastollisen mallin avulla ja vertaa niitä tunnetuilla sanoilla ja lauseilla. Tämän jälkeen ohjelma määrittää, mitä käyttäjä todennäköisesti sanoi ja tulostaa tuloksen tekstinä. (Grabianowski, 2006)

Sekä Suomessa, että monessa muussa maissa puhutaan eri murteilla, mikä vaikeuttaa puheentunnistusjärjestelmän tehtävää. Minun testi tulokset näyttivät, että kyllä Googlen Speech-to-Text toiminto ymmärtää myös jonkun verran murteita mutta jos halutaan saada melko täydellistä tulosta, käyttäjien on puhuttava "puhdasta Suomen kieltä".

```

listenForSpeech():void {

  this.androidOptions = {
    language: 'fi-FI',
    showPopup: false,
    prompt: 'Nyt voit puhua'
  }

  this.isSpeechSupported();
  this.getPermission();

  this.speech.startListening(this.androidOptions).subscribe(data => {this.reply = data[0];
  this.stopListening();
  this.reply = this.reply.substring(0,1).toUpperCase() + this.reply.substring(1); // Capitalize first letter of a reply
  console.log("Reply: " + this.reply);
  this.ques.questions[this.j].answer = this.reply; // adding given answer to question j
  this.ques.questions[this.j].date = moment(); // adding replying moment(date)
  console.log("updated ques: ", this.ques);
}, error => this.stopListening() // if error is true (No Match), we call stopListening function.
);
  this.isRecording = true;
  this.ref.detectChanges();
  console.log("recording: ", this.isRecording);
}

stopListening() {
  window.plugins.speechRecognition.stopListening(success => {
    this.isRecording = false;
    this.ref.detectChanges();
    console.log("recording: ", this.isRecording);
  }, error => console.log("error"));
}

```

Kuva 9. Angular koodi funktiosta, joka kuuntelee vastauksia

ListenForSpeech –funktiossa kuunnellaan potilaan vastauksia. Ihan ensimmäisenä määritetään asetukset funktiolle. Kieleksi annettu suomi, eli sovellus osaa kuunnella ainoastaan suomen kieltä. Seuravaksi showPopup alustetaan falseksi. Tämä estää androidin oma kuuntelija ikkunan ilmestymistä näyttöön. Ikkunan ilmestymisen jälkeen kuuluu äänimerkki, joka tarkoittaa, että laite on nyt kuuntelemassa meitä. Prompt on lause mikä näkyy androidin ikkunassa.

Joka kerta kun, käyttäjä painaa mikrofoni ikonia tabletti näkymässä, kutsutaan startListening – funktiota. Funktio ajetaan äsken määritettyjen asetusten avulla ja osumat tallennetaan data taulukkoon. Googlen Speech-to-Text toiminto lopettaa kuuntelemisen automaattisesti, kun käyttäjä on ollut muutaman sekunnin hiljaa. Tallennetaan "data" taulukkoon maksimissaan 5 eri osua. Data taulukon nolanneessa indeksissä oleva vastaus on todennäköisempi oikein, joten määritetään vastaukseksi data[0] eli ensimmäinen vastaus. Selvällä suomen kielellä annettu vastaus ymmärretään melko täydellisesti. STT ei välttämättä aina osannut kirjoittaa lauseen ensimmäisen kirjaimen isolla alkukirjaimella, joten se piti manuaalisesti korjata koodissa.

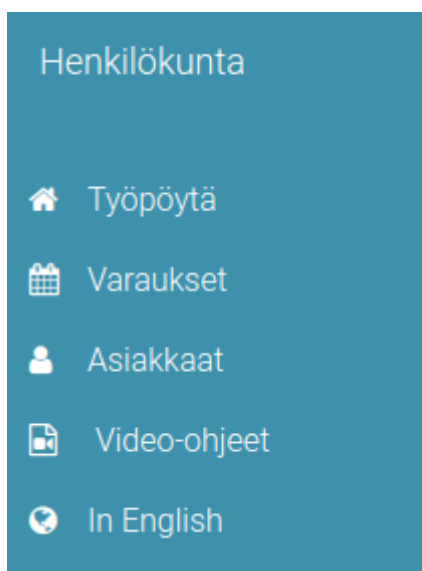
Halusimme näyttää ammattilaiselle potilaan vastauksien lisäksi myös se hetki, milloin on mihin kysymykseen vastattu. Ensin tallennettiin annettu vastaus taulukkoon kysymys kohdassa j, sen perusteella missä kysymyksessä ollaan menossa. Samalla lisättiin vastaus hetki taulukkoon moment – funktion avulla. Moment.js on kevyt JavaScript kirjasto, jonka avulla pystyy käsittelemään aikaa koo-

dissa. Moment.js on tosi helppokäyttöinen ja kätevä kirjasto jos pitää tallentaa tai näyttää aika tietyssä formaatissa. Tämän jälkeen asetetaan isRecording boolean muuttujan arvoksi true. Virhe tilanteissa kutsutaan stopListening -funktiota, jossa lopetetaan kuuntelemista ja isRecording muuttujan arvo muutetaan falseksi. Tämän muuttujan avulla (isRecording) pystytään "Vastaa" napin väriä muuttamaan. Jos laite on kuuntelemassa meitä, nappi näkyy punaisena ja kun se lopettaa kuuntelemisen näkyy sinisenä. Yleisin virhe on verkkoyhteys ongelma. Tällaisessa tilanteessa tulostetaan virheilmoitus konsoliin, että mistä se saattaisi johtua.

4 KÄYTTÖLIITTYMÄ

Käyttöliittymän tarkoitus on antaa ammattilaisille mahdollisuuden lähettämään kyselyitä ainoastaan omiin asiakkaisiin. Sekä ammattilaisten, että potilaiden on kirjautuva Vireum -järjestelmään, jotta palvelua pystyisivät käyttämään.

4.1 Sivupalkki



Kuva 10. Sivupalkki

Käyttöliittymään kirjautumisen jälkeen ammattilaiselle näkyy kuva 10:ssä oleva sivupalkki. Meitä kiinnostaa "Asiakkaat" kohta, jonka alta löytyy jokaisen ammattilaisen omat asiakkaat listattuna. Organisaatioilla voi olla monta ammattilais- ja asiakas käyttäjiä. Jokainen ammattilainen näkee ainoastaan omat asiakkaansa, ei mitään ylimääräistä tai jonkun, joka ei kuulu hänen hoitoon.

4.1.1 Asiakasprofiili

Test Patient 2_1_21
Ikä 63
Syntymäaika 11.11.1955

Ilmoitukset

✉ Sähköpostiosoite patient2_1_21@sensoftia.fi

☎ Puhelinnumero +35804123123

📍 Katuosoite Microkatu 1 , 70001 , Kuopio , Suomi

📄 ☰ 📈 🔗


Kuva 11. Asiakasprofiili ammattilaisen näkymästä

Ylhäällä on "Asiakkaat" välilehden painalluksen jälkeen avautuva näkymä. Ammattilainen näkee asiakkaiden profiilia, josta voisi saada yleistä tietoa kyseisestä asiakkaasta. Tällöin estetään ammattilaisen kysymään turhia kysymyksiä, kuten "Mikä sinun nimesi on?" tai "Kuinka vanha olet?". Asiakasprofiili näkymään lisätty oikeaan ylänurkkaan "Ilmoitukset" nappi, joka näyttää modaali näkymässä asiakkaalle lähetettyjä ilmoituksia ja kyselyitä, jos niitä on (muuten kerrotaan käyttäjille, ettei ole ilmoituksia tai kyselyitä vielä luotu).

4.2 Kyselyn luonti

Modaalissa oikeasta ylänurkasta löytyy "Luo kysely" nappi, joka avaa seuraavanlaisen näkymän:

LUO KYSELY


Matti Meikäläinen

Käytä valmis kyselylomake

Tunnetko itsesi terveeksi? ✕


Häiritseekö jokin yöntasi tai nukutko huonosti? ✕

Onko sinulla jokin pitkäaikainen sairaus tai vaiva? ✕

Onko sinulla toistuvaa päänsärkyä? ✕

Onko sinulla allergioita? ✕

+ Lisää uusi kenttä

LÄHETÄ 

Kuva 12. Kyselyn luominen

Avautuu uusi modaali, jossa on aluksi tyhjä lomake. Ylhäällä ammattilainen pystyy varmistamaan vielä kerran, kenelle kyselyä luodaan, että menee määritetyt kysymykset käyttäjälle, jolla oli tarkoitus eikä väärälle henkilölle. Aluksi näytetään ammattilaiselle ainoastaan yksi tyhjä kenttä, johon voi määrittää oma kysymyksensä. Ensimmäisen kysymyksen jälkeen "Lisää uusi kenttä" nappia painamalla voi lisätä niin monta tyhjiä kenttiä, kun haluaa. Kysymyksiä voi olla vähintään yksi tai niin monta, kun halutaan.

Jokaisen kentän kohdalla on "X" ikoni, jolla voidaan poistaa kysymyksiä. "Lähetä" napin funktiossa tehdään kaikki mahdolliset tarkastukset, ennen kun kyselyä pystyy lähettämään. Kysymys saa olla pitkä, siihen ei olla puuduttu mutta mikään kentistä ei saisi olla tyhjiä tai kyselyä ei tietenkään voi lähettää, jos käyttäjä on poistanut kaikki kentät.

Jos ammattilainen haluaa yleisiä terveydenhuollon kysymyksiä esittää asiakkailleen, voi käyttää valmiista kyselylomaketta. Valmiista kyselylomakkeesta löytyy 12 yleistä kysymystä, jotka sopivat sekä

nuorille, että iäkkäille käyttäjille. Kyselylomakkeessa olevat kysymykset ovat muokattavissa omanlaiseksi, jos siltä tuntuu tai voi tietyn kysymyksen poistaa kokonaan. Tärkeintä on kirjoittaa kysymykset oikein, jotta Speech-to-text toiminto pystyy lukemaan kysymyksen oikein loppu käyttäjälle.

Jos kaikki oli hyvin, ilmoitetaan toastin avulla, että kysely on lähetetty onnistuneesti, muuten kerrotaan, mistä virhe ilmoitus tulee.

4.3 Kysely näkymä Web-käyttöliittymässä

 Kysely
 06.03.2019, 10:38 Poista kysely ✕

Kysymys	Potilaan vastaus	Vastattu (pvm & aika)
Tunnetko itsesi terveeksi?	Kyllä tunnen itseni terveeksi	06.03.2019, 14:25:28
Häiritseekö jokin yöuntasi tai nukutko huonosti?	Ei häiritse	06.03.2019, 14:26:36
Onko sinulla jokin pitkäaikainen sairaus tai vaiva?	Ei onneksi ole	06.03.2019, 14:26:50
Onko sinulla toistuvaa päänsärkyä?	Kyllä minulla on toistuvaa päänsärkyä	06.03.2019, 14:27:07
Onko sinulla allergioita?	Ainoastaan koivulle ja heinälle	06.03.2019, 14:27:23

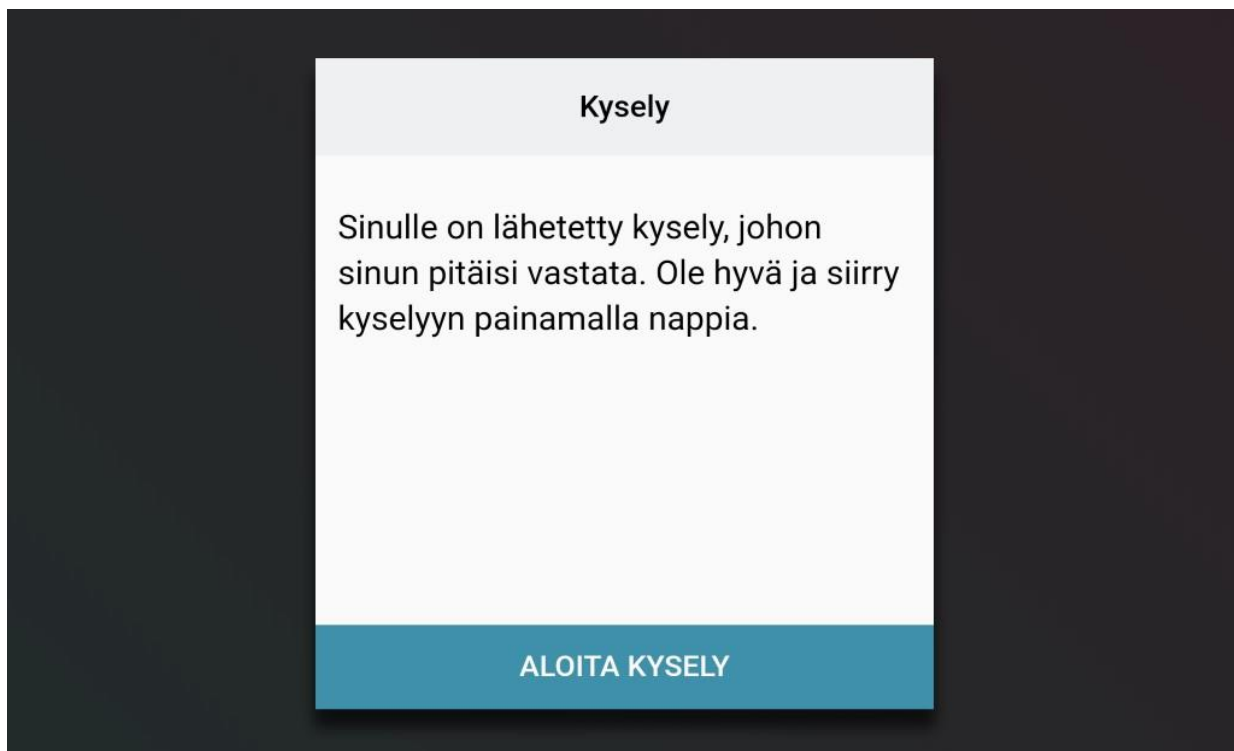
Kuva 13. Ammatilaisen kysely näkymä taulukossa

Kun kysymyksen on luotu, ne tallennetaan tietokantaan ja lähetetään (JSON data) kysely valitulle potilaalle saman tien, jonka jälkeen näytetään ammatilaiselle listassa juuri luotu kysely. Aluksi ammatilainen näkee listassa kysymykset allekkain ja vastaus kohta tyhjänä. Sitten, kun saadaan vastaukset potilaalta, tyhjät sarakkeet korvataan potilaan vastauksilla ja näytetään myös vastaus hetki, että mihin aikaan potilas on vastannut mihinkin kysymykseen (voi olla tärkeä ammatilaisten osalta).

4.4 Kyselyn vastaanotto

Kuten aiemmin mainittu, luodut kyselyt lähtevät saman tien valitulle potilaalle laite tunnistimen perusteella eli ammatilainen ei pysty (ainakaan tässä vaiheessa) ajastamaan kyselyn lähettämisaikaa.

4.4.1 Kyselyn ilmoittaminen

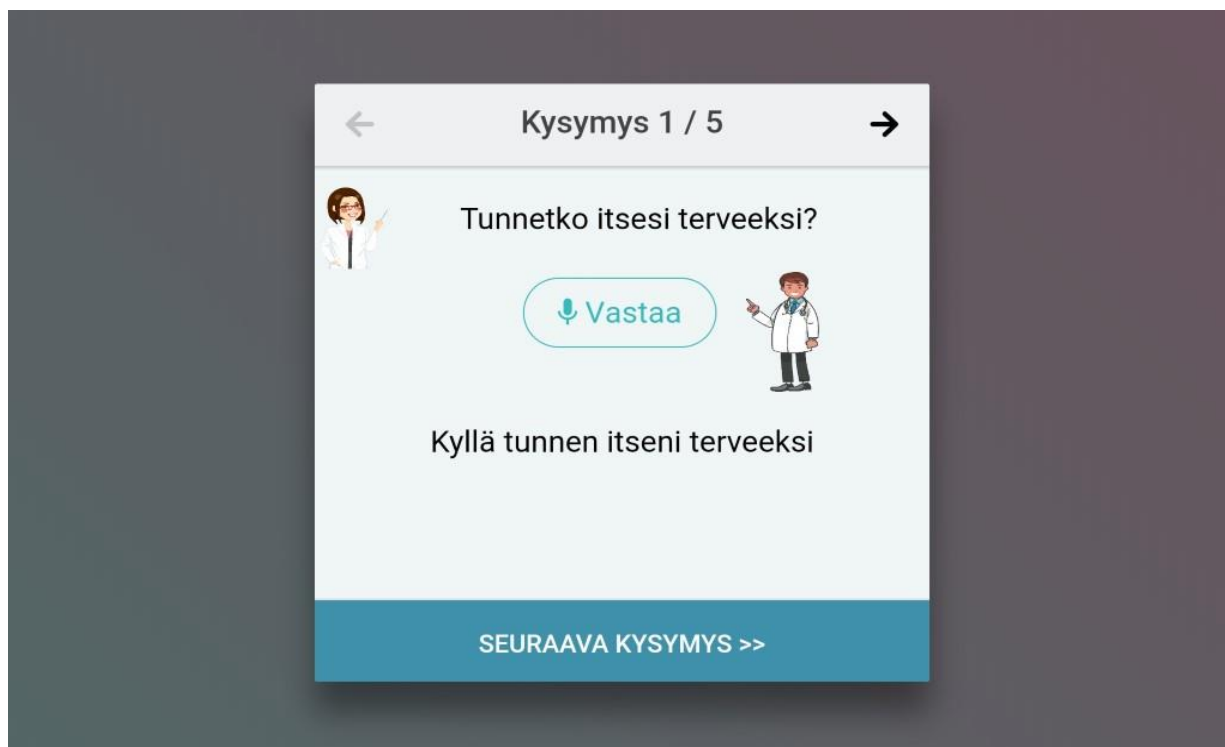


Kuva 14. Kysely ilmoitus potilaalle

Vireum-home hybridi sovellus on potilaan tabletissa auki koko ajan. Potilas on huoneessa, jonka häntä vastaava ammattilainen on luonut. Kun ammattilainen liittyy huoneeseen, hänen kuva näkyy isona ja oikeassa yläkulmassa potilaan oma kuva pienenä. Muussa tapauksessa näkyy pelkkä taustakuva.

Kun kysely saapuu perille, kuuluu äänimerkki ja näyttöön ilmestyy kuvassa 14. oleva ilmoitusnäkyvä. Ensin näytetään ilmoitus, jossa kerrotaan potilaalle mistä on kyse. Sitten, kun hän on tietoinen asiasta, että pitää vastata kyselyyn, voi aloittaa kyselyn napin painalluksen jälkeen. Ei ole pakko vastata heti, jos on jotain muuta tärkeää kesken tai ei halua vastata just silloin. "Aloita kysely" nappi ohjaa potilasta "Lobby" sivulle, jossa avataan kysely.

4.4.2 Miten chattibotti ominaisuus käytännössä toimii?



Kuva 15. Varsinainen TTS & STT toimintojen käyttö

Ylhäällä on Ionic frameworkillä toteutettu kysely modaali, jossa käydään yksi kysymys kerrallaan. Potilaalle luetaan kysymys ääneen ja odotellaan vastaus siihen. Kun, käyttäjä haluaa vastata, hänen pitää painaa "Vastaa" nappia, joka aktivoi mikrofonia ja alkaa kuuntelemaan vastausta. "Vastaa" napin väri muuttuu punaiseksi, kun aloitetaan vastauksen kuunteleminen ja kuuluu myös äänimerkki, joka tarkoittaa, että nyt voidaan vastata. Vastaa nappi vaihtuu takaisin siniseksi, kun lopettaa kuuntelemisen. Sitten, kun ollaan saatu vastaus ensimmäiseen kysymykseen, otetaan samalla talteen sen hetken tiedot muuttuinaan (päivämäärä ja kellonaika), joka näytetään myöhemmin ammattilaiselle taulukossa. Siirrytään seuraavaan kysymykseen taas napin painalluksen jälkeen. Kyselyä voidaan jättää kesken ja jatkaa myöhemmin samasta kohtaa.

Jokainen kysymys käydään samalla tavalla läpi. Luetaan ensin ammattilaisen määrittämä kysymys, siihen odotellaan vastaus ja siirrytään seuraavaan. Joitakin kysymyksiä voidaan jättää vastaamatta. Käyttäjä voi halutessaan palata edelliseen kysymykseen ja muokata vanha vastaus uudella. Se tapahtuu helposti, kun käyttäjä painaa vasen ylänurkassa olevaa "takaisin" nuoli ikonia. Tästä pääsee aina yksi kysymys taaksepäin, kunnes ollaan ensimmäisessä (jos ollaan ensimmäisessä kysymyksessä, edellinen nappi disabloidaan). Vastauksen muokkaaminen onnistuu samalla tavalla, kuin vastataan ensimmäistä kertaa eli "Vastaa" napista, joka kuuntelee uuden vastauksen ja korvaa edellinen vastaus uudella.

4.4.3 Yhteenveto kyselystä



Kuva 16. Yhteenveto kyselystä

Kun, käyttäjä on vastannut jokaiseen kysymykseen, kiitetään potilasta vastauksista (Text-to-Speech avulla) ja näytetään yhteenveto kyselystä ennen, kun lähetetään vastaukset takaisin ammattilaiselle. Tässä näkymässä käyttäjälle näytetään kaikki kysymykset ja omat vastaukset samassa taulukossa. Yhteenveto näkymässä käyttäjällä on viimeinen mahdollisuus tarkistaa ja muokata vastaukset. Viimeisenä lähetetään vastaukset, jonka jälkeen ammattilainen näkee omassa kysely taulukossaan kysymyksien lisäksi vastaukset ja vastaamishetken.

5 YHTEENVETO

Työn tavoitteena oli toteuttaa luomaan ja lähettämään kyselyitä potilaiden tabletteihin suoraan käyttöliittymästä, jossa potilaat pystyisivät vastaamaan kysymyksiin ilman manuaalista kirjoittamista. Eli tarvittiin jonkunlaista bottia, mikä osaisi lukea ammattilaisen määrittämät kysymykset yksi kerrallaan ääneen ja kuuntelemaan potilaan vastauksia. Tarkoitus oli erota muista saman alan yrityksistä ja samalla tehdä sovelluksesta entistä helppokäyttöisen.

Luonnollisen kielen puhuva chattibotti idea kuulosti tosi mielenkiintoiselta ja oli mukava tehdä alusta loppuun. Edetessä opinnäytetyössä tuli opittua uusia ohjelmointikieliä ja tosi paljon uusia asioita, joista en ollut edes kuullut aikaisemmin. Se on aina vaikeata oppia jatkuvasti uusia asioita mutta tämmöistä on tietotekniikan insinöörien työ. Teknologia kehittyy kovaa vauhtia ja meidän on pysyttävä kärryllä.

Opinnäytetyöhön olisi voinut lisätä semmoisen ominaisuuden, kun ammattilainen määrittää kysymykset allekkain, saisi myös samalla määrittää millä kielellä halutaan vastaukset. Oletuksena olisi suomi valittuna ja poikkeuksellinen tilanne ei tuottaisi ongelmia. Nyt on koodiin asetettu se kieli, jota mikrofoni kuuntelee. Toinen asia, mitä olisi hyvä on se, että ammattilainen pystyisi ajastamaan kyselyn. Päivämäärän ja kellonajan saisi valita kyselyn luonti vaiheessa ja kysely lähtisi vasta silloin perille eikä saman tien, ellei haluta heti lähettää. Ainakin näitä kahta asiaa tullaan luonnollisen kielen chattibottiin lisäämään jatkossa.

Olin ensimmäistä kertaa näin isossa projektissa mukana ja tämä oli todella hyvä kokemus. Opinnäytetyössä onnistuin hyvin ja olen onnellinen siitä, että päästiin asetettuihin tavoitteisiin aikataulussa. Sekä toimeksi antaja, että itse olemme tyytyväisiä lopputulokseen. Uskon, että tullaan VireBottia kehittämään jatkossa käyttäen tekoälyä hyödyksi, jotta saadaan yhä parempia lopputuloksia.

6 LÄHDELUETTELO

- Ami, B. (23. 08 2018). *FuzzyCloud*. (FuzzyCloud) Haettu 10. 03 2019 osoitteesta FuzzyCloud:
<https://medium.com/fuzzycloud/why-ionic-443ebff1d2fd>
- Anadea. (05. 01 2018). *Medium*. (Medium) Haettu 11. 03 2019 osoitteesta Medium:
<https://medium.com/swlh/what-is-a-chatbot-and-how-to-use-it-for-your-business-976ec2e0a99f>
- Bradley, A. (24. 07 2018). *Ionic Framework Blog*. (Ionic Framework) Haettu 10. 03 2019 osoitteesta Ionic Framework Blog: <https://blog.ionicframework.com/announcing-ionic-4-beta/>
- Brown, A. (2009). *Human resource management lessons*. Kuopio: Savonia-ammattikorkeakoulu. Liiketalouden yksikkö. Lokakuu 2009. Luento.
- Carney, B. (07. 04 2017). *Ionic Framework Blog*. (Ionic Framework) Haettu 10. 03 2019 osoitteesta Ionic Framework Blog: <https://blog.ionicframework.com/ionic-3-0-has-arrived/>
- Colotte, V. (14. 05 2017). *Citeseerx*. (Citeseerx) Haettu 10. 03 2019 osoitteesta Citeseerx:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.5121&rep=rep1&type=pdf>
- Ganatra, A. (09. 12 2015). *LinkedIn*. (LinkedIn) Haettu 10. 03 2019 osoitteesta LinkedIn:
<https://www.linkedin.com/pulse/what-ionic-framework-why-its-growing-really-fast-akash-ganatra>
- Grabianowski, E. (10. 11 2006). *howstuffworks*. (howstuffworks) Haettu 10. 03 2019 osoitteesta howstuffworks:
<https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition1.htm>
- Hakala, J. T. (2004). *Opinnäyteopas ammattikorkeakoululle*. Helsinki: Gaudeamus.
- Hindy, J. (24. 02 2019). *Android Authority*. (Android Authority) Haettu 10. 03 2019 osoitteesta Android Authority:
<https://www.androidauthority.com/best-personal-assistant-apps-android-667299/>
- Hintikka, K. A. (2009). *Twitter pikaviestii suuria uutisia*. Haettu 14. 9 2019 osoitteesta Helsingin Sanomat:
<http://www.hs.fi/arkisto/artikkeli/HS20090804SI1AT017cv>
- House, G. S. (19. 9 2017). *Gorrion Software House*. (Gorrion Software House) Haettu 10. 03 2019 osoitteesta Gorrion Software House: <https://gorrion.io/blog/angularjs-vs-angular>
- Ionic, F. (01. 01 2019). *Framework, Ionic*. (Ionic Framework) Haettu 10. 03 2019 osoitteesta Framework, Ionic:
<https://ionicframework.com/getting-started#cli>
- Jaatinen, P. (2004). *Miltä SAMKin opinnäytetyöt näyttävät toisin silmin? Satakunnan ammattikorkeakoulun vuoden 2002 opinnäytetöiden arvioinnista tehty tutkimus*. Pori: Satakunnan ammattikorkeakoulu.
- Järvinen, P. (23. 5 2007). *Ammattina esimies*. Haettu 11. 03 2019 osoitteesta
<http://www.wsoypro.fi/wsoypro.aspx?navi=Omat-sisallot§ion=books>
- Kananen, J. (2010). *Opinnäytetyön kirjoittamisen käytännön opas*. Jyväskylä: Jyväskylän ammattikorkeakoulu, liiketoiminta ja palvelut -yksikkö. Jyväskylän ammattikorkeakoulun julkaisuja 111.
- Konkka, P. (10. 01 2017). *Petri Konkka*. (Petri Konkka) Haettu 10. 03 2019 osoitteesta Petri Konkka:
<https://petrikonkka.com/fi/google-firebase-fiksu-paasee-vahemmalla/>
- Koulutus.fi. (05. 07 2018). *Koulutus.fi*. (Koulutus.fi) Haettu 11. 03 2019 osoitteesta Koulutus.fi:
<https://www.koulutus.fi/oppaat/ohjelmoinnin-suosio-kasvaa-mika-ohjelmointikieli-sopii-sinulle-13947>
- Microsoft. (31. 05 2018). *Microsoft*. (Microsoft) Haettu 10. 03 2019 osoitteesta Microsoft:
<https://docs.microsoft.com/en-us/windows/desktop/lwef/text-to-speech-engines-text-normalization>
- Miyanaga, Y. (12. 02 2015). *ResearchGate*. (ResearchGate) Haettu 10. 03 2019 osoitteesta ResearchGate:
https://www.researchgate.net/figure/Flowchart-of-a-speech-recognition-system_fig1_3451150

- Mopapa, S. (07. 05 2017). *Quora*. (Quora) Haettu 10. 03 2019 osoitteesta Quora: <https://www.quora.com/What-is-Phonetic-Analysis>
- Mölsä, H. (2005). *Verkko-opetuksen käyttöönottoon vaikuttavia tekijöitä*. (Helsingin yliopisto. Viestinnän laitos. Pro gradu -tutkielma) Haettu 11. 03 2019 osoitteesta <http://ethesis.helsinki.fi/julkaisut/val/viest/pg/molsa/verkkoop.pdf>
- pgAdmin. (07. 03 2019). *pgAdmin*. (pgAdmin) Haettu 10. 03 2019 osoitteesta pgAdmin: <https://www.pgadmin.org/>
- Polkhovskiy, N. (10. 01 2019). *Pushwoosh*. (Pushwoosh) Haettu 10. 03 2019 osoitteesta Pushwoosh: <https://help.pushwoosh.com/hc/en-us/articles/360000364923-What-is-a-Device-token>
- PostgreSQL. (14. 02 2019). *PostgreSQL*. (PostgreSQL) Haettu 10. 03 2019 osoitteesta PostgreSQL: <https://www.postgresql.org/about/>
- Pylsy, A. (2010). *Tuotemallisto nostalgian inspiroimana*. Kuopio: Savonia-ammattikorkeakoulu.
- Pynnä, S. (3. 8 2009). Satakunnan ammattikorkeakoulun henkilöstön kehittämispäivät 21.8.2009 klo 9.00-16.00 [sähköpostiviesti]. *Vastaanottaja Pirkko Tenkama*.
- Source, H. (21. 08 2012). *HTML Source*. (HTML Source) Haettu 10. 03 2019 osoitteesta HTML Source: <https://www.yourhtmlsource.com/starthere/whatishtml.html>
- Suhonen, P.;& Tenkama, P. (2010). *Raportointiohjeet*. Kuopio: Savonia-ammattikorkeakoulu.
- TECHNOLOGIES, M. (01. 01 2019). *MATRID TECHNOLOGIES*. (MATRID TECHNOLOGIES) Haettu 10. 03 2019 osoitteesta MATRID TECHNOLOGIES: <https://www.matridtech.net/5-key-features-of-angularjs-that-make-it-the-best-for-web-development/>
- Techopedia. (01. 01 2018). *Techopedia*. (Techopedia) Haettu 10. 03 2019 osoitteesta Techopedia: <https://www.techopedia.com/definition/23767/speech-to-text-software>
- Tutorialspoint. (01. 01 2019). *Tutorialspoint*. (Tutorialspoint) Haettu 10. 03 2019 osoitteesta Tutorialspoint: https://www.tutorialspoint.com/css/what_is_css.htm
- Ulrich, D. (2007). *Henkilöstöjohtamisella huipulle*. Helsinki: Talentum.
- Understood. (01. 02 2018). *Understood for learning & attention issues*. (Understood) Haettu 10. 03 2019 osoitteesta Understood for learning & attention issues: <https://www.understood.org/en/school-learning/assistive-technology/assistive-technologies-basics/text-to-speech-technology-what-it-is-and-how-it-works>
- W3schools. (01. 01 2019). *W3schools*. (W3schools) Haettu 10. 03 2019 osoitteesta W3schools: <https://www.w3schools.com/js/>
- Wickramasinghe, C. (05. 04 2017). *Medium*. (Medium) Haettu 10. 03 2019 osoitteesta Medium: <https://medium.com/@yashoda.charith10/everything-you-need-to-know-about-ionic-2-before-getting-into-developing-4e63febfc389>
- Vidgren, M. (1. 3. 2012a). Ohjeet kypsysnäytteen tarkistamiseen. *Savonian radio*. Kuopio: Savonia-ammattikorkeakoulu.
- Vidgren, M. (1. 3 2012b). OIS-ONT käytännössä. *Savonian uutiset*. Yle. Ykkönen, Kuopio.
- Wikipedia*. (15. 11 2018). (Wikipedia) Haettu 29. 03 2019 osoitteesta <https://fi.wikipedia.org/wiki/GitLab>
- Wikipedia*. (20. 02 2019). (Wikipedia) Haettu 10. 03 2019 osoitteesta <https://en.wikipedia.org/wiki/Firebase>
- Wikipedia*. (05. 03 2019). (Wikipedia) Haettu 10. 03 2019 osoitteesta <https://en.wikipedia.org/wiki/AngularJS>
- Wikipedia*. (05. 03 2019). *Wikipedia*. (Wikipedia) Haettu 10. 03 2019 osoitteesta Wikipedia: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

Wikipedia. (05. 03 2019). *Wikipedia*. (Wikipedia) Haettu 10. 03 2019 osoitteesta Wikipedia:

[https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

Vireum. (01. 10 2017). *Vireum*. (Vireum Oy) Haettu 10. 03 2019 osoitteesta Vireum: <https://home.vireum.fi/>

Yeremeeva, Y. (19. 12 2017). *Quora*. (Quora) Haettu 10. 03 2019 osoitteesta Quora:

<https://www.quora.com/What-is-GitLab>