

Informaatioteknologian tiedekunnan julkaisu
No. 68/2018

Jarno Hänninen, Jukka Juntti, Pekka Neittaanmäki, Riku Kukkanieni,
Eerik Lehtomäki, Riku Nyrhinen, Tommi Riipinen ja Matti Savonen

Tekoälypohjaisten teknologioiden testaus prototyypisovelluksella



Informaatioteknologian tiedekunnan julkaisuja
No. 68/2018

Editor: Pekka Neittaanmäki

Covers: Petri Vähäkainu ja Matti Savonen

Copyright © 2018

Petri Vähäkainu ja Jyväskylän yliopisto

ISBN 978-951-39-7652-1 (verkkokj.)

ISSN 2323-5004

Jyväskylä 2018

Tekoälypohjaisten teknologioiden testaus prototyypisovelluksilla

Jarno Hänninen
Jukka Juntti
Pekka Neittaanmäki
Riku Kukkaniemi
Eerik Lehtomäki
Riku Nyrhinen
Tommi Riipinen
Matti Savonen

Tämä julkaisu on toteutettu osana WHC-hanketta, johon Jyväskylän yliopisto on saanut rahoituksen Business-Finlandilta.

Business Finland-hanke: WHC

SANALISTA

PROTOTYYPPI	Demonstraatio sovelluksesta, joka ei ole kuitenkaan vielä tuotantoon valmis versio.
TEKOÄLY-TEKNOLOGIA	Teknologia, joka kykenee tekemään älykkäinä pidettäviä toimintoja.
CHATBOT	Tietokoneohjelma, jonka kanssa voi keskustella vapaasti.
Google STT	STT (Speech-To-Text) muuttaa puhutun äänen tekstimuotoon.
API-KUTSU	API (application programming interface) on protokolla määritelmä, jolla ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään.
FaaS	FaaS (function as a service) on verkossa toimiva funktio, jota voidaan kutsua sovelluksesta esimerkiksi API-kutsulla.
SANAVEKTORI	Dokumentin luokitteluun käytetty menetelmä, joka kertoo jokaisen dokumentissa esiintyvän sanan perusmuodon ja frekvenssin.
NLU	NLU (Natural Language Understanding) tarkoittaa luonnollisen kielen ymmärtämistä eli ihmisten välisessä keskustelussa käytettyä kieltä.
PUHEOHJAUS	Ohjelmaa voidaan ohjata puheen avulla.
QR-KOODI	Kaksiulotteinen kuviokoodi, johon on koodattu informaatiota. QR-koodeja voidaan lukea matkapuhelimella erillisten sovellusten avulla.
LEMMATISOINTI	Sanan muuttaminen perusmuotoon.

KUVIOT

KUVIO 1. Kuvankaappaus Watson Assistantin käyttöliittymästä	2
KUVIO 2. Liikuntakyselyssä käytetty opetusdata	5
KUVIO 3. Liikuntakyselyssä käytetty tietokannan funktio-sarja	6
KUVIO 4. Liikuntamotivaatio prototyypin käyttöliittymänäkymä	7
KUVIO 5. MuistiAppi prototyypin aloitusnäkymä	8
KUVIO 6. MuistiAppi prototyypin piirrostehtävä	10
KUVIO 7. MuistiAppi prototyypin kellotaulu tehtävä	11
KUVIO 8. MuistiAppi prototyypin tulospäätelmä	12
KUVIO 9. Aulasovelluksen käyttöliittymänäkymä	16
KUVIO 10. Aulasovellus prototyyppi testikäytössä Agora-rakennuksessa	17
KUVIO 11. Stressitesti prototyypin käyttöliittymä	20
KUVIO 12. Stressitesti prototyypin tulospäätelmä	21
KUVIO 13. Virtuaalilääkäri prototyypin aloitusnäkymä	23
KUVIO 14. Virtuaalilääkäri -prototyypin diagnoosityökalun toiminta	24
KUVIO 15. Virtuaalilääkäri prototyypin potilasnäkymä	25
KUVIO 16. Verenpaineen visualisointi	26

SISÄLLYSLUETTELO

1	Johdanto	1
2	IBM Watson -Palvelutarjoma	2
3	Prototyypit	4
3.1	Liikuntamotivaatiokysely	4
3.1.1	Liikuntamotivaatiokysely -prototyypissä käytetyt teknologiat	4
3.1.2	Liikuntamotivaatiokysely prototyypin tulokset	6
3.1.3	Liikuntamotivaatiokysely prototyypin pohdinta ja jatkokehitys	7
3.2	Muistitesti toteutettuna mobiilialustalla	8
3.2.1	MuistiAppi prototyypissä käytetyt teknologiat	9
3.2.2	MuistiAppi prototyypin tulokset	12
3.2.3	MuistiAppi prototyypin pohdinta ja jatkokehitys	13
3.3	Puheohjattu aulasovellus	15
3.2.1	Aulasovellus -prototyypissä käytetyt teknologiat	16
3.2.2	Aulasovellus prototyypin tulokset	17
3.2.3	Aulasovellus -prototyypin pohdinta ja jatkokehitys	18
3.4	Stressitesti	19
3.4.1	Stressitesti prototyypissä käytetyt teknologiat	19
3.4.2	Stressitesti -prototyypin tulokset	21
3.4.3	Stressitesti -prototyypin pohdinta ja jatkokehitys	21
3.5	Virtuaalilääkäri	22
3.5.1	Virtuaalilääkäri -prototyypissä käytetyt teknologiat	23
3.5.2	Virtuaalilääkäri -prototyypin tulokset	24
3.5.3	Virtuaalilääkäri -prototyypin pohdinta ja jatkokehitys	25
4	Yhteenveto	27
	LÄHTEET	29

1 Johdanto

Tässä raportissa paneudutaan IBM Watson Cloud –palveluihin, käydään läpi kokemuksia niiden käytöstä ja verrataan niitä muiden toimijoiden vastaaviin. Tarkoituksena on antaa lukijalle ymmärrys palveluiden mahdollisuuksista, rajoitteista ja sijoittumisesta vastaaviin ratkaisuihin nähden. Prototyyppekehityksen tarkoituksena on ollut tiedon kerääminen teknologioiden toimivuudesta ja ominaisuuksista kehitysprosessin aikana, sekä tiedon esille tuominen konkreettisin esimerkein sovellusten avulla.

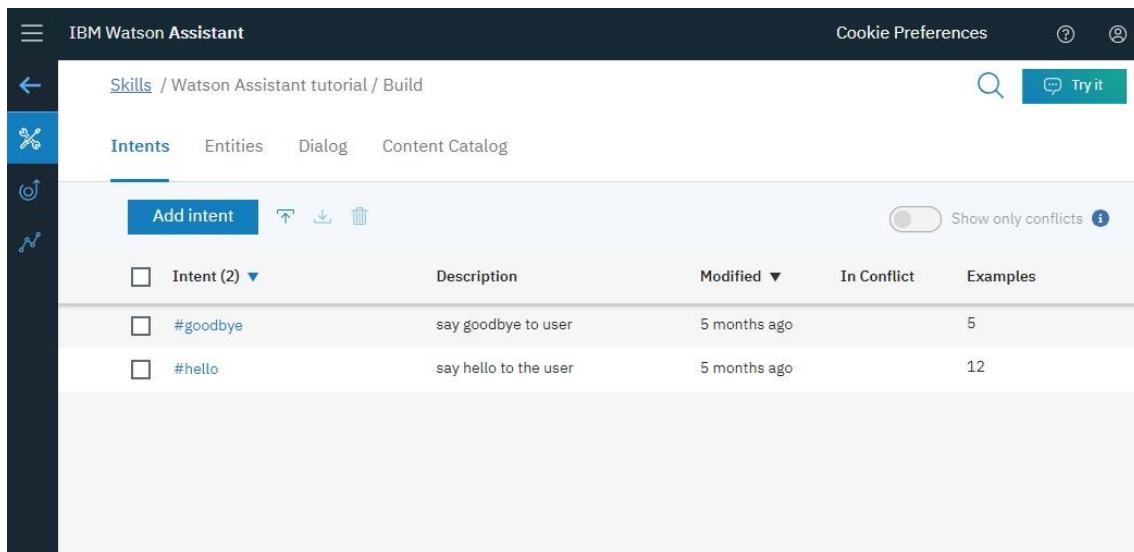
Kuvaamme raportissa vuoden 2018 aikana Watson Health Cloud Finland -projektissa toteutettuja prototyyppeisovelluksia, niissä käytettyjä teknologioita ja havaintoja käytetyistä tekoälyteknologioista. Prototyyppeisovellusten kehitys on jatkumo vuoden 2017 aikana Value From Public Health Data -hankkeessa toteutetuille prototyypeille, jotka on raportoitu kyseisen hankkeen loppuraportissa. Kehitys tapahtui käytännön toteutuksena kuuden hengen työryhmällä, josta kehitystyöhön täysimääräisenä kului neljän hengen työpanos. Tämä kertoo suuntaa-antavasti kehitystyön investointivaatimuksista.

Motivaationa prototyyppeisovellusten toteuttamiselle osana Watson Health Cloud Finland -hanketta on ollut terveydenhuollon prosessien ja ihmisten terveyden edistämiseen tähtäävä teknologiatestaus. Tarkoituksena ei ole ollut lopullisten tuotteistettujen palveluiden tuottaminen, vaan tutkimuksen kannalta oleellinen testaus. Pienimuotoiset prototyyppeisovellukset on myös luotu siten, että ne voidaan nähdä suurempaa kokonaisuutta. Prototyyppeisovellusten potentiaali tuleekin nähdä terveydenhuollon kustannuskehitykselle positiivisena vaikutuksena, jossa toistuvien tehtävien automaatiolla ja edistyksellisten teknologioiden käyttöönotolla pyritään nostamaan kustannustehokkuutta.

2 IBM Watson -Palvelutarjoma

Watson Assistant –palvelu

Watson Assistant -palvelu on luonnollisen kielen tunnistamiseen kehitetty tekoälyteknologia, joka pyrkii muuttamaan ihmisen käyttämän kielen tietokoneelle ymmärrettävään muotoon. Watson Assistant -palvelua voidaan käyttää tiettyjen sanojen tunnistamiseen vapaasti kirjoitetusta tekstistä tai chatbottien rakentamisessa. Palvelulle täytyy ensiksi opettaa sanat, jotka sen halutaan tunnistavan. Opetus tapahtuu käyttöliittymäpohjaisessa IBM Cloud -palvelussa (Kuvio 1), jossa pääsee käyttämään myös muitakin IBM Watson palveluita. Palvelun voisi esimerkiksi opettaa tunnistamaan erilaisia sanoja kuten "Hei", "Terve", "Mitä kuuluu", jotka voisi kohdentaa otsikon "Tervehdys" alle. Nyt palvelu tunnistaisi nämä sanat vapaasta tekstistä tervehdykseksi ja palvelun käyttäjä voisi määrittää mitä tapahtuu, jos palvelu tunnistaa jotain sille opetettuja sanoja vapaasta tekstistä.



KUVIO 1. Kuvankaappaus Watson Assistantin käyttöliittymästä

Kun Watson Assistant -palvelulle on opetettu halutut tunnistettavat sanat, voidaan sitä käyttää osana jotain sovellusta. Palvelun käyttö sovelluksesta käsin tapahtuu API-kutsujen avulla, johon IBM tarjoaa kattavat ohjeet dokumentaationsivuillaan (Watson Assistant, 2018a). API-kutsut tukevat Java, Node, Python ja Ruby ohjelmointikieliä.

Watson Assistant -palvelu tukee tällä hetkellä 13:a erilaista kieltä, joihin ei lukeudu suomen kieli (Watson Assistant, 2018b). Hankkeessa toteutetuissa prototyypeissä olemme kuitenkin todenneet, että palvelu on osittain kieliriippumaton, eli sen voi opettaa tunnistamaan myös suomenkielisiä sanoja. Palvelusta on tarjolla ilmais-, normaali- ja korkean tasoinen versio. Ilmaisisessa kokeiluversiossa on tarjolla 10 000 API-kutsua kuukaudessa (Watson Assistant, 2018c), mikä riittää hyvin prototyyppien ja

vähän käytettyjen sovellusten käyttöön. API-kutsujen täytyttyä palvelu lopettaa toimintansa ja se on päivitettävä maksulliseen versioon. Korkean tason versio tarjoaa palvelulle paremman tietoturvallisuuden ja datansiirtonopeuden.

Watson Assistant -palvelu julkaistiin vuonna 2016, mutta se ei ole mitenkään uusi keksintö. Vastaavanlaisia tekoälyteknologioita löytyy myös muilta suurilta yrityksiltä kuten Googlelta, Microsoftilta ja Facebookilta. Isojen yritysten teknologioiden lisäksi markkinoilla on paljon pienempiä vastaavanlaisia teknologioita ja teknologian ohjelmoiminen itse ei ole mitenkään mahdoton tehtävä. Taulukossa 1 on esitelty Watson Assistantin vastaavien palveluiden tietoja. Tiedot perustuvat raportin kirjoitushetkellä oleviin tietoihin ja tiedot, kuten tuettujen kielten määrä, saattavat muuttua.

	Watson Assistant	Wit.ai	LUIS	Dialogflow
Omistusyhtiö	IBM	Facebook (2015)	Microsoft	Google (2016)
Julkaisuvuosi	2016	2013	2015	2010
Tuetut kielet	13	73	11	21
Käyttökustannukset	10 000 API-kutsua ilmaiseksi kuukaudessa.	Täysin ilmainen, ei käyttörajaa.	10 000 API-kutsua ilmaiseksi kuukaudessa.	15 000 ääni-API-kutsua ilmaiseksi kuukaudessa. Rajaton teksti-API-kutsujen käyttö.

Taulukko 1. Watson Assistant verrattuna muihin vastaaviin palveluihin

Hankkeessa toteutetuissa prototyypeissä olemme keskittyneet käyttämään Watson Assistantia, mutta olemme myös testanneet sen kilpailevaa palvelua Wit.ai:ta, joka tarjoaa palveluunsa suomenkielisen tuen. Vaikka Wit.ai väittääkin tarjoavansa suomenkielisen tuen, emme ole toistaiseksi huomanneet siinä mitään eroavaisuuksia suomen kielen tuen kannalta Watson Assistantiin verrattuna. Avoimeen lähdekoodiin perustuvalla Rasa NLU:lla on erilainen lähestymistapa kielitukeen. Teknologialle pystyy määrittämään oman sanavektorin, jonka perusteella se tunnistaa haluttua kieltä. Teknisistä ongelmista johtuen, emme ole kuitenkaan päässeet kunnolla testaamaan, tunnistako Rasa NLU:n tapainen teknologia paremmin suomen kieltä kuin esimerkiksi Watson Assistant. Seuraavissa luvuissa raportoimme hankkeessa toteutetuista prototyypeistä, joissa Watson Assistant -palvelu on ollut keskeisessä roolissa osana toteutusta.

3 Prototyypit

3.1 Liikuntamotivaatiokysely

Liikuntamotivaatiokysely on puheohjatusti toimiva kysely, jonka avulla voidaan selvittää vastaajan liikuntamotivaatio-profiili. Kysely koostuu 40 monivalintakysymyksestä, joihin käyttäjä vastaa puhuen. Perinteisessä kyselyssä vastaaja vastaa vaihtoehdoilla 1 - 5 (täysin eri mieltä, jokseenkin eri mieltä...). Toteutetussa prototyyppi-sovelluksessa kyselyn väittämiin vastataan luonnollisella puhekielellä, jolloin sovellus tunnistaa vastauksesta sanoja ja lauseita, sekä arvottaa vastaukset samalla tavoin kuin perinteisessä kyselyssä välille 1 - 5. Kyselyn vastaukset järjestelmä analysoi ja antaa lopuksi vastaajalle palautteen sekä keskiarvon muiden vastauksista.

Prototyypin tavoitteena oli testata IBM:n Watson Assistant teknologiaa, ja erityisesti toimivuutta suomen kielellä. Erityisen mielenkiintoista tästä teki sen, että lähtökohtaisesti IBM ei lupaa palveluilleen suomen kielen tukea. Ajatuksena oli aluksi käyttää IBM:n omaa käännöstyökalua, mutta kielituen puutteen vuoksi suunnittelimme käyttävämmä Googlen Translate palvelua. Kuitenkin huomasimme, että IBM:n toteutus Watson Assistantille on kieliriippumaton, eikä erillistä käännöstyökalua tarvita ollenkaan. Teknologian testaamisen lisäksi haluttiin myös testata puheohjatun kyselyn toimivuutta vaihtoehtona perinteiselle paperiselle kyselylomakkeelle.

3.1.1 Liikuntamotivaatiokysely -prototyypissä käytetyt teknologiat

Liikuntamotivaatiokyselyn -prototyyppi toteutettiin Java-pohjaisena Android-sovelluksena, joka nähtiin sen levinneisyyden vuoksi hyvänä alustana toteuttaa prototyyppi sekä testata sen toimivuutta eri kehitysvaiheissa. IBM tarjoaa palveluilleen hyvät dokumentaatiot useille eri ohjelmointikielille, myös Javalle, ja niiden turvin on hyvä päästä alkuun IBM:n teknologioiden integrointiin omiin sovelluksiin.

Keskityimme prototyyppi-sovelluksessa tekoälyteknologioista erityisesti luonnollisen kielen analysointiin ja puheentunnistukseen, hyödyntämällä IBM:n Watson Assistant palvelua kielen analysointiin ja Googlen Speech-to-Text palvelua puheentunnistukseen. Koulutimme tekoälyn tunnistamaan yleisimmät tavat vastata annettuihin kysymyksiin (kuvio 2), ja pikkuhiljaa kun sovellusta testattiin ja käytettiin erilaisissa koetilaisuuksissa, opetusdata kasvoi ja tekoälyä kyettiin opettamaan paremmin tunnistamaan erilaisia tapoja vastata kysymyksiin. Opetusvaihe onkin kielentunnistuksessa tärkein, mutta myös usein työläin vaihe koska kehittäjän täytyy erikseen opettaa tuntemattomat sanat koneelle.

← | #agree

Intent name
#agree

Description
Add a description to this intent

Add user examples
Add user examples to this intent

Add example

User examples (23) ▼

Aika usein ✎

ehdottomasti ✎

ehkä joo ✎

Hankala sanoa todennäköisesti kyllä ✎

ilman muuta ✎

jep ✎

jepjep ✎

KUVIO 2. Liikuntakyselyssä käytetty opetusdata

Watson Assistant tunnistaa vain tekstimuotoista kieltä, joten puhuttu ääni piti muuttaa tekstiksi. Tässä käytimme avuksi Googlen STT -palvelua, joka on sisäänrakennettuna useimmissa Android-laitteissa. Laitteen sisäänrakennetun puheentunnistusjärjestelmän rajoitteena on se, että se käyttää samaa kieltä kuin järjestelmä, joten esimerkiksi ruotsinkielinen käyttöjärjestelmä yrittää tunnistaa vain ruotsinkielistä puhetta. Prototyyppi-vaiheen kehityksessä ongelma huomioitiin, mutta ratkaisu jää mahdolliselle jatkokehitysvaiheelle ja käytimme aina suomenkielistä järjestelmää.

Lisäksi hyödynsimme IBM:n CloudantNoSQL-tietokantaa, jonne vastaukset ja kysymykset tallennettiin. Tiedot on tallennettu helposti luettavaan JSON-muotoon ja niiden hyödyntäminen onnistuu helposti HTTP-protokollan avulla. Lisäksi käytimme hyödyksi IBM Functions palveluita, joka on IBM:n tarjoama FaaS –palvelu (kuvio 3). Tietokannan vastauksista laskettiin keskiarvo, joka toteutettiin Funktioilla.

Sequence Actions

ACTIONS	PACKAGE
list-documents	liikunta-cloudant-average
getRev	Default Package
calcAverage	Default Package
update-doc	Default Package

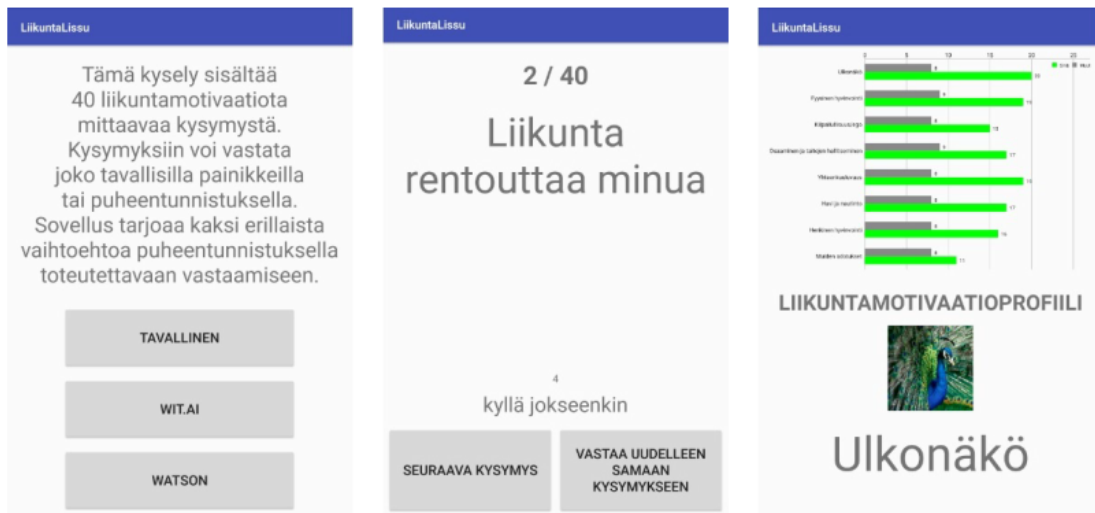
KUVIO 3. Liikuntakyselyssä käytetty tietokannan funktio-sarja

3.1.2 Liikuntamotivaatiokysely prototyypin tulokset

Prototyypin tutkimuksellisenä tavoitteena oli mitata Watson Assistant:in NLU teknologian toimivuutta erityisesti suomen kielellä. Mielenkiintoisena havaintona tehtiin, että vaikka IBM listaa hyvin suppean listan tuetuista kielistä sivuillaan, niin käytetty tekniikka kielen analysointiin Watson Assistantilla on kieliriippumaton. Tämän ansioista pystyimme opettamaan tekoälyä ymmärtämään suomenkielisiä sanoja ja lauseita sille syötetystä saman kielisestä tekstistä.

Prototyypissä tutkimme myös Watson Assistantin rinnalla toista kilpailevaa tuotetta, joka lähtökohtaisesti NLU:n osalta vastasi samoja ominaisuuksia Watson Assistantin kanssa, erityisesti olimme kiinnostuneita testaamaan palveluiden eroja, koska kilpaileva tuote valmistajan mukaan tukee suomen kieltä. Liikuntakyselytestissämme eri valmistajan tuotteilla ei noussut merkittäviä eroja esille vastausten tarkkuudessa. Huomioitavaa on, että testitapaukset olivat hyvin yksinkertaistettuja tapauksia, joissa järjestelmä kategorisoi vastaukset asteikolla 1 - 5 tai 'en ymmärrä vastausta' ryhmiin.

Käytettävyyden näkökulmasta keskeisenä huomiona tehtiin se, että pitkäkö 40 kysymyksen kysely on hyvin uuvuttavaa vastata puhumalla ja on hyvin todennäköistä, että vastaaja keskeyttää tai ei ainakaan keskity kyselyn loppuun saakka. Nähtiinkin, että perinteinen 1- 5 asteikolla toteutettava kysely on selkeästi käyttäjäystävällisempi ja tehokkaampi tapa vastata kyselyyn.



KUVIO 4. Liikuntamotivaatio prototyypin käyttöliittymänäkymä

3.1.3 Liikuntamotivaatiokysely prototyypin pohdinta ja jatkokehitys

Liikuntamotivaatiokysely oli mielestämme mielenkiintoinen ja onnistunut prototyyppi, vaikkakaan se ei ollut riittävän haastava tuomaan selkeästi esille eroja eri valmistajien ratkaisujen välillä. Se kuitenkin osoitti, että ainakin yksinkertaisten käyttötapauksen kohdalla on teknologia jo riittävällä tasolla tuomaan lisäarvoa käyttäjälle ja kehittäjille.

Jatkotutkimuksena voitaisiin nähdä mielenkiintoisena toteuttaa erilaisia vasteaikojen mittauksia sekä testejä, jotka suunniteltaisiin haastavimmiksi, jolloin saataisiin teknologioiden erot ja ongelmakohdat esille. Esimerkiksi haastetta voitaisiin muodostaa monimutkaisemmilla syötteillä sekä lauserakenteilla, joissa sisällön rakenne on merkityksellinen lopputuloksen kannalta. Prototyypin kehitykseen kului aikaa noin kaksi kuukautta ja kehityksessä oli mukana kolme henkilöä.

3.2 Muistitesti toteutettuna mobiilialustalla

MuistiAppi on Android-pohjainen prototyyppi-sovellus mobiililaitteille. Sovellus suorittaa käyttäjälle tämän muistikuntoa arvioivan testin, joka pohjautuu validoituun ja laajasti käytettyyn mini-mental state examination (MSSE) testiin. Testissä käyttäjältä kysytään sarja erilaisia kysymyksiä sekä pyydetään suorittamaan erilaisia tehtäviä. Kysymyksiin ja tehtäviin vastataan ohjeiden mukaan puhumalla, kirjoittamalla tai suorittamalla jokin toimenpide. Kysymyksillä testataan käyttäjän orientaatiota aikaan ja paikkaan, muistia ja ymmärrystä.



KUVIO 5. MuistiAppi prototyypin aloitusnäky

Testin tehtävät voidaan jaotella 5 erilaiseen osa-alueeseen, joissa **piirtotehtävässä** on piirrettävä mallin mukainen kuva. **Asettelutehtävässä** testataan kolmen ohjeen noudattamista oikein ja oikeassa järjestyksessä. Testattavan on vedettävä pallo neliön sisään, käännettävä tabletti vaaka-asentoon ja painettava vasenta painiketta. **Luetunymmärtämistehtävässä** testattavaa pyydetään lukemaan ruudulla näkyvä teksti ääneen ja noudatettava sen ohjetta. Ruudulla näkyy teksti ”SULKEKAA SILMÄNNE” luettuaan tämän käyttäjän on toimittava ohjeiden mukaan. **Kirjoitustehtävässä** testattavaa pyydetään kirjoittamaan vapaasti kieliopillisesti järkevä lause. Sisällöllä ei ole väliä. **Kellotehtävässä** testattavan on koottava ruudulla olevista osista kellotaulu järkeväksi ja asetettava se näyttämään annettua kellonaikaa.

3.2.1 MuistiAppi prototyypissä käytetyt teknologiat

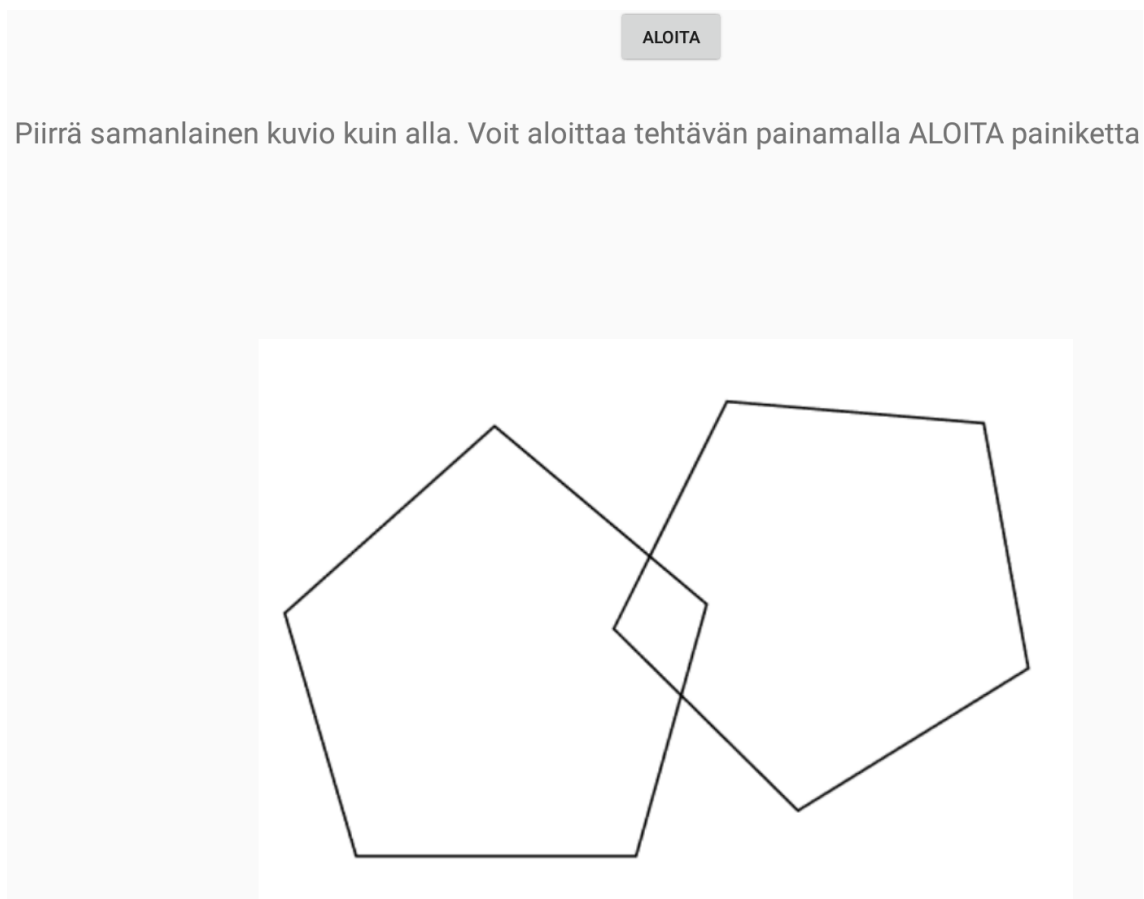
MuistiAppi –sovellus on toteutettu Android-sovelluksena Java-ohjelmointikielellä. Prototyypin haluttiin olevan käytettävissä tabletilla, jolloin se olisi helposti saatavilla ja käytettävissä kaikille käyttäjille. Android-tabletit ovat myös suosittuja ja kohtuullisen edullisia, minkä johdosta niiden käyttäminen testaus- ja kehityskäytössä on perusteltua, koska voidaan olettaa sovellusten käyttäjien käyttävän vastaavantyyppisiä laitteita. Tablet-laitteet eivät vaatineet erikseen ladattavia tai hankittavia lisäosia, ainoastaan kehitetyn sovelluksen lataamisen.

MuistiAppi käyttää puheen muuntamiseen tekstiksi Googlen STT -palvelua, jota myös aikaisemmassa prototyypissä on käytetty. Sen avulla pystyttiin kääntämään puhuttu ääni tekstiksi helposti ja Googlen STT tukee suomen kieltä. Muodostettu teksti lähetettiin IBM:n Watson Assistant palveluun, jossa tekoälyä käytettiin tunnistamaan sanoja ja löytämään haluttuja asioita tekstistä.

Prototyypissä käsiteltiin myös kirjoitettua tekstiä, tehtävässä jossa käyttäjää pyydetään kirjoittamaan suomenkielinen lause. Lauseen analysointiin käytettiin tekstianalytiikan menetelmiä, joiden avulla kyetään analysoimaan lauseen kieliopillista oikeellisuutta. Hyödynsimme Turun yliopiston alaisen BioNLP Group -tutkimusryhmän kehittämää avoimen lähdekoodin Finnish Dependency Parseria, joka kykenee lemmatisoimaan sanoja sekä tunnistamaan sanan sanaluokan. Lemmatisointi on tekstianalytiikan menetelmä, jossa sanat muutetaan niiden perusmuotoon (elatiivi: kaupasta -> nominatiivi: kauppa (substantiivi)). Jyväskylän Yliopistolla on myös oma palvelinympäristö, jossa Finnish Dependency Parser on käytettävissä. Lisäksi hyödynsimme projektissa toimivan tutkimusavustaja Riku Nyrhisen kehittämää algoritmia, joka tunnistaa lauseen kieliopillisen oikeellisuuden.

Prototyyppi hyödynsi kuvantunnistusmenetelmiä siihen, että se havaitsi käyttäjän kasvoista onko tämän silmät auki vai kiinni. Siinä hyödynnettiin todennäköisyyslaskentaa, jossa järjestelmä laskee kasvoista otetusta kuvasta todennäköisyyden sille, että onko käyttäjän silmät auki vai kiinni. Alkuperäisen tarkoituksena oli käyttää IBM:n tuottamaa Watson Visual Recognition palvelua, mutta kun havaitsimme, että palvelun käyttö vaatii opetusdatamateriaalin, jolla se voidaan kouluttaa havaitsemaan suljetut silmät kasvoista. Päädyimme hyödyntämään Googlen Vision -kuvantunnistusohjelmaa, joka käyttää matemaattista mallia laskemaan silmien aukioloa.

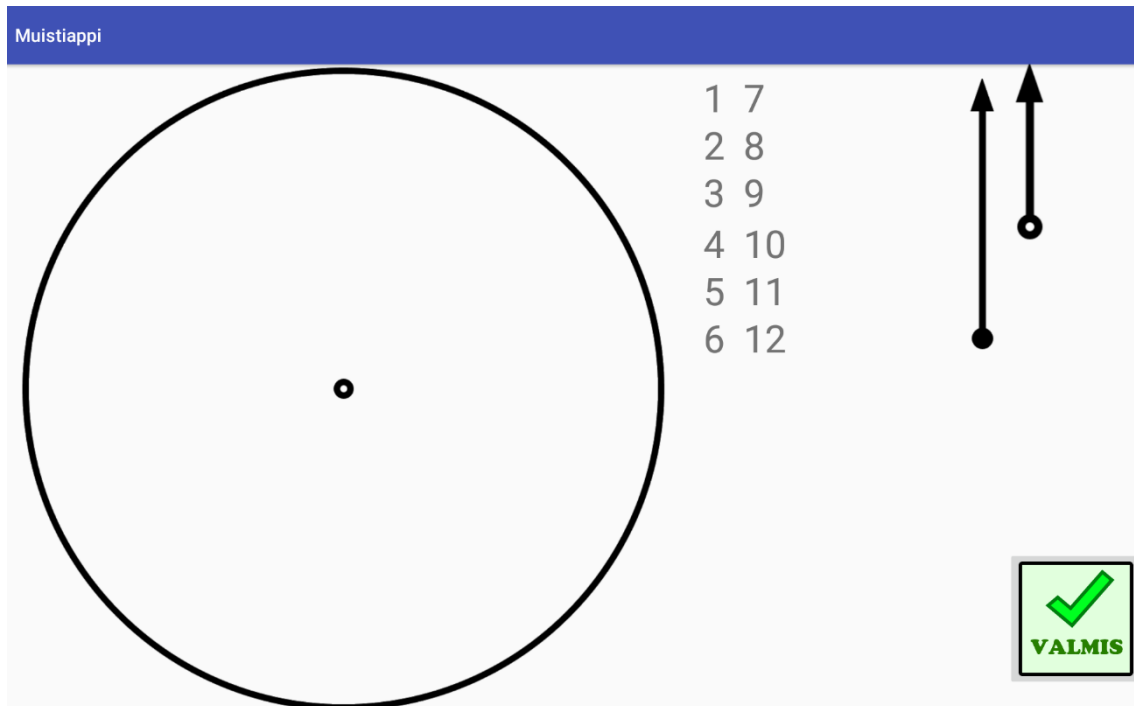
Prototyypin piirtotehtävien analysointiin hyödynnettiin itse kehitettyä algoritmia, joka vertaili piirrosjäljestä suoria ja janoja (kuvio 6). Analysoinnissa hyödynnettiin euklidisen tasogeometrian sääntöjä, johon perustuen kyettiin laskemaan piirretyt kuvat, kuvioiden leikkauskohdat ja tekemään analyysi siitä, että onko kuva piirretty oikein.



KUVIO 6. MuistiAppi prototyypin piirrostehävä

Kellon asettamistehtävässä pyydetään rakentamaan kellotaulu annetuista osista, eli numeroista ja viisareista (kuvio 7). Ja käyttäjän täytyy osata asettaa numerot oikeille paikoilleen ja viisarit osoittamaan pyydettyä kellonaikaa. Kellotaulun analysointiin

käytettiin matemaattista mallia, jossa lasketaan jokaiselle numerolle oikeat sijainnit ja analysoidaan, että onko käyttäjä asettanut oikean numeron sille varatulle paikalle. Lisäksi analysoidaan viisareiden asentoa ja lasketaan osoittaako kello haluttua aikaa.



KUVIO 7. MuistiAppi prototyypin kellotaulu tehtävä

Testin päätyttyä saa käyttäjä itselleen arvion omasta suoriutumisestaan sekä yhteenlasketut pisteet (kuvio 8). Järjestelmä näyttää myös pisteiden jakautumisen eri tehtävien välille.

Muistiappi

Näin suoriuduitte testissä

Testin tulokset **pistemäärä / maksimipisteet**

26 / 33

Lisätietoja

Tehtävä 1

Annettu vastaus (mitä laite ymmärsi): 2018

Oikea vastaus: 2018

Saadut pisteet 1/1

Tehtävä 2

Annettu vastaus (mitä laite ymmärsi): talvi

Oikea vastaus: syksy

Saadut pisteet 0/1

Tehtävä 3

Saadut pisteet 0/0

KUVIO 8. MuistiAppi prototyypin tulostulosnäky

3.2.2 MuistiAppi prototyypin tulokset

Puheentunnistuksessa käytetty Googlen STT -palvelu osoittautui käytössä hyvin toimivaksi, sen tarkkuus suomen kielellä oli hyvää ja pääsääntöisesti testikäytössä tunnisti sanat oikein. Ongelmaa aiheutti ajoittain foneettisesti lähellä toisiaan olevat sanat kuten esimerkiksi "perho" ja "verho". Myös nopea ääntäminen aiheutti ajoittain ongelma, jolloin palvelu ei tunnistanut kaikkia sanoja tai yhdisti esimerkiksi kolme erillistä sanaa lauseeksi, kuten sanoista "harja, tuoli, sieni" muodostui "harjattu oli sieni". Suomen kielen mielivaltaisuus yhdyssanojen suhteen tuotti myös ajoittain haasteita, sanapareista "leipä" ja "kirja" tuli "leipäkirja". Lisäksi murteet ja heikko ääntäminen tuottivat heikentyneitä tuloksia puheentunnistuksessa.

Puheentunnistuksen tarkkuutta paransi selkeät tauot ja rauhallinen ääntäminen. Pyrimme myös tehtävissä valitsemaan käytettyjä sanoja siten, että ne olisivat ääntämiseltään mahdollisimman poikkeavia keskenään sekä selkeitä, jolloin niiden tunnistaminen helpottuu.

Prototyypissä hyödynnettiin IBM:n Watson Assistant palvelua, jota käytettiin päivämäärien tunnistuksessa puheesta. Numeraalien ja päivämäärien ilmaisemiseen on suomen kielessä lukuisia erilaisia tapoja, joka tuotti hankaluuksia. Watson Assistantin käyttämän tekoälyn turvin pystyimme yhdistämään eri tavoin ilmaistuja päivämääriä ja vertailemaan sitä haluttuun päivämäärään. Aluksi käytimme itse luotua taulukoita hyödyksi, mutta Watson Assistantin Fuzzy Matching -ominaisuus helpotti ja mahdollisti syötteiden yhdistämisen numeroihin, eikä numeroita tarvinnut erikseen opettaa järjestelmälle.

Prototyypissä käytetty kuvantunnistusohjelmisto Google Vision teknologia toimi hyvin, mutta vaatii hyvän valaistuksen, jotta käyttäjästä sen ottama kuva olisi hyvä ja selkeä. Lisäksi järjestelmä vaatii käyttäjää pitämään tablettia kohtalaisen suorassa kohti käyttäjän kasvoja, jotta saadaan hyvä kuva kasvoista. Haasteena nähtiin käyttäjät, joilla on vaikeuksia pitää tablettia vakaana ja ylhäällä tarpeeksi kauan, jotta järjestelmä kykenee suorittamaan kasvojen analyysin.

3.2.3 MuistiAppi prototyypin pohdinta ja jatkokehitys

MuistiAppin tarkoitus on mahdollistaa muistitestin ja yleisen kognitiivisen kyvykkyyden testaamiseen kotiloissa ja hoitolaitoksissa ilman erillistä valvontaa. Sovellus pystyisi valvomaan testattavaa henkilöä kuten ihminen ja kirjaamaan ylös testin aikana tehtyjä havaintoja.

Tällä hetkellä prototyyppi kykenee hyvin rajatusti suorittamaan yksinkertaisen muunnelman testistä ja osoittaa mielestämme, että jo nykyisillä työkaluilla automatisoitu testaaminen on ainakin osittain mahdollista. Kuitenkin rajoitteena nähdään selkeästi se, että tekoäly-sovellukset eivät kuitenkaan kykene samanlaiseen intuitiiviseen, ammatilliseen tai kokemukselliseen arviointiin kuten ihminen ja siten osa testin tuomasta informaatiosta katoaa.

Prototyypin tai oikeammin MMSE-testin rajoitteena on muistettava kehityksessä se, että itsessään ja yksinään se ei ole validi diagnosointi- tai seurantatyökalu kaikkiin muistisairauksiin ja vaatiikin aina terveydenhuollon ammattilaisten mukana oloa hoidossa, seurannassa ja diagnosoinnissa. MMSE –testin luonteen ja rajoitteiden vuoksi toteutettua sovellusta ei voida käyttää muistisairauden ennakoivaan tunnistamiseen, mutta käytetyn teknologian pohjalta toteutettuna myös aikaisen hoitotarpeen ennakoiva tunnistaminen olisi mahdollista, mikäli siihen käytetään siihen suunniteltua lääketieteellistä pohjaa.

Ensimmäinen koko testin kattava versio prototyypistä ei ole vielä nähnyt suurta joukkoa testausta ja jatkokehitykselle on vielä paljon tilaa. Jatkokehitys tulisi sisältämään käytettävyyden parantamista ja lisää luonnollisen kielen prosessointia esimerkiksi vastausten relevanttiuden arvioinnissa. Lisäksi potentiaalista jatkokehitystä voisi olla uusien testimuotojen lisääminen sovellukseen, kuten MoCa –testin tyyliä syvempää keskittymistä testaavia testejä. MuistiAppi-prototyyppiä on kehitetty yhteistyössä Muistiliiton asiantuntijoiden kanssa ja tavoitteena on siirtää MuistiAppi Muistiliiton sisäiseen testaukseen, ja mahdollisesti potilastestaukseen tulevaisuudessa. Prototyypin kehitykseen kului aikaa noin kolme kuukautta ja sitä työsti kolme henkilöä.

3.3 Puheohjattu aulasovellus

Prototyypin ideana oli testata puheohjatun aulasovelluksen toimivuutta isoissa sairaaloissa, joissa erilaisten huoneiden ja henkilöiden löytäminen voi olla haastavaa. Sovelluksen vaatimuksena oli liikkuvuus, jolloin sitä voisi käyttää mobiililaitteilla missä tahansa rakennuksen sisällä. Liikkuvuuden lisäksi sairaalassa voisi olla kiinteitä näyttöjä tai tabletti-laitteita, joita voisi löytyä eri puolilta rakennusta. Toisena vaatimuksena sovelluksella oli toimia puheohjatusti, jotta sovelluksen käyttäminen olisi nopeaa ja vaivatonta. Puheohjauksen avulla sovelluksen käyttö ei vaatisi fyysistä kosketusta, joka lisäisi sairaalaympäristössä hygieenistä käyttöä ja sillä voitaisiin täten vähentää erilaisten bakteerien leviämistä.

Prototyyppi haluttiin toteuttaa selainpohjaisesti, jolloin se ei olisi sidottu tiettyyn laitteeseen, vaan toimisi millä tahansa laitteella, jossa on internetyhteys. Mobiililaitte, joka meiltä lähes jokaiselta löytyy taskusta, valittiin sopivaksi toteutuslaskelmaksi sovellukselle, koska eräs keskeinen käyttötapaus oli käyttäjä, joka navigoi sokkeloisessa sairaalaympäristössä ja helposti unohtaa reittiohjeet, jos ne löytyvät vain kiinteästi sisääntuloaulasta. Samaa sovellusprototyyppiä voisi käyttää siis niin kännykällä, tabletilla kuin tietokoneellakin, jolloin sovellus toimii samalla tavalla käyttöalustasta riippumatta. Selainpohjaisuus nähtiin myös saatavuuden ja käytettävyyden kannalta hyvänä ratkaisuna, koska näin käyttäjän olisi helppo saada sovellus käyttöön ilman erillistä ladattavaa sovelluksia. Sovellus olisi käytettävissä verkko-osoitteessa, joka olisi sairaalan aulassa helposti QR-koodina tai selkokielisenä luettavissa, jolloin se olisi kaikkien käytettävissä.

Prototyypin toteutuksessa ja testaamisessa kohdattiin haasteena se, että saatavilla ei ollut pohjapiirroksia sairaalasta, jossa sovellusta voitaisiin testata. Testaus nähtiin myös ongelmallisena sairaalaympäristössä, koska Jyväskylän uusi sairaala oli vielä prototyypin toteutuksen aikana rakennusvaiheessa. Täten prototyyppi päädyttiin toteuttamaan Jyväskylän Yliopiston Agora-rakennuksen tiloihin sopivaksi, jolloin toteutuksen kannalta tarvittavat resurssit olivat helposti saatavilla. Totesimme, että julkisena rakennuksena Agora voisi imitoida sairaalan ympäristöä riittävässä määrin prototyyppisen sovelluksen toteutuskehityksessä sekä prototyypin käytännöntestaaminen onnistuisi helposti.

Toteutettu prototyyppi esittää käyttäjälle haetun tilan tai henkilön sijainnin kartalla (kuvio 9). Prototyypin hakutoiminto toimii sekä puheohjatusti että kirjoittamalla. Prototyyppi toimii verkkoselaimen kautta ja se on käytettävissä niin tabletilla, puhelimella kuin tietokoneella. Prototyyppiä pääsee testaamaan osoitteesta <https://jyu-kompassi.herokuapp.com>. Teknologisista rajoitteista johtuen, prototyyppi toimii parhaiten Chrome-selaimessa, muiden selainten kanssa puheohjaus ei toimi.



KUVIO 9. Aulasovelluksen käyttöliittymänäkymä

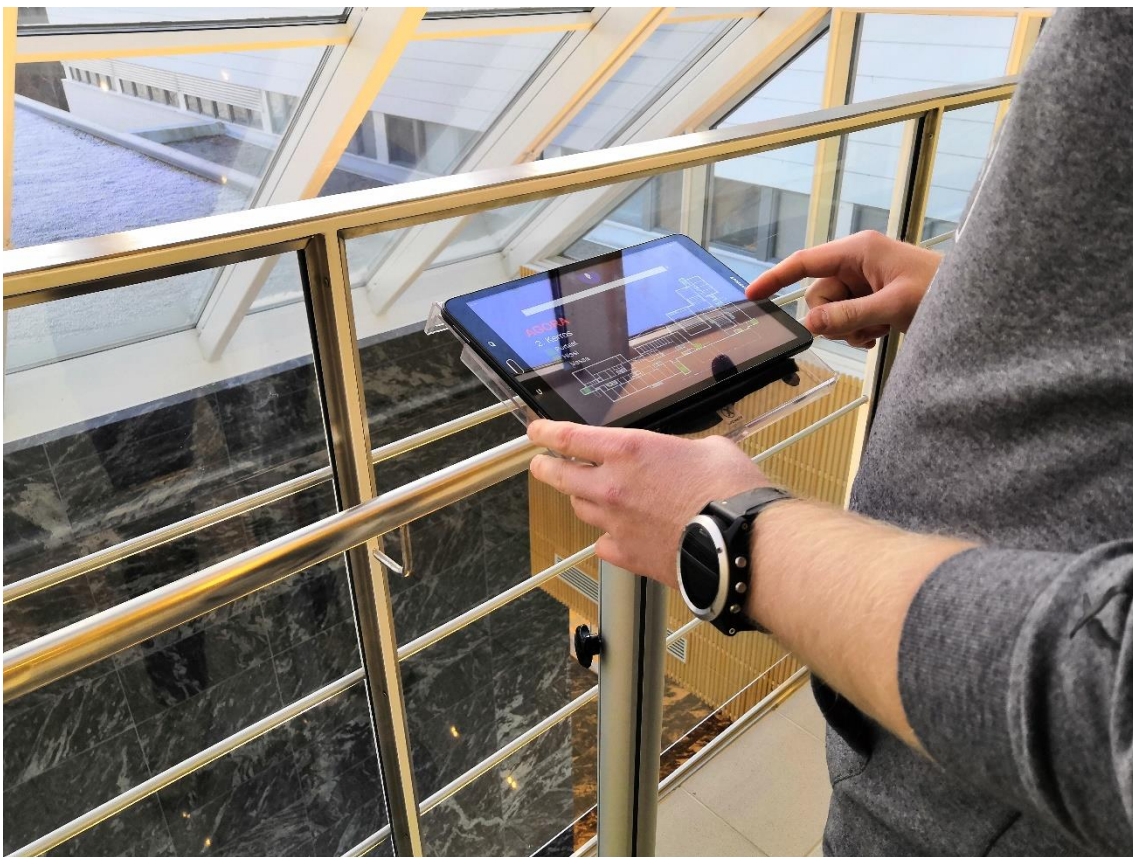
3.2.1 Aulasovellus -prototyypissä käytetyt teknologiat

Aulasovellus -prototyypin käyttöliittymä eli 'Frontend' on toteutettu JavaScript-ohjelmointikielellä, käyttäen jQuery-kirjastoa sekä Bootstrap 4:sta ulkoasun luomisessa. Prototyypin 'Backend' eli serveripuolen toiminnallisuus on kirjoitettu Node.js:llä, joka on alustariippumaton JavaScript ympäristö. Edellä mainittujen lisäksi toteutuksessa on käytetty Googlen omistaman Firebase-alustan tarjoamaa reaaliaikaista tietokantaa. Puheentunnistus on toteutettu Web Speech API:n avulla, joka on JavaScript API (Application programming interface), joka mahdollistaa puheen muuntamisen tekstiksi.

Prototyypin taustalla on käytetty Facebookin tarjoamaa Wit.ai tekoälyalustaa, jota on käytetty hyväksi hakutoiminnon toteutuksessa ja se mahdollistaa luonnollisen kielen tunnistamisen. Toiminnallisuuksiltaan Wit.ai on hyvin samankaltainen IBM:n Watson Assistantin kanssa. Alustojen erona on kuitenkin suomen kielen tuki, jonka Wit.Ai tarjoaa mutta Watson Assistant ei tue suomen kieltä. Eräs prototyypin tavoitteista olikin testata IBM:n kilpailevien tuotteiden suomen kielen tukea.

3.2.2 Aulasovellus prototyypin tulokset

Prototyypin valmistumisen jälkeen prototyyppi oli testikäytössä Jyväskylän Yliopiston Agora-rakennuksen aulaassa noin kuukauden, jonka aikana prototyypin käytöstä kerättiin anonymisti hakutuloksia ja palautetta (kuvio 10). Testiajan aikana yksittäisiä hakuja kertyi sovellukselle noin 600 kappaletta. Käyttäjälle annettiin mahdollisuus antaa palaute hakutuloksesta, oliko se onnistunut vai ei. Kerätyn testidatan avulla voitiin löytämään mahdolliset virheet tuloksissa sekä kartoittamaan käyttötapoja, joilla prototyyppiä voitaisiin kehittää sairaalaympäristöön.



KUVIO 10. Aulasovellus prototyyppi testikäytössä Agora-rakennuksessa

Prototyyppi käyttää Facebookin Wit.ai tekoälyalustaa, jota hyödynnettiin suomen kielen käsittelyssä. Wit.ai suoriutui pääsääntöisesti hyvin suomen kielen käsittelystä hyvin, joskin työmäärä Watson Assistenttiin verrattuna ei mitenkään vähentynyt, vaikka Wit.ai tarjoaakin suomen kielen tuen palveluunsa. Tekoälyteknologioissa on otettava huomioon se, että ne vaativat aina koulutusdataa, jotta ne toimisivat mahdollisimman hyvin. Prototyypin testausvaiheessa kerättiin juuri tätä arvokasta opetusdataa. Tekoälyn opettaminen vaatii aktiivista osallistumista tutkijoilta, jotta tekoälyä voidaan jatkokouluttaa kerätyllä datalla.

Erityisesti huomionarvoista on se, että tietyille sanoille on opetettava erilaisia taivutusmuotoja tarkkuuden lisäämiseksi, joka on hyvin työlästä. Mahdollisena ratkaisuna taivutusmuotojen tuomalle epätarkkuudelle nähtiin 'lemmatisoinnissa'. Se on tekstianalytiikasta tuttu tekniikka, jossa sanat muunnetaan perusmuotoon. Lemmatisoinnin avulla voitaisiin muuttaa käyttäjän syöte perusmuotoon ja vähentää teknologian opettamiseen vaativaa aikaa. Lemmatisoinnilla parannettaisiin myös teknologian tarkkuutta, koska erilaiset taivutusmuodot eivät tuottaisi enää ongelmia. Tätä tekniikkaa testattiin käytännössä, ja se nähtiin mahdollisesti toimivana ratkaisuna. Lemmatisointi-palvelua ajettiin itsetoteutetulla serverikoneella, jolle syötteet lähetettiin http-kutsuilla. Päädyttiin kuitenkin siihen, että tällä hetkellä palvelun saatavuutta ei voitu taata, joten sitä ei liitetty julkaistuun prototyyppiin.

3.2.3 Aulasovellus -prototyypin pohdinta ja jatkokehitys

Aulasovellus -prototyyppi oli mielestämme onnistunut, ja se tarjosi näkymän siihen, että vastaavaa tekniikkaa voitaisiin soveltaa toimimaan sairaalaympäristöön. Sovelluksen toteutus ja ylläpito olisi kuitenkin työlästä, sillä jokainen haettava huone ja henkilö on lisättävä erikseen tietokantoihin. Tämä tarkoittaa sitä, että henkilökunnan vaihtuessa sovellusta tulisi päivittää ja ylläpitää. Lisäksi jatkokehitysideana nähtiin navigointiominaisuus, joka voisi tarjota käyttäjälle reitin haluttuun paikkaan ja opasta hänet haluttuun kohteeseen.

Prototyyppi toimii tällä hetkellä ainoastaan Chrome-selaimella johtuen prototyypissä käytettävästä Web Speech API:sta, joka tarjoaa puheentunnistuksen vain Googlen Chrome -selaimelle. Prototyypin jatkokehityksessä olisi oleellista tutkia vaihtoehtoja puheentunnistukselle, joka tukisi myös muita selaimia kuten Firefox, Safari ja Microsoft Edge.

Kyseessä on mielenkiintoinen sovellus, joka oikein toteutettuna ja viimeisteltyinä voisi ratkaista suurissa ja sokkeloisissa tiloissa olevan ydinongelman, navigoinnin. Kuinka sairaalan asiakkaat, potilaat, vierailijat ja muut kävijät voisivat löytää haluamansa henkilön tai tilan helposti ja vaivattomasti. Sovelluksen avulla voitaisiin myös helpottaa vastaanottohenkilökunnan työtä ja parantaa sairaalassa asioivien kokemusta. Sovellus on jo nykyteknologioin toteutettavissa, vaikka parantamistakin aina on. Kehitystä kaippaa vielä ainakin luonnollisen kielen tunnistaminen ja erityisesti suomen kielen tunnistus joka on kovin lapsenkengissä, ja valtaosa tekoälyalustoista ei tarjokkaan tukea suomelle. Prototyyppiä kehitti yksi henkilö kolme kuukautta ja toinen henkilö yhden kuukauden.

3.4 Stressitesti

Stressi voi olla keskeinen osa meidän jokaisen elämää. Sitä tarvitaan erilaisten tavoitteiden saavuttamiseen ja sopivana määränä se edistää meidän suoriutumistamme ja on positiivinen voimavara. Liiallisena stressi on kuitenkin haitallista, jopa vaarallista. Syitä stressille voi olla erilaisia, kuten työ- tai kotiongelmia. (Matti, 2018.)

Virtuaalisen stressitestin tarkoitus oli testata ja luoda työkalua työnantajille, työterveydenhuollolle ja esimiehille, jolla voidaan seurata työntekijöiden stressitasoa erilaisilla ajanjaksoilla, esim. päivittäin tai viikoittain. Seuraamalla stressitason kehitystä voidaan osata tehdä ennaltaehkäisevästi tarvittavia toimenpiteitä työhyvinvoinnin ja työntekijöiden jaksamisen suhteen. Testi on testattavissa osoitteessa: <https://stressiapp.eu-gb.mybluemix.net/>

Sovelluksen vaatimusmäärittelyssä päädyttiin siihen, että testin tulee olla helposti käytettävissä ja sen täyttämiseen ei saa kulua aikaa montaa minuuttia. Päädyttiin käyttämään PSS (Perceived Stress Scale), joka on kansainvälisesti käytössä oleva kymmenkohtainen kysely, jossa käyttäjä arvioi kokemaansa stressiä asteikolla 1 - 5.

3.4.1 Stressitesti prototyypissä käytetyt teknologiat

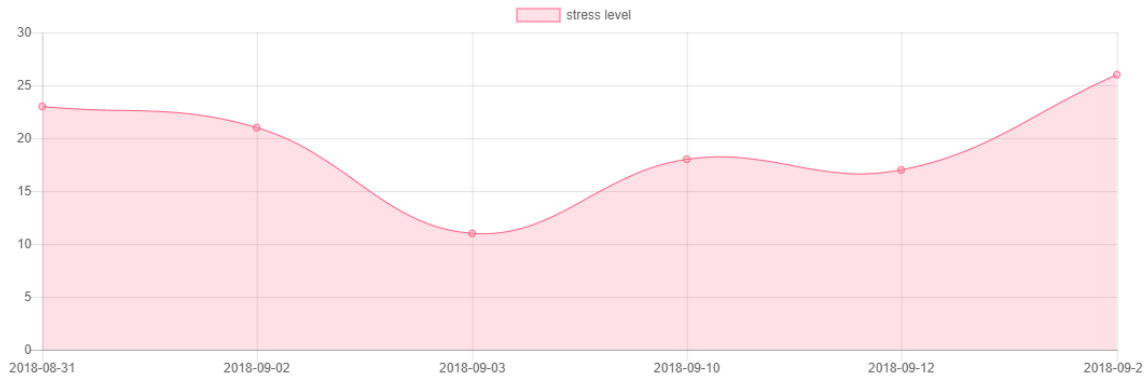
Stressitesti toteutettiin käyttäen Node.js alustaa ja käyttäen JavaScriptin React.js kehystä. Kehitysvaiheessa pohdittiin pitkään mahdollisuutta puhumalla täytettävään testiin, mutta aikaisempien kokemusten valossa se nähtiin vastaajan kannalta kuormittavana tekijänä, joka pidentää ja vaikeuttaa testin tekoa. Joten päädyimme luopumaan puheohjatuista testistä ja toteuttaa se perinteisellä mallilla, joka vastasi paremmin vaatimusmäärittelyä.

Vastaa kyselyyn:

1. Viimeisen kuukauden aikana, kuinka usein olet suuttunut jonkin odottamattoman tapahtuman vuoksi?
 ei ollenkaan hyvin usein
2. Viimeisen kuukauden aikana, kuinka usein sinusta on tuntunut, että et voi hallita elämäsi tärkeitä asioita?
 ei ollenkaan hyvin usein
3. Viimeisen kuukauden aikana, kuinka usein sinusta on tuntunut hermostuneelta tai stressaantuneelta?
 ei ollenkaan hyvin usein
4. Viimeisen kuukauden aikana, kuinka usein sinulla on ollut luottavainen olo siitä, että kykenet hallitsemaan henkilökohtaiset ongelmasi?
 ei ollenkaan hyvin usein
5. Viimeisen kuukauden aikana, kuinka usein sinusta on tuntunut, että asiat sujuvat hyvin?
 ei ollenkaan hyvin usein

KUVIO 11. Stressitesti prototyypin käyttöliittymä

Tekoälyllisiä ominaisuuksia pohdittaessa, ajatuksena oli toteuttaa laajempi ohjausnäkyvä jossa työnantaja tai työterveyslaitos (tai muu taho) voisi seurata työntekijöiden hyvinvointia. Ja saatavilla olisi myös mahdollisesti muita työhyvinvoinnin ja –seurannan mittareita. Päädettiin kuitenkin tässä prototyypissä jättämään pois laajemmat näkymät ja luotiin seurantanäkyvä, jossa voidaan seurata yksittäisten työyksiköiden työntekijöiden stressikeskiarvoa päivätasolla ja sen kehitystä ajan funktiona (KUVIO 12). Keskiarvot tallennetaan IBM:n Compose for PostgreSQL – tietokantaan ja niiden haku ohjausnäkyvässä tapahtuu SQL-haulla.



KUVIO 12. Stressitesti prototyypin tulostusnäky

3.4.2 Stressitesti -prototyypin tulokset

Stressitestin käyttö ja tulokset olivat hyvin suoraviivaisia sovelluksen yksinkertaisuuden vuoksi. Sovellus ei päässyt laajempaan testikäyttöön, mutta sai muutamia kehittäjäryhmän ulkopuolisia testauskäyttöjä Jyväskylän Yliopiston järjestämässä Tutkijoiden Yö -tapahtumassa, jossa sitä kävi testaamassa noin 30 vierailijaa. Reaktiot sovellukseen olivat positiivisia ja se nähtiin mielekkäänä käyttää ja mahdollisesti mielenkiintoisena mittarina arkeen.

Prototyypin kehityksen alkuvaiheessa haluttiin integroida sovellukseen IBM:n tarjoama Cognos Analytics dashboard. Kuitenkin totesimme itse, sekä IBM:n konsultoinnin tuloksena myös todettiin, että kyseinen palvelu on kohtalaisen monimutkainen soveltaa kyseisen prototyypin kontekstiin ja vastaa paremmin suurten organisaatioiden datan visualisointi- ja analytiikkatarpeisiin.

3.4.3 Stressitesti -prototyypin pohdinta ja jatkokehitys

Prototyyppi nähtiin onnistuneena teknologiademona, jolla voitiin näyttää, että on hyvinkin helppo toteuttaa yksinkertaisia kyselylomakkeita sähköisesti ja tuottaa tärkeää informaatiota työterveydenhuollolle ja työntekijöille hänen hyvinvoinnistaan. Mielenkiintoisena havaintona tehtiin Stressitesti-protota kehittäessä, että useinkaan yksinkertaisten ja helppojen sovellusten toteuttamiseen ei välttämättä kannata yrittää sisältää tekoälyä, koska se ei tuo mitään lisäarvoa sovellukselle tai sen käyttäjille.

Stressitestin tulostusnäky laajentaminen kattamaan erilaisia mielenkiintoisia tuloksia, esimerkiksi yksilökohtaisempia tuloksia ja ylitason organisaation tuloksia voidaan jatkossa kehittää, jos ne nähdään oleellisina ja mielenkiintoisina lukuina. Integrointi osaksi isompaa hyvinvoinnin arvioinnissa käytettävää myös 'työkalupakkia' olisi

mielenkiintoinen. Prototyyppi kehitettiin erittäin nopealla aikataululla kolmessa viikossa kolmen henkilön voimin.

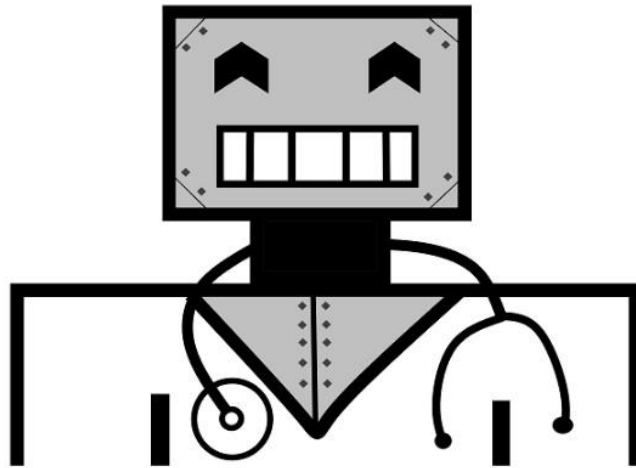
3.5 Virtuaalilääkäri

Lääkärin vastaanotolle mentäessä ongelmana on yleensä rajoitettu pääsy kasvotusten tapahtuviin vastaanottoaikoihin ja hintavat konsultoinnit rutiinitapauksissa. Prototyypin tarkoituksena oli toteuttaa virtuaalinen hoitoratkaisu, joka mahdollistaisi potilaalle lääketieteellisen neuvonnan 24/7 ja täten nopeuttaen diagnosointia. Prototyypin tarve oli ideoitu hankeen ensimmäisessä vaiheessa, jossa järjestettiin työpajoja sairaanhoitopiirin henkilökunnan kesken. Prototyypiltä odotettiin ominaisuuksia kuten:

- Keskustelullista oireiden keruuta, seulontaa varten
- Edistynyttä oireiden kyselyä ja etukäteisdiagnoosia rajoitetulle määrälle sairauksia
- Verkko- tai videokeskustelua lääkärin kanssa
- Kuvan lähettäminen lääkärille diagnoosia helpottamaan
- Pääsy omiin yksityisiin henkilötietoihin
- Lähetettyjen kuvien automatisoitu etukäteisdiagnosointi rajoitetulle määrälle sairauksia

Ominaisuuksien lisäksi prototyyppi haluttiin toteuttaa selainpohjaisesti, jotta sovellus toimisi kaikilla mahdollisilla laitteilla eikä sovellusta tarvitsisi erikseen ladata käytettävälle laitteelle. Prototyypille asetetut tavoitteet saatiin toteutettua, lähetettyjen kuvien automatisoitua diagnosointia lukuun ottamatta. Kuvien diagnosoinnin ominaisuuteen paneuduttiin ja se todettiin täysin mahdolliseksi toteuttaa, mutta sen toteuttamista varten oltaisiin tarvittu suuri määrä opetusdataa ja aikaa teknologian opettamista varten.

Prototyyppiä pääsee testaamaan osoitteessa <https://robodoc.herokuapp.com>, jossa se on saatavilla (kuvio 13). Prototyyppiä käytettäessä käyttäjän täytyy ensin kirjautua tai rekisteröityä sisään. Tämän jälkeen käyttäjä voi luoda uuden oirekuvauksen, jonka pohjalta tekoälyteknologia seuloo tekstistä oireet. Rakensimme prototyypin ohella myös työkalun, joka toteuttaa diagnosoinnin syötetyistä oireista. Diagnosointi näkyy vain sovellusta käyttäville lääkäreille, eikä alustavat diagnoosit näy potilaille.



Mikä on RoboDoc?

RoboDoc on apuväline lääkärin ja potilaan väliseen vuorovaikutukseen. RoboDoc pitää kirjaa potilaan oireista ja auttaa lääkäriä päätöksenteossa.

KUVIO 13. Virtuaalilääkäri prototyypin aloitusnäky

3.5.1 Virtuaalilääkäri -prototyypissä käytetyt teknologiat

Virtuaalilääkäri -prototyypin käyttöliittymä on toteutettu Nuxt.js ohjelmointikielellä, joka on JavaScriptiin perustuva framework. Nuxt.js optimoi itsestään koodin parhaaksi mahdolliseksi, joka mahdollistaa laadukkaiden selainpohjaisten sovellusten luomisen. Prototyypin tekoälypuolella on käytetty Watson Assistantia, joka seulo oirekuvauksesta käyttäjän kuvailemat oireet. Oireet puolestaan syötettiin diagnosointityökalulle, joka antaa listauksen mahdollisista diagnooseista oireiden perusteella (kuvio 14). Diagnosointityökalu ohjelmointiin Python ohjelmointikielellä. Diagnosointityökalun toiminta perustuu verkosta löytyvään lääketieteelliseen dataan. Sovelluksen tietokantana käytettiin aiemmista prototyypeistä tutuksi tullutta Firebase-alustan tarjoamaa reaaliaikaista tietokantapalvelua.

RoboDoc
Potilaat Kirjautu ulos

05.09.2018

Potilas: 9g2p7LjCDNkqh4HijN29bKzPFv1
Muokattu viimeksi: 19.10.2018

Oirekuvaus


nuha, kuume, yskä, ruokahaluttomuus.

Tunnistettut oireet

nuha, kuume, yskä, ruokahaluttomuus

Mahdolliset diagnoosit

Nuhakuume, flunssa
Flunssa lapsella
Kurkkupaise (nielupaise)
Sivuontelotulehdus (poskiontelotulehdus) lapsella
Korvatulehdus lapsella



KUVIO 14. Virtuaalilääkäri -prototyypin diagnoosityökalun toiminta

Näiden lisäksi yritimme käyttää Watson Visual Recognition palvelua kuvien automatisoitua diagnosointia varten, mutta meillä ei ollut tarpeeksi opetusdataa ominaisuuden toteuttamista varten. Totesimme kuitenkin, että sen toteuttaminen ja käyttöönotto toteutettuun prototyyppiin olisi mahdollista.

3.5.2 Virtuaalilääkäri -prototyypin tulokset

Watson Assistant suoriutui hyvin oireiden seulonnassa, ottaen huomioon sen, ettei Watson Assistant tue suomen kieltä. Watson Assistantin opettamiseen käytettiin paljon aikaa, koska oireiden tunnistamisen opettamisvaiheessa otettiin huomioon suomen kielelle ominaisten taivutusmuotojen opettaminen. Jokaisen taivutusmuodon opettaminen on kuitenkin työlästä ja aikaa vievää työtä.

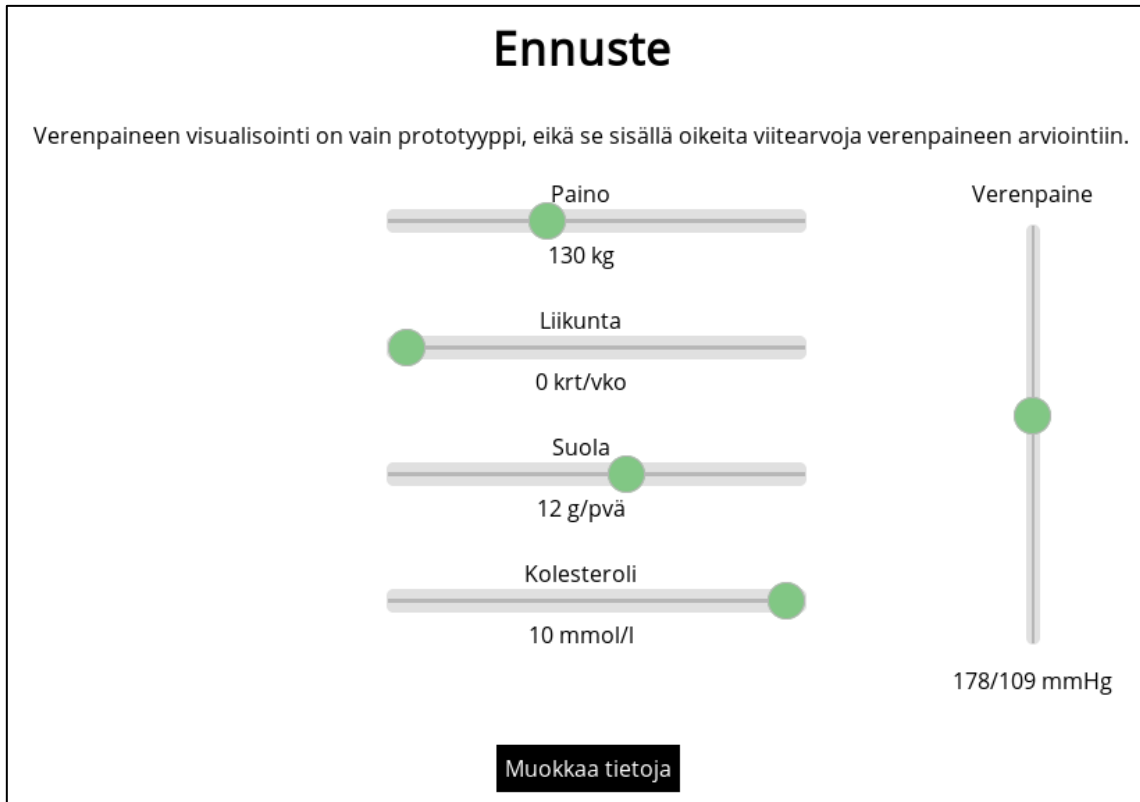
Diagnosointityökalu toimi tiettyjen oirekuvauksien kanssa hyvin, esimerkiksi kuvatessa flunssan oireita diagnosointityökalu ehdotti ensimmäisenä diagnoosiksi flunssaa (kuvio 14). Diagnosointityökalu ei kuitenkaan ollut täydellinen ja osassa oirekuvauksissa se antoikin joko yleispäteviä tai vääriä diagnooseja. Diagnosointityökalu perustui yleispätevään lääketieteelliseen dataan ja se olisi voinut toimia paremmin, jos siihen käytetty opetusdata olisi ollut paremmin tarkennettu käyttötarkoitukseen nähden. Näiden asioiden lisäksi olimme tyytyväisiä prototyypin käyttöliittymään ja toimivuuteen (KUVIO 15).



KUVIO 15. Virtuaalilääkäri prototyypin potilasnäkyvä

3.5.3 Virtuaalilääkäri -prototyypin pohdinta ja jatkokehitys

Virtuaalilääkäriin lisättiin elintapojen vaikutusta verenpaineeseen kuvaava käyttöliittymä, joka kysyy käyttäjältä tämän painon, viikoittaisen liikuntamäärän, päivittäisen suolansaannin, kolesterolin ja nykyisen verenpaineen. Sen jälkeen Virtuaalilääkäri siirtyy näkymään, jossa käyttäjä pääsee liukureiden avulla muuttamaan aikaisemmassa näkymässä antamia arvoja. Sivun oikeassa laidassa näkyvä verenpainemittari liikkuu muutosten tahdissa reaaliaikaisesti ja näyttää, kuinka muutettu arvo vaikuttaa verenpaineeseen. Esimerkiksi painon nousu 90 kilogrammasta 130 kilogrammaan nosti verenpaineen alkulukemista 170/100 (mmHg) loppulukemiin 178/109 (mmHg). Elintapojen vaikutusta arvioivaa komponenttia ja virtuaalilääkäriä voidaan myös hyödyntää potilaan hoidon suunnittelussa ja arvioinnissa.



KUVIO 16. Verenpaineen visualisointi

Mahdollista jatkokehitystä ideoitiin yhdessä KSSH:n kehittäjäylilääkäri, sisätautien ja kardiologian erikoislääkäri Pirjo Mustosen kanssa. Mustonen näki mahdollisuuden sovelluksen toimivuudesta esimerkiksi sairaalan päivystyksessä, jolloin voitaisiin saada potilaalta ennakkotietoa hänen käynnin syystä. Prototyypin ensimmäinen versio valmistui noin kahdessa kuukaudessa, mutta sitä on sen jälkeen vielä paranneltu ja lisätty erilaisia ominaisuuksia. Prototyyppiä on työstänyt yhteensä neljä henkilöä.

Prototyypin raportointivaiheessa saimme myös tietoomme, että Klinik on toteuttanut hiljattain samantyyllisen tuotteen kuin meidän tuottama prototyyppi <https://klinik.fi/tunnista-vaivasi>. Konsepti on siis todettu toimivaksi ja tarpeelliseksi myös muiden toimesta.

4 Yhteenveto

Tässä raportissa käsiteltävät prototyypit ovat keskittyneet enimmäkseen siihen, miten hyvin tekoälyllä voidaan käsitellä suomenkielistä luonnollista kieltä. Lopputuloksena on se, että käyttämämme teknologiat pystyvät käsittelemään suomen kieltä, mutta niiden vaatima opetusdata täytyy olla kattava ja kattavan opetusdatan toteuttaminen on todella työlästä.

Opetusdataa voidaan vähentää lemmatisoimalla teknologialle lähetettävä syöte eli muuttamalla se perusmuotoon. Tällöin teknologioille ei tarvitse opettaa kaikkia tunnistettavan sanan taivutusmuotoja, jolloin myös opetusdata pienenee. Teknologiat toimivatkin englannin kielellä huomattavasti paremmin, koska englannin kielessä sanojen taivutusmuodot eivät ole samalla tavalla itse sanassa kuin suomen kielessä. Suomen kielen tunnistaminen on siis haastavaa tekoälylle, eikä käyttämillämme teknologioilla ole vielä hyvää suomen kielen tukea. Suomen kielen tuen toteuttaminen on haastavaa ja työlästä teknologioille, mutta ei kuitenkaan mahdotonta.

Tässä raportissa käsitellyt teknologiat liittyvät oleellisesti muihin nykyaikaisiin läpimurtoteknologioihin. Tällaisista teknologioista ensimmäisenä mainittakoon 5G ja IoT, eli internet-of-things, jotka linkittävät yhä useammat elämäämme koskettavat ympäristöt tietoverkkoihin ja sallivat entistä laajemman tietomassan keräämisen ja hyödyntämisen, sekä verkon yli tapahtuvan ympäristöön vaikuttamisen. Esimerkiksi sairaala- ja kotihoidossa tämä tarkoittaisi entistä laajempaa sensorijärjestelmää ja niistä saadun tiedon yhdistämistä.

Toisena oleellisesti tekoälyn soveltamiseen liittyvänä teknologiana mainittakoon ohjelmisto- ja palvelurobotiikka, joka mahdollistaa esimerkiksi prosessien automatisointia ja teknologioiden käytettävyyden parantamista. Robotiikan ja tekoälyratkaisuiden yhteisvaikuttavuudesta esimerkkinä annettakoon esitellyn aulasovellus –prototyypin potentiaalinen integroiminen palvelurobotialustaan, jossa robotiikka-alustan ja tekoälyteknologian kyvykkyydet täydentäisivät toisiaan.

Lisäksi esiteltyjen teknologioiden yhdistäminen muihin tekoälypohjaisiin ratkaisuihin tulee olemaan palvelukehityksen kulmakivi. Esimerkiksi strategisella tasolla tämä voi tarkoittaa esitellyn stressitesti –prototyypisovelluksen kaltaisen ratkaisun, sekä strategisen tason päätöksentekojärjestelmän yhdistämistä järjestelmään, jonka avulla organisaatiossa voidaan luoda ennustavia toimintasuosituksia.

Raportoitujen teknologioiden luomien uusien mahdollisuuksien sekä automaation luomien hyötyjen käyttöönotto terveydenhuollon prosesseissa vaatii vielä pohjatyötä, sekä teknologioiden tuotteistamista. Kehitystyöhön liittyy myös suurempi kuva terveydenhuollon rakenteellisesta muutoksesta ja digitaalisten palvelualueiden kehityksestä. Prototyypisovelluksina toteutetut ratkaisut ovatkin suunniteltu siten, että ne soveltuvat osaksi suurempaa läpimurtoteknologioiden käyttöönottoa tukevaa palvelualuekehitystä, jossa kansalaisille pyritään tarjoamaan optimoituja ja räätälöityjä sosiaali- ja terveydenhuollon palveluita.

LÄHTEET

Mattila, A. 2018. Stressi. Lääkärikirja Duodecim. Saatavilla: 16.11.2018
https://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=dlk00976

Watson Assistant. 2018a. Dokumentaatio palvelun käyttöön. Saatavilla: 19.11.2018
<https://console.bluemix.net/apidocs/assistant>

Watson Assistant. 2018b. Palvelun tukemat kielet. Saatavilla: 19.11.2018
<https://console.bluemix.net/docs/services/assistant/lang-support.html#supported-languages>

Watson Assistant. 2018c. Palvelun hinnoittelu. Saatavilla: 19.11.2018
https://console.bluemix.net/catalog/services/conversation?hideTours=true&?cm_sp=WatsonPlatform-WatsonPlatform--OnPageNavCTA-IBMWatson_Conversation--Watson_Developer_Website

Informaatioteknologian tiedekunnan julkaisuja
No. 68/2018

ISBN 978-951-39-7652-1 (verkkokj.)
ISSN 2323-5004