OULUN AMMATTIKORKEAKOULU

Manda Kosola

PUHEOHJAUS WEB-SOVELLUKSISSA

PUHEOHJAUS WEB-SOVELLUKSISSA

Manda Kosola
Opinnäytetyö
Kevät 2020
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

---

Tekijä: Manda Kosola
Opinnäytetyön nimi: Puheohjaus web-sovelluksissa
Työn ohjaaja: Anne Keskitalo
Työn valmistumislukukausi ja -vuosi: Kevät 2020 Sivumäärä: 38 + 2 liitettä

Tämän opinnäytetyön tarkoituksena oli perehtyä puheentunnistuksen teknologiaan, sekä selvittää, miten web-sovelluksia voidaan ohjata puheella. Työn taustalla oli oma kiinnostus web-ohjelmointiin ja puheohjattaviin käyttöliittymiin. Tavoitteena oli myös tutkia, mihin puheohjattavien sovellusten tekniikka pohjautuu sekä mikä on tämän teknologian nykytilanne ja tulevaisuuden näkymät.

Työ koostuu sekä teoria- että sovellusosuudesta. Työssä käydään ensin läpi yleisesti puheentunnistusta ja tutkitaan sitten tarkemmin puheohjauksen soveltuvuutta web-sovelluksiin. Teoriaosuuteen hyödynnettiin useita eri lähteitä mm. kirjoista, verkkosivuista ja podcasteista. Opitut asiat otettiin käytännön kokeiluun oman puheohjauksella toimivan web-sovelluksen suunnitteluun ja kehittämiseen. Sovelluksessa kiinnitettiin erityisesti huomiota puheella toimivan käyttöliittymän suunnitteluun.

Työn perusteella voitiin todeta, että puheentunnistus on hyvin mielenkiintoinen teknologian ala, joka tulee todennäköisesti yleistymään enemmän lähitulevaisuudessa. Kehityksen myötä voidaan olettaa, että puheentunnistusta tullaan käyttämään enenevissä määrin myös web-sovelluksissa. Tällä hetkellä puheohjauksen teknologiaan liittyy joitakin ongelmia, joita tässä työssäkin käsitellään.

Tulevaisuudessa puheohjaus saattaa mullistaa käyttöliittymäajattelua, sillä se voi parantaa teknisten laitteiden kanssa työskentelevien ihmisten työergonomiaa, sekä nopeuttaa tekstin syöttämistä ja tietokoneelle komentojen antamista.

Asiasanat: puheentunnistussovellus, äänentunnistus, puheentunnistus, puheteknologia, web-sovellukset, web-ohjelmointi

ABSTRACT

Oulu University of Applied Sciences
Degree Programme of Information Technology, Software Development

---

Author: Manda Kosola
Title of thesis: Speech control in web applications
Supervisor: Anne Keskitalo
Term and year when the thesis was submitted: Spring 2020 Number of pages: 38 + 2

The purpose of this thesis was to investigate the technology of speech recognition, and to find out how voice control can be utilized in web applications. The work was based on own interest in web development and voice-controlled interfaces. The aim was also to study what is the technology behind voice-controlled applications, as well as the current situation and future prospects of this technology.

The work starts with general knowledge of speech recognition and voice-controlled applications, and then focuses on a more detailed study of the suitability of voice control for web applications. The thesis consists of both theoretical and practical parts. Several different sources have been utilized for the theoretical part, including books, websites and podcasts. Practical part of work was developing voice-controlled web application for testing purposes.

Based on the work, it was concluded that speech recognition is likely to become more common in the near future. As it is currently being developed at a good pace, it can be assumed that speech recognition will soon be increasingly used in web applications and websites as well. Currently, there are some problems with voice control techniques that were also addressed in this work.

In the future, voice control may revolutionize user interfaces, improving the ergonomics of people working with technical devices, as well as speeding up text input and giving commands to a computer.

Keywords: Voice Recognition Software, Speech Recognition, Voice Recognition, Web Apps, Web Development

SISÄLLYS

SANASTO

API (Application Programming Interface) – Ohjelmointirajapinta, jonka avulla eri ohjelmat voivat vaihtaa tietoja keskenään ennalta määriteltyjen menetelmien mukaisesti.

CSS (Cascading Style Sheets) – Tyyliohjeiden laji ulkoasun luomiseen web-sivuille ja -sovelluksille.

Foneemi – Puhutun kielen pienin merkityksiä toisistaan erottava äänneyksikkö.

HTML (Hypertext Markup Language) – Kuvauskieli web-sivujen ja -sovellusten rakentamiseen.

JavaScript – Web-ympäristöön soveltuva ohjelmointikieli.

JSON (JavaScript Object Notation) – Avoimen standardin tiedostomuoto tiedon tallentamiseen.

Natiivisovellus – Sovellusalustan omalla kehitysympäristöllä toteutettu sovellus.

Ohjelmistokehys – Ohjelmoinnin apuväline, joka muodostaa rungon rakennettavalle ohjelmalle.

SPA (Single Page App) – Yhden sivun web-sovellus.

STT (Speech-To-Text) – Puheen kääntäminen tekstiksi automaattisen puheentunnistuksen avulla.

TLS (Transport Security Layer) – Salausprotokolla verkkosivujen ja web-sovellusten suojaamiseen. Aiemmin tunnettu nimellä Secure Sockets Layer (SSL).

TTS (Text-To-Speech) – Tekstin muuttaminen synteettiseksi puheeksi.

Vue.js – JavaScript ohjelmistokehys.

Web-sovellus – Verkkoselaimessa toimiva sovellus.

1 JOHDANTO

Tässä työssä perehdyttiin puheohjauksen ja puheentunnistuksen teknologioihin sekä tutkittiin, miten niitä voidaan nyt ja tulevaisuudessa hyödyntää web-sovelluksissa. Puheentunnistusta kokeiltiin myös käytännössä suunnittelemalla ja rakentamalla puheohjattava web-sovellus.

Puheohjaus ja **puheentunnistus** sekoittuvat joskus termeinä, ja osittain ne tarkoittavatkin samaa asiaa. Tässä työssä puheohjauksella tarkoitetaan kokonaisen puheohjattavan sovelluksen toimintaa ja puheentunnistuksella teknologiaa, joka mahdollistaa puheen muuttamisen tekstiksi.

Tietotekniikan kehityksessä huomattiin jo hyvin varhain, että komentojen kirjoittamisen sijaan ihmisen olisi mahdollista ohjata tietokonetta puheella. Tekniikan Maailma kuvaili jo vuonna 1984 ilmestyneessä numerossaan sen ajan puheen tunnistamisen mahdollisuuksia ja hyötyjä. Artikkelissa ennustettiin, että tällainen ihmisen ja koneen välinen puheella käytävä keskustelu olisi tulossa yleiseen käyttöön aikaisintaan seuraavalla vuosikymmenellä eli 90-luvulla. (1, s. 58.)

90-luvulla ei kuitenkaan vielä saavutettu kovin suuria mullistuksia puheentunnistuksessa. Vasta 2010-luvun alkupuolella julkaistut virtuaaliavustajat, kuten Applen Siri ja Amazonin Alexa, toivat puheentunnistuksen tavallisten kuluttajien käyttöön älypuhelimiin ja muihin älylaitteisiin. Puheen tarkka ja reaaliaikainen prosessointi vaatii huomattavia laskentatehoja ja resursseja, mikä on osittain syy hitaalle kehitykselle. Nykypäivänä se on mahdollista kone- ja syväoppimisen, tekoälyn sekä suurten datamäärien myötä. Puheentunnistuksen teknologiaan liittyy silti edelleen haasteita. Erityisesti suomen kieli on vaikea opetettava koneille, sillä sanoihin sisältyy valtavasti muunteluita, erilaisia taivutuksia ja pitkiä yhdyssanoja. (2.)

Ellei tulevaisuudessa keksitä vielä parempia käyttöliittymiä, tulee puheentunnistus varmasti kehittymään pitkälle, jolloin sen rooli jokapäiväisessä elämässämme voi olla merkittäväkin. Useissa lähteissä ennustetaan puheentunnistusteknologian nopeaa yleistymistä ja kehittymistä, joten jo lähitulevaisuudessa se voi olla yhä arkipäiväisempi toiminnallisuus sekä välttämätön työkalu joillekin yrityksille. Kaikkia puheentunnistuksen mahdollisuuksia on toistaiseksi mahdotonta edes arvata.

Jos ennusteet puheentunnistuksen tulevaisuudesta käyvät toteen, olisi tärkeää alkaa ottamaan sen potentiaali huomioon kaikessa kehityksessä. Huomionarvoista nykypäivänä on se, että melkein

joka kolmas maapallon ihmisistä kantaa mukanaan älypuhelinta (3). Tämän myötä myös internetiä käytetään enenevissä määrin puhelimen pieneltä ruudulta, ja vuonna 2018 tehdyn tutkimuksen mukaan verkkosivustojen liikenteestä jo yli puolet kävijöistä käytti älypuhelinta (4). Pienen ruudun selaaminen ja etenkin sillä kirjoittaminen ei ole ihmiselle kovin luontainen, ergonominen tai nopea tapa toimia. Moni verkkoselaimessa toimiva sovellus ja sivusto voisi hyötyä uudenlaisesta käyttöliittymäajattelusta, jonka puheentunnistuksen teknologia mahdollistaa.

2 PUHEOHJAUKSEN TEKNOLOGIA

Puheohjauksen teknologia voidaan jakaa karkeasti kolmeen tehtävään: automaattinen puheen tunnistus, aikeen tunnistaminen ja näihin reagoiminen (5).

Ensimmäisen vaiheen **automaattinen puheentunnistus** (engl. Automatic Speech Recognition) on teknologia, joka kääntää äänisignaaleina havaitun puheen sitä parhaiten vastaavaksi tekstiksi. Yleensä puhutaan pelkästä puheentunnistuksesta. Tunnistusmenetelmä perustuu puheen äänteiden ominaispiirteisiin, joita verrataan suurista puheaineistoista saatuihin tilastollisiin malleihin. (6, s. 336.)

Toisessa vaiheessa tekstiksi muutetusta puheesta pyritään **tunnistamaan aikeet** (engl. Intent Recognition) eli se mitä käyttäjä puheellaan tarkoittaa (5). Aikeiden tunnistamiseenkin voidaan hyödyntää suuria tekstiaineistoja ja koneoppimista. Käytännössä kaikissa puheohjattavissa sovelluksissa on vielä kolmaskin vaihe, jossa ohjelma **reagoi puheeseen tai aikeeseen** tarvittavalla tavalla. Tässä käytettävät algoritmit riippuvat sovelluksen tarkoituksesta. Usein puheohjattavissa sovelluksissa on loogista vastata käyttäjälle puheella puheeseen puhesyntetisaattorin avulla, mutta vastaus voidaan näyttää myös visuaalisessa käyttöliittymässä.

2.1 Puhe ja kieli

Puhe on ihmisten välinen monipuolinen **kommunikaation signaali**, johon sisältyy paljon informaatiota, kuten kirjoitetun kielen sääntöjä, tunteita ja eri painotuksia (7). Fyysisiltä ominaisuuksiltaan puhe on hienovaraisesti säädeltyä ilmanvirtausta. Niin sanottu **puheketju** alkaa keskushermostosta, kun puhujan mieleen tulee ajatus, jonka hän haluaa ilmaista ääneen. Täysin tiedostamattomasti puhuja suunnittelee viestinsä kielellisen sisällön ja välittää sitten keskushermostosta toimintakäskyt puhe-elimistön lihaksiin. Puhe-elimistö alkaa liikehtiä ja saa ilman virtaamaan. Puhe-elinten liikkeitä säätelemällä ilmanvirtaus muuttuu nopeiksi ilmanpaineen vaihteluiksi eli akustiseksi energiaksi. Puheketjun akustinen vaihe kantautuu vastaanottajan korviin ja on nykyään helposti tallennettavissa myös teknisillä laitteilla. (8, s. 15–17.)

Ymmärrettävän puheen tuottamiseksi ja vastaanottamiseksi tarvitaan lisäksi tietoa puhutusta kielestä. Puhuttu kieli on kuin koodausta, sillä se perustuu ennalta tiedettyihin äänteiden ja merkitysten välisiin suhteisiin, joista muodostetaan merkityksellisiä sanoja ja lauseita. Yhteisten kielen sääntöjen avulla ajatusten merkitykset muutetaan sanoiksi, jotka puheen vastaanottava ihminen dekoodaa eli purkaa takaisin merkityksiksi. (8, s. 15.)

2.2 Puheentunnistus

Puheentunnistus on siitä mielenkiintoinen teknologian ala, että sen kehityksessä pyritään usein ottamaan mallia ihmisen kyvystä kuulla ja ymmärtää. Tiedämme, että on olemassa biologinen mekanismi, joka kykenee tuottamaan, tunnistamaan ja ymmärtämään puhetta, mutta samankaltaisen järjestelmän toteuttaminen teknisesti ei ole aivan helppoa. Teknologiaan liittyy matemaattisia ongelmia, sillä kaikkia äänen ja kielellisen viestin yhteyksiin vaikuttavia tekijöitä ei vielä tunneta. (9, s. 73–75.)

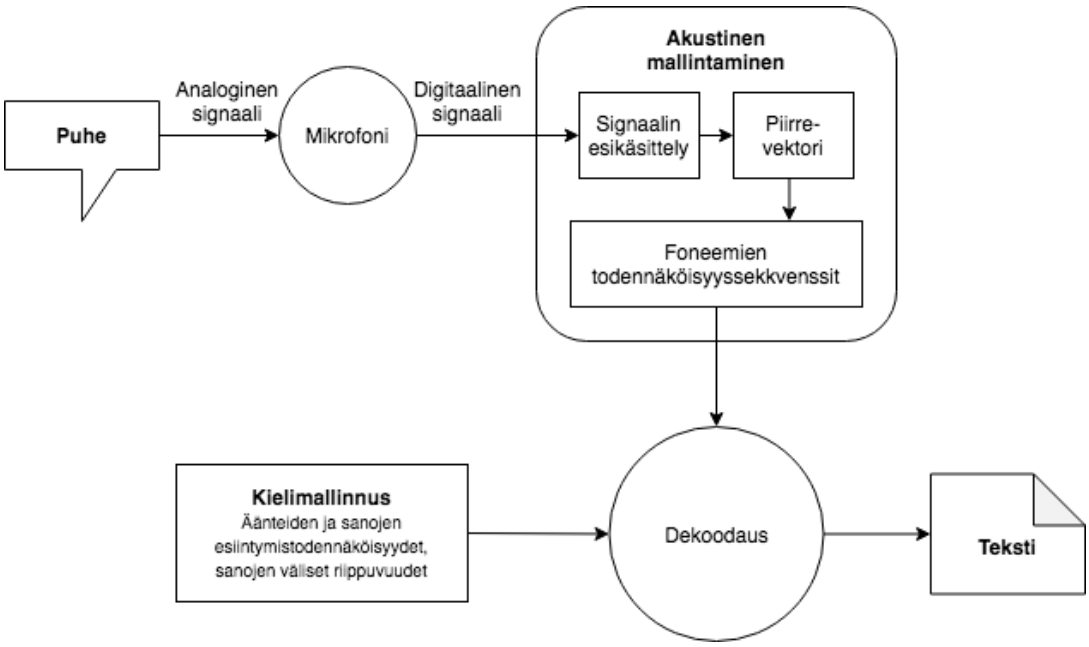
Ihmiselle puheen kuuleminen ja ymmärtäminen on niin luontaista ja hyvin mukautunutta, että usein koneelliselle puheentunnistuksellekin asetetaan kovat vaatimukset. Virheettömien ja puhtaiden äänteiden tunnistaminen ja käsittely onkin koneelle selkeää, mutta kun yhtälöön lisätään oikean ympäristön kaikki muuttujat, prosessi muuttuu vaikeaksi. Normaalin puheen lisäksi ympäristöstä kuuluu jatkuvasti taustahälyä ja ylimääräisiä ääniä, joten tietokoneen täytyisi pystyä erottamaan, mikä havaitusta äänestä on puhetta. Lisäksi äänisignaalien virroista pitäisi pystyä erottamaan kirjaimet, sanat ja lauseet, sekä sivuuttaa puheentunnistuksen kannalta merkityksettömät äännähdykset, kuten yskäisyt ja epäröinnit. Tietokoneen olisi myös pystyttävä ymmärtämään inhimilliset virheet puheessa, kuten yhteen puhutut tai väärin lausutut sanat. Kun tähän lisätään vielä kaikki maailman eri kielet ja niiden murteet, joita on joidenkin arvioiden mukaan jopa yhdeksäntuhatta (10), on tiedossa hyvin haasteellinen tehtävä. (9, s. 73–75.)

2.2.1 Puheen tunnistaminen ja muuttaminen tekstiksi

Puheentunnistusta käyttävän laitteen on ensin havaittava äänisignaalit mikrofonin avulla. Äänisignaalit havaitaan ilmanpaineen vaihteluina eli **analogisina signaaleina**. Mikrofoni muuttaa analogisen äänisignaalin **sähköisiksi signaaleiksi**, jonka jälkeen sitä voidaan käsitellä digitaalisesti (11).

Tavallinen menetelmä on jakaa digitaalinen signaali noin kymmenen millisekunnin pituisiksi signaalin aikaikkunoiksi, joista lasketaan taajuusspektrit. Mielenkiintoisimpien kaistojen valintaan hyödynnetään yleensä ihmiskorvan taajuusherkkyyttä, sillä tavoitteena on löytää parhaiten puheen äänteitä kuvaavat piirteet. Havainnot muodostetaan **piirrevektoreiksi**, joita äännemalleihin vertaamalla voidaan laskea todennäköisyyksiä kullekin foneemille kullakin hetkellä. Tätä kutsutaan akustiseksi mallintamiseksi. (6, s. 337–339; 9, s. 76–78.)

Pelkkien äännemallien kautta puheen tunnistaminen olisi mahdollista, mutta virhealtista. Suurten tekstiaineistojen perusteella eri kielistä voidaan muodostaa kielimalleja, joissa määritellään sanojen esiintymistodennäköisyyksiä ja todennäköisimpiä ääntämismalleja. Tekstiaineistojen perusteella voidaan myös muodostaa tilastollinen malli sanojen välisistä riippuvuuksista, jolloin saman kuuloiset sanat voidaan helpommin yhdistää oikeisiin konteksteihin. Puheen dekoodaus tekstiksi pohjautuu siis akustisen mallintamisen ja kielimallinnuksen todennäköisyyksiin (kuva 1). (6, s. 337–339.)



KUVA 1. Puheentunnistuksen toimintakaavio (mukaillen 9, s.78)

Puheentunnistuksen akustinen mallintaminen on pitkälti perustunut kätketyn Markov-tilamallin (engl. Hidden Markov Model, HMM) kanssa käytettyyn Gaussian Mixture -jakaumamalliin. Puhe on hyvin monimutkainen signaali, jossa yksi äännekin voi muodostua monin eri tavoin. Aikaikkunoihin jaetusta äänisignaalista voidaan kuitenkin tunnistaa tilojen välisiä riippuvuuksia tilastollisella mallinnuksella, joka helpottaa äänteiden tunnistamista. Kätketyn Markov- ja Gaussian Mixture -mallien

avulla voidaan laskea äänteiden todennäköisyyksiä näiden tunnettujen äänteiden välisten riippuvuuksien perusteella. Nykyään hyödynnetään myös **syväoppivia neuroverkkoja** (engl. Deep Neural Networks, DNN), jotka ovat viime aikoina mahdollistaneet uusia suuntauksia puheentunnistuksessa, niin akustiseen kuin kielelliseen mallinnukseenkin. (12, s. 20, 29.)

2.2.2 Puheen ymmärtäminen

Ihmisen aivoissa äänteet ja sanat vastaavat tiettyjä neurosignaalikuviota, ja hermoverkko mahdollistaa niistä merkitysten ymmärtämisen sekä uusien assosiaatioiden luomisen. Tekoälylle puheen ymmärtäminen taas on iso haaste. Sillä ei ole ihmisille tyypillisiä ymmärtämisen ominaisuuksia. Se pystyy kyllä noudattamaan ennalta määrättyjä ohjeita, kuten lausejäsentelyitä ja kieliopin perusteita, ja se kykenee vastaamaan sille esitettyihin kysymyksiin opetettujen sääntöjen mukaisesti. Tästä voi syntyä illuusio, että tekoäly ymmärtää puhetta, vaikka se ei oikeasti ymmärrä. Tavallinen tietokone ottaa informaation vastaan numeroina, mutta se ei tiedä, mitä sen käsittelemä numerotieto edustaa. (13, s. 45, 218.)

Tekoälyn ei tosin edes tarvitse ymmärtää kaikkea. Lähes kaikissa puheohjattavissa sovelluksissa on täysin riittävää, että ohjelma osaa tunnistaa vain tiettyjä sille opetettuja sanoja tai lauseita (13, s. 45, 50). Puheen ymmärtäminen, eli käyttäjän aikeiden tunnistaminen yhdistää tekniikoita käyttäjämallinnuksesta, luonnollisen kielen ymmärtämisestä, todennäköisyyksistä ja koneoppimisesta (14). Usein sen tavoitteena on yhdistää käyttäjän puhe ennalta tiedettyihin komentoihin, riippumatta käyttäjän käyttämistä sana- tai lausemuodoista.

2.2.3 Valmiit puheentunnistuspalvelut

Tekoäly tarvitsee kymmeniä tuhansia tunteja koulutusmateriaalia tehokasta ja tarkkaa puheentunnistusta varten (5). Oman puheentunnistuskoneen rakentamiseen vaadittavaa puhedataa ei vielä ole tarpeeksi saatavilla avoimina aineistoina, etenkään suomen kielellä. Puheentunnistukseen ja aikeiden tunnistamiseen on kuitenkin olemassa valmiita, usein isojen teknologiayritysten ylläpitämiä rajapintoja, joita kehittäjät voivat hyödyntää. Esimerkiksi Googlen tekoälyyn ja koneoppimiseen pohjautuva Speech-to-Text API tarjoaa rajapinnan puheen kääntämiseksi tekstiksi, ja se kattaa yli 120 eri kieltä mukaan lukien suomen kielen (15). Googlella on myös Dialogflow -palvelu, joka

auttaa luonnollisen kielen ymmärtämisessä ja aikeiden tunnistamisessa (16). Monilla muilla toimijoilla, kuten Microsoftilla ja Amazonilla, on myös vastaavia puheohjaukseen soveltuvia rajapintoja.

2.2.4 Puheentunnistuksen nykytilanne suomen kielellä

Suomi tuntuu osittain olevan puheentunnistusteknologian jälkijunassa. Esimerkiksi Yhdysvalloissa puheohjattavat älylaitteet ja assistenttisovellukset ovat paljon yleisempiä. Suomen kielistä tekoälyä on haastavaa kehittää, sillä julkista suurta puhedatapankkia suomen kielellä ei ole saatavilla. Ongelmaan on jo kuitenkin reagoitu, ja yleiseen käyttöön tarkoitetun puhedatan kerääminen on käynnissä. Suomalainen yritys Onerva on aloittanut yleisen suomenkielisen puhedatan keräämisen vuonna 2019. On odotettavissa, että puhedatan määrän lisääntyessä suomen kielen puheen tunnistus kehittyy paremmaksi ja tarkemmaksi. (17; 18.)

Teknologiajättien puheentunnistuspalvelut vaikuttavat kuitenkin olevan melko kehittyneitä jo suomen kielelläkin. Esimerkiksi puheentunnistuksella toimiva Speechnotes-web-sovellus käyttää Googlen puheentunnistuspalvelua ja kääntää suomalaisen puheen hyvin tekstiksi (19). Oman kokeilun perusteella ongelmia suomen kielen puheentunnistuksessa tuottavat kuitenkin pitkät yhdyssanat, kuten *puheentunnistusteknologia*, sekä suomen kielelle tyypilliset moniliitteiset sanat, kuten *mentäisiinköhän* ja *mielenkiintoisiakin*.

2.3 Esimerkkejä sovelluksista ja käyttötarkoituksista

Puheentunnistusta voidaan käyttää mm. tiedonhakuihin, saneluun ja äänikomentoihin. Tunnettuja käyttötarkoituksia tällä hetkellä ovat esimerkiksi virtuaaliavustajat, automaattiset puhelinvastaajat, reaaliaikainen kielen kääntäminen sekä tekstitysten lisääminen videoihin (7).

Puheohjausta voitaisiin hyödyntää myös terveysteknologiassa helpottamaan esimerkiksi vanhusten sekä näkö-, kuulo- ja liikuntaelinvammaisten arkea. Tekoäly ja puheentunnistus voisivat jopa pelastaa ihmishenkiä. Esimerkiksi tanskalainen yhtiö Corti on kehittänyt tekoälyä, joka tunnistaa hätäkeskukseen soittavan äänestä mahdollisia sydäninfarktin merkkejä (20), ja Google on kehittänyt kuuroille ja huonokuuloisille tarkoitetun sovelluksen, joka tunnistaa puheen muuttaen sen tekstiksi reaaliajassa (21).

Puheentunnistukselle voisi olla käyttöä myös monilla muilla aloilla. Esimerkiksi sähköpostiviestien kirjoittaminen, raporttien tekeminen sekä varastojen ja inventaarioiden hallinta nopeutuisivat, jos niitä voisi tehdä puheella. Moni suomalainen ohjelmistokehityksen alan yritys onkin jo ottanut puheentunnistusteknologiaa haltuun ja tarjoaa tällaisia palveluita.

2.4 Äänikäyttöliittymät

Puheohjaus tuo uudenlaisen haasteen käyttöliittymien ja visuaalisen ulkoasun luomiseen. Olemme niin tottuneita visuaalisiin käyttöliittymiin, että äänikäyttöliittymää suunniteltaessa on erityisesti huomioitava, ettei sovellukselle puhuminen ole käyttäjälle aina itsestäänselvyys. Puheohjaus tulee varmasti vaikuttamaan paljon käyttöliittymäsuunnitteluun, oletettavasti samoin kuin noin kymmenen vuotta sitten yleistyneet kosketusnäytöt, jotka mullistivat verkkosivujen suunnittelua. Puheohjauksen yleistyminen voi kuitenkin olla huomattavasti nopeampi muutos, ja käyttöliittymäsuunnittelussa tullaan ehkä näkemään täysin uudenlaisiakin suuntauksia. (22.)

Puheohjauksen käyttöliittymä voi olla hyvin kaukana totutusta näyttöön perustuvasta interaktiosta. Täysin puheohjauksella toimiva sovellus voisi perustua pitkälti käyttäjän kuuloon ja puheeseen, jättäen ruudun katsomisen ja käsin tehtävät toiminnot minimiin tai kokonaan pois. Puheohjattavissa käyttöliittymissä kohdataan kuitenkin saavutettavuuteen liittyviä ongelmia, erityisesti kuulo- ja puhevammaisille sopivien palveluiden kehittämisessä. (22.)

2.5 Puhesynteesi

Puhesynteesi tarkoittaa ohjelmaa, joka muuttaa annetun tekstin puheeksi. Puhuva tietokone luo illuusion koneen kanssa keskustelusta, jolloin puhekäskyjen antaminen voi tuntua ihmiselle luonnollisemmalta.

Maailmalla on jo käytössä melko luontevan kuuloisia puhesyntetisaattoreita. Melko luonteva on yleensä ihan riittävä, sillä mukautuvuutensa ansiosta ihminen pystyy ymmärtämään tietokoneen puhumaa tekstiä hyvin. Synteesille vaikeita opetettavia ovat sanojen ja lauseiden painotukset sekä intonaatiot, jonka takia puhesynteesistä ei ole saatu luotua täysin ihmispuheen kaltaista. (23, s. 57.)

3 WEB-SOVELLUKSET

3.1 Web-sovelluksen määritelmiä

Web-sovellus eli verkkopalvelu on ohjelmisto, joka sijaitsee verkkopalvelimella ja jota käytetään verkkoselaimessa. Toisin kuin natiivi- tai työpöytäsovellus, web-sovelluksen käyttäminen ei vaadi asentamista laitteelle. (24.)

Web-sovelluksen ja **verkkosivun** raja on häilyvä ja ne tarkoittavatkin osittain samaa asiaa. Molemmat sijaitsevat verkkopalvelimella ja toimivat selaimessa. Yksinkertaistettuna verkkosivu on se, jota selataan ja web-sovellus on se, jota käytetään. Web-sovellus yleensä tarjoaa jonkin käyttötarkoituksen tai toiminnon, ja verkkosivulla esitetään sisältöä. Usein verkkosivustot ovat näiden yhdistelmiä, niissä on sekä staattista että interaktiivista sisältöä. (25.)

Toisinaan web-sovellus-termillä viitataan **PWA**- eli Progressive Web App -sovelluksiin. Web-sovellus tai verkkosivu voidaan rakentaa PWA-sovellukseksi, jolloin se on ladattavissa käyttäjän laitteelle samaan tapaan kuin natiivisovellus. PWA on hyödyllinen etenkin mobiililaitteille, jolloin web-sovelluksesta saadaan nopeammin latautuva, sitä voidaan käyttää offline-tilassa, ja sillä pystytään käyttämään laitteen sisäisiä toimintoja hyödyksi enemmän kuin tavallisella web-sovelluksella. (26.) Tässä työssä web-sovelluksella kuitenkin tarkoitetaan kaikkia verkkoselaimessa käytettäviä web-sovelluksia, ei vain PWA-sovelluksia.

SPA- eli Single Page App -sovellukset ovat nykyaikaisia **yhden sivun web-sovelluksia**. Niiden rakentamiseen käytetään usein JavaScript ohjelmistokehyksiä (engl. frameworks), kuten Angular, Vue.js tai React. Yhden sivun taktiikka tarkoittaa käytännössä sitä, että jokaista linkin klikkausta sovelluksessa ei avata uuteen sivuun. Sen sijaan sisällöt haetaan aina tarvittaessa datana ja päivitetään sovelluksen näkymiin. Tämä tekniikka vähentää palvelimelle tehtävien pyyntöjen määrää, eikä sivuja tarvitse renderöidä jatkuvasti uudestaan. SPA-sovellukset ovat usein nopeita verrattuna monen sivun sovelluksiin. (27.)

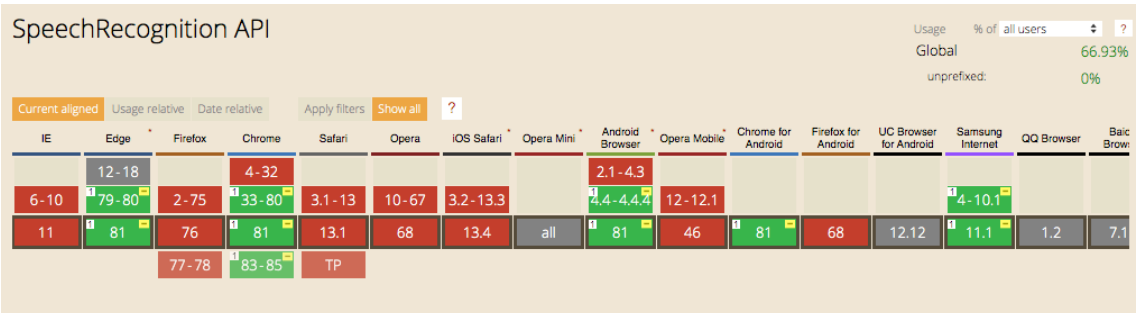
3.2 Puheohjauksen mahdollisuudet web-sovelluksissa

Web-sovelluksiin on toistaiseksi melko rajalliset puheentunnistusmahdollisuudet. Tällä hetkellä suuri osa puheentunnistusalgoritmeista sekä puhetta sisältävistä datapankeista on suljettuja aineistoja. Kehittäjien on kuitenkin mahdollista käyttää valmiita kolmannen osapuolen puheentunnistuksen palveluita rajapinnan kautta, joita tarjoavat mm. Google Cloudin Speech-To-Text- ja Microsoft Azuren Speech to Text -palvelut. Nämä ja lähes kaikki muutkin puheentunnistuksen palvelut ovat käyttömaksullisia, mutta mahdollistavat usein ilmaisen kokeilun tai kevyen käytön (15; 28).

3.2.1 Web Speech API

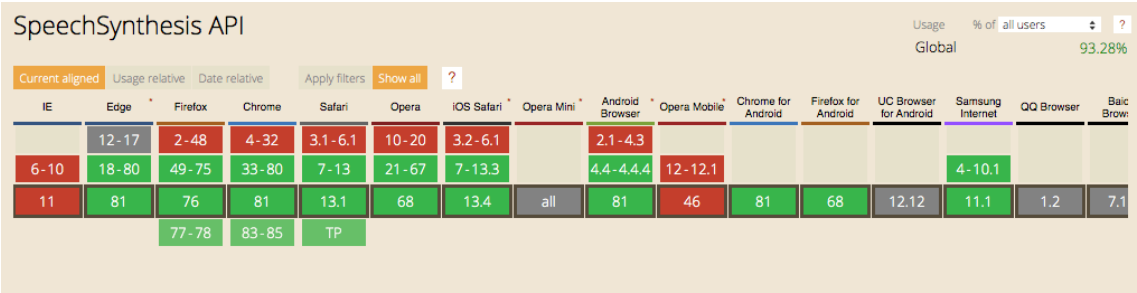
WWW-standardeja kehittävä W3C-yhteisö julkaisi vuonna 2012 Web Speech API -määritelmän, joka mahdollistaa puheentunnistuspalveluiden helpomman käytön web-sovelluksissa. Se tarjoaa web-kehittäjille ilmaisen tavan käyttää puheentunnistusta sekä puhesynteesiä selaimen kautta. Web Speech APIa tukeva selain lähettää web-sovelluksesta tulevat pyynnöt käyttämäänsä puheentunnistus- tai puhesynteesipalveluun ja palauttaa sitten tulokset takaisin web-sovellukseen. (29; 30.)

Tällä hetkellä Web Speech APIn Speech Recognition -puheentunnistusohjain on kokeellinen ja vain rajallisesti tuettu eri selaimissa. Lähteiden mukaan sille olisi osittainen tuki Chrome-, Edge-, Samsung Internet- ja Chrome Android-selaimilla (kuva 2). (31.)



KUVA 2. Kuvakaappaus Can I Use -sivustolta. Speech Recognition APIn selaintuki. (31.)

Web Speech API:n Speech Synthesis -puhesynteesiohjain on myös toistaiseksi kokeellinen toiminto, mutta on jo paljon paremmin tuettu eri selaimissa kuin puheentunnistus (kuva 3).



KUVA 3. Kuvakaappaus Can I Use -sivustolta. Speech Synthesis APIn selaintuki. (31.)

3.2.2 Avoin lähdekoodi ja data

Web-sovellusten puhetunnistusominaisuuksien kehitystä hidastaa avoimen lähdekoodin ja puhedatan puute. Mozilla kehittää parhaillaan omaa Deep Speech -nimistä avoimen lähdekoodin palvelua puheentunnistuksen tarkoituksiin. Mozilla on käynnistänyt Common Voice -projektin ja kerää nyt verkkosivustonsa kautta puhedataa, jonka tuottamiseen ja tarkistamiseen kuka tahansa voi osallistua. Common Voice -projektin tarkoitus on kerätä laaja ääniaineisto, joka tarjoaisi kehittäjille julkisen äänipankin omien puheentunnistussovellusten kehittämiseen. Avoin data ja avoimen lähdekoodin palvelu avaisivat paljon uusia mahdollisuuksia puheentunnistuksen kehittämiselle myös web-sovelluksia varten. Todennäköisesti Mozilla lisää sitten viimeistään Web Speech APIn Speech Recognition -ohjaimen tuen Firefox-selaimellekin. Toistaiseksi tuki on vasta Mozillan kokeellisessa Nightly-selaimessa, joka käyttää Googlen puheentunnistuspalveluita. (32; 33.)

4 PUHEOHJAUKSELLA TOIMIVA WEB-SOVELLUS

Työn konkreettisempana vaiheena kokeiltiin puheohjattavan web-sovelluksen kehittämistä. Sovellus rakennettiin vain kokeilua varten, mutta sitä tai sen osia voi hyödyntää jatkossa muissa omissa projekteissa. Sovelluksen toteutuksessa haluttiin erityisesti testata saatavilla olevien puheentunnistusrajapintojen toimintaa verkkoselaimessa, sekä kokeilla miten puheohjattavan web-sovelluksen käyttöliittymää voitiin ideoida ja kehittää mahdollisimman helppokäyttöiseksi.

4.1 Käyttötarkoitus ja toiminnallisuus

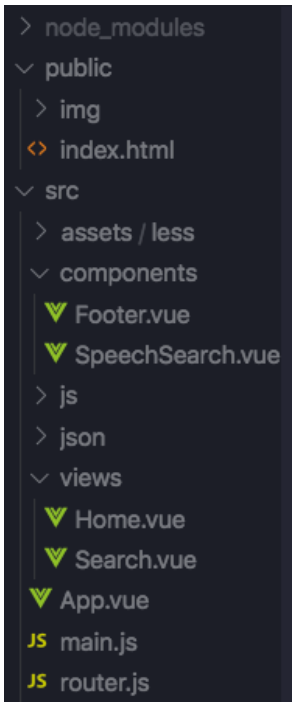
Tavoitteena oli rakentaa web-sovellus, jolla voi etsiä tietoa lintulajeista puheentunnistuksen avulla. Tarkoituksena oli tehdä toiminnallisuus, joka mahdollistaa helpon puhehaun joko lintulajin nimellä tai linnun tuntomerkeillä. Lisäominaisuutena sovellukseen haluttiin lisätä mahdollisuus kuunnella lintujen kuvaustekstejä puhesynteesin avulla.

Pidemmälle kehitettynä tällainen sovellus voisi toimia maastossa lintujen tunnistuksen apuna ja muissa tilanteissa, joissa lintukirjan selaaminen tai hakukoneen käyttäminen lintujen tunnistamiseen olisi hidasta tai vaikeaa. Ajatuksena vastaava puhehaku voisi olla hyödyllinen myös monissa muissa web-sovelluksissa, joissa haetaan tietoa hakukriteerien perusteella.

4.2 Tekninen toteutus

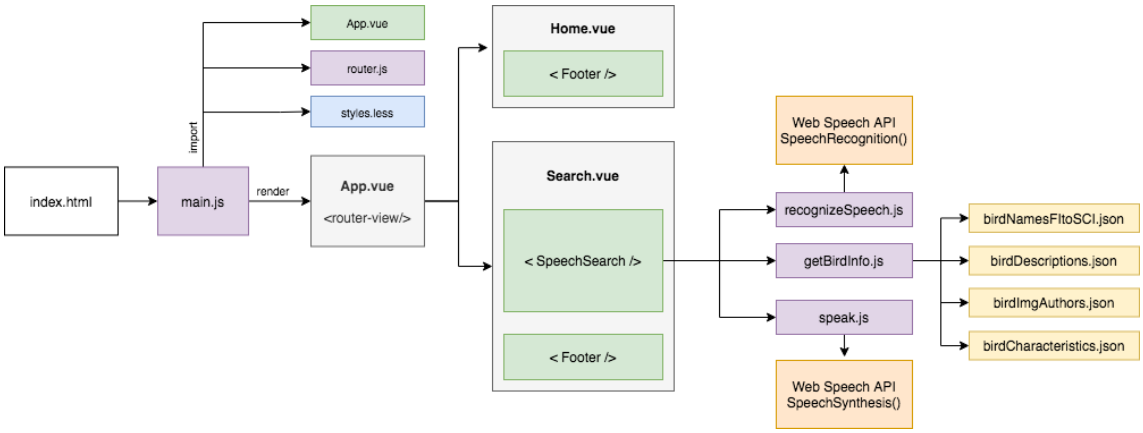
4.2.1 Sovelluksen rakenne

Selaimessa toimiva sovellus rakennettiin käyttäen komponenttipohjaista Vue.js:ää, joka on JavaScript-ohjelmointikehys. Vue.js:n avulla oli helppo luoda SPA-arkkitehtuuri eli yhden sivun web-sovellus. Rakenne ja ulkoasu muodostuivat HTML-kuvauskielestä ja CSS-tyylimuotoiluista, jossa apuna käytettiin Less-CSS-esiprosessoria. Sovelluksen rakenne muodostui Vue.js-sovellukselle tyypillisistä näkymistä ja komponenteista (kuva 4).



KUVA 4. Sovelluksen tiedostorakennetta

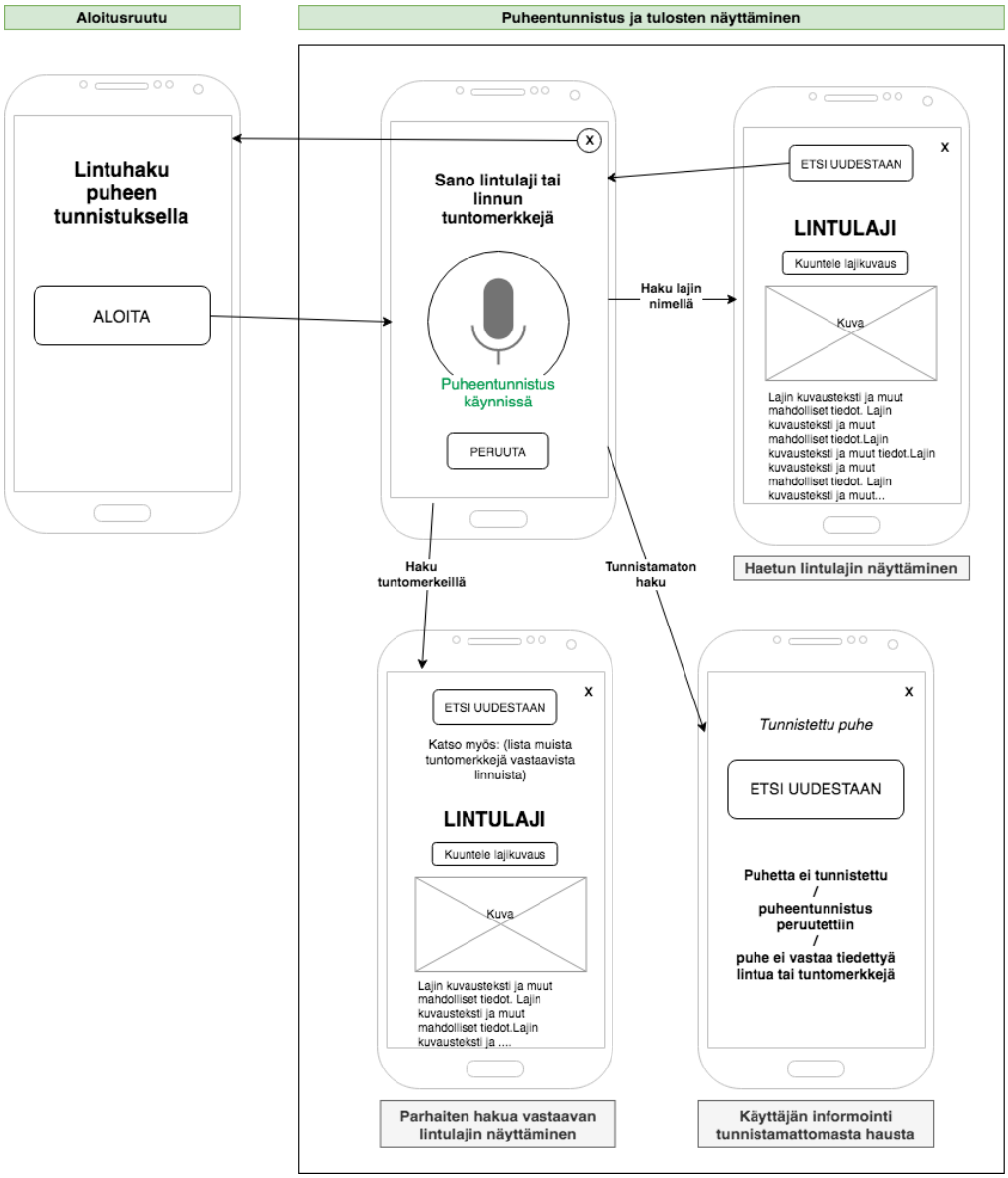
Selaimessa käyttäjälle avautuu *index.html*-sivu. Se pitää sisällään *main.js*-tiedoston, joka tuo tarvittavat komponentit ja tyylitiedostot sekä renderöi *App.vuen* eli sovelluksen näkymän sivulle. Puheentunnistus, puhesynteesi ja lintujen tietojen haku suoritetaan SpeechSearch-komponenttiin tuotujen erillisten JavaScript-moduulien avulla. (Kuva 5.)



KUVA 5. Sovelluksen komponenttien rakenne

4.2.2 Näkymät

Sovelluksen käyttöliittymä koostuu kahdesta eri näkymästä: aloitusruudusta ja puheentunnistuksesta. Puheentunnistuksen näkymässä tulokset vaihtuvat dynaamisesti sisältöalueisiin, joka mahdollistaa sovelluksen sujuvan käytön ilman sivujen vaihtoa. (Kuva 6.)



KUVA 6. Sovelluksen näkymien mockup-suunnitelma

Sovellus jaettiin tarkoituksellisesti kahteen eri näkymään, jotta eri näkymiin pääseminen suoraan selaimen osoiteriviltä olisi mahdollista. Vue.js toteutukseen lisättiin Vue Router -paketti, joka synk-

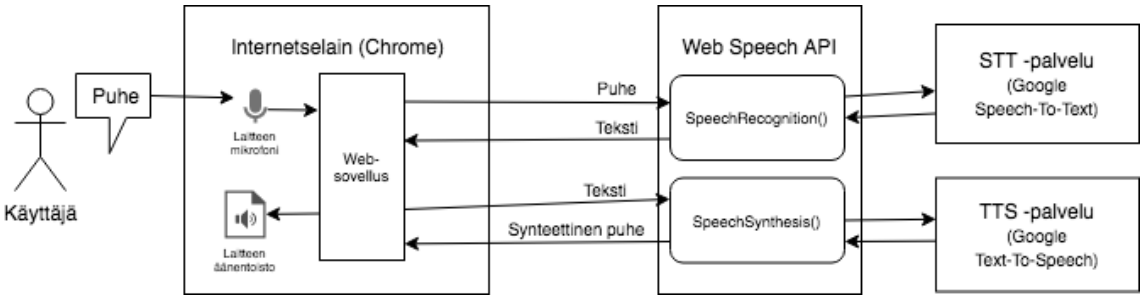
ronoi SPA-sovelluksen näkymät URL-osoitteiksi. Sen avulla aloitusruudun osoitteeksi voitiin määritellä */lintuhaku*, ja puheentunnistuksen osoitteeksi */lintuhaku/puhu*, mikä mahdollistaa tarvittaessa nopeamman pääsyn suoraan puhehakunäkymään.

4.2.3 Puheentunnistus ja puhesynteesi

Puheentunnistuksen ja puhesynteesin toteutukseen käytettiin **Web Speech API** ohjelmointirajapinnan Speech Recognition- ja Speech Synthesis -ohjaimia.

Käyttäjän avatessa hakunäkymän sovellus luo uuden SpeechRecognition-objektin ja aloittaa puheentunnistuksen, kun näkymä on asentunut (engl. mounted). Käyttääkseen puhehakua käyttäjän on kuitenkin ensin annettava web-sovellukselle lupa tallentaa ääntä laitteen mikrofonilla. Web Speech API vastaa selaimessa mikrofonin käyttöluvan kysymisestä.

Kun käyttäjä on aloittanut puhehaun, Web Speech API tunnistaa milloin puhe loppuu ja lähettää pyynnön STT-palveluun, jossa äänitetty puhe käännetään tekstiksi. Samoin toimii puhesynteesi, mutta puheen sijasta TTS-palveluun lähetetään teksti, joka käännetään synteettiseksi puheeksi. (Kuva 7.)



KUVA 7. Sovelluksen puheentunnistuksen ja puhesynteesin prosessi

Chrome-selaimessa tallennettu audio lähtee Googlen verkkopalvelimelle tunnistettavaksi. Puheentunnistusominaisuutta ei voi käyttää ilman verkkoyhteyttä. (30.)

Ennen puheentunnistuksen käyttöönottoa sovelluksessa, tarkistetaan tukeeko käytetty selain Speech Recognition -ohjainta, ja tieto tästä tallennetaan Boolean-muuttujaan. Tuetuille selaimille määritellään tarvittavat referenssit. Speech Recognition on vasta kokeellinen ominaisuus, joten sen

määrittelyn eteen vaaditaan *webkit*-etuliite (engl. prefix). Mahdollista tulevaa vakaata puheentunnistusominaisuutta varten se voitiin kuitenkin jo lisätä etuliitteettömänäkin valmiiksi mukaan. OR-vertailuoperaattorin avulla valitaan käytettäväksi kokeellinen ominaisuus, ellei selain vielä tarjoa vakaata ominaisuutta. (Kuva 8.)

```
var supported = false;

if (('webkitSpeechRecognition' in window) || ('SpeechRecognition' in window)) {
  supported = true;
  var SpeechRecognition = SpeechRecognition || webkitSpeechRecognition;
  var SpeechGrammarList = SpeechGrammarList || webkitSpeechGrammarList;
} else {
  supported = false;
}
```

KUVA 8. Web Speech APIn määrittelyt sovelluksen recognizeSpeech-moduulissa

Puheentunnistuksen alkaessa luodaan uusi SpeechRecognition-objekti, jolle voidaan antaa eri määrityksiä. Tässä sovelluksessa käytettiin vain arvoa *lang* määrittelemään käytettävää kieltä. Puheentunnistusobjektille olisi voitu määritellä ennalta tiedetty sanasto, joka rajaisi puheentunnistuksen tunnistamia sanoja. Tässä sovelluksessa yritettiin lisätä lintujen suomenkieliset nimet sisältävä sanasto puheentunnistuksen käyttöön, mutta useista kokeiluista huolimatta sen toimivuutta ei pystynyt vahvistamaan, eikä sen toiminnallisuudesta löytynyt tarkempaa tietoa. Puheentunnistus tunnisti suomenkieliset lintujen nimet hyvin ilmankin sitä, mutta se tunnisti myös kaikki muut sanat. (Kuva 9.)

```
this.recognition = new SpeechRecognition();
this.recognition.lang = 'fi-FI';
this.recognition.start();

//this.speechRecognitionList = new SpeechGrammarList();
//this.grammar = '#JSGF V1.0; grammar phrase; public <phrase> = ' + fiBirdNames.FIbirds.join(' | ') + ' ;';
//this.speechRecognitionList.addFromString(this.grammar, 1);
//this.recognition.grammars = this.speechRecognitionList;
```

KUVA 9. Speech Recognition -ohjausliittymän määritykset

Puheentunnistusobjektille lisätään tapahtumakuuntelijoita, joiden avulla puheentunnistuksen tulokset on mahdollista saada sovelluksen käyttöön. Tapahtuma *result* aktivoituu, jos puheentunnistus on onnistunut ja tulokset on vastaanotettu puheentunnistuspalvelusta sovellukseen. Tekstiksi käännetty puhe näytetään käyttäjälle lisäämällä se käyttöliittymän HTML-elementtiin, jonka jälkeen se saadaan myös muiden sovelluksen funktioiden käyttöön jatkokäsittelyä varten. (Kuva 10.)

```
this.recognition.addEventListener('result', event => {
  //console.log(event.results);
  var speech = event.results[0][0].transcript.toLowerCase();
  this.result.innerHTML = speech;
})
```

KUVA 10. Puheentunnistusobjektin result-tapahtumakuuntelija

Tapahtumasta *result* palautuu SpeechRecognitionResultList-objekti, jonka ensimmäinen alkio sisältää tuloksen. Tulosten ensimmäinen alkio sisältää SpeechRecognitionAlternative-objektin, josta saadaan ominaisuuksina todennäköisin tekstiksi muutettu puhe, sekä sen varmuusprosentti. (Kuva 11.)

```
                                                    recognizeSpeech.js?9b90:67
▼SpeechRecognitionResultList {0: SpeechRecognitionResult, length: 1} i
  ▼0: SpeechRecognitionResult
    ▼0: SpeechRecognitionAlternative
      confidence: 0.9122292995452881
      transcript: "pähkinänakkeli"
```

KUVA 11. Result-tapahtumakuuntelijan palauttama todennäköisin tulos, kun puheentunnistus on tunnistanut ja kääntänyt puheen tekstiksi

Tapahtumakuuntelijoiden avulla annetaan käyttäjälle tietoa puheentunnistuksen tilasta ja onnistumisesta. Tapahtuma *audiostart* aktivoituu, kun puheentunnistus on valmis kuuntelemaan puheen, *speechend,* kun puhe on tunnistettu loppuneeksi, ja *error*, jos puheentunnistus on peruuntunut tai puhetta ei tunnistettu. (Kuva 12.)

```
this.recognition.addEventListener('audiostart', () => {
  this.startBtn.innerHTML = 'Sano lintulaji tai linnun tuntomerkkejä';
  this.mic.style.display = "flex";
  this.mic.classList.add("recognizing");
})

this.recognition.addEventListener('speechend', () => {
  this.stopRecognizing();
  this.mic.style.display = "none";
  this.mic.classList.remove("recognizing");
})

this.recognition.addEventListener('error', event => {
  var err = event.error;

  switch(err) {
    case 'abort':
      this.result.innerHTML = 'Puheentunnistus peruutettu. Kokeile uudestaan?';
      break;
    case 'no-speech':
      this.result.innerHTML = 'Puhetta ei tunnistettu. Kokeile uudestaan?';
      break;
  }
  this.stopRecognizing();
  this.mic.style.display = "none";
  this.mic.classList.remove("recognizing");
})
```

KUVA 12. Puheentunnistusobjektin tapahtumakuuntelijoita

Käyttäjä voi sovelluksella myös kuunnella lintujen lajikuvaustekstit puhuttuina. Puhesynteesiä varten haetaan ensin referenssi SpeechSynthesis-ohjaimelle. (Kuva 13.)

```
var synth = window.speechSynthesis;
```

KUVA 13. Puhesynteesin referenssi

Sovelluksen SpeechSearch-komponentista kutsutaan *startSpeech*-funktiota, jolle annetaan ominaisuutena puheeksi käännettävä teksti. Mahdollinen edellinen puhe keskeytetään. Jos tekstimuuttuja ei ole tyhjä, luodaan uusi SpeechSynthesisUtterance-objekti eli puhesynteesin ääni, joka saa ominaisuudekseen luettavan tekstin. Ennen puheen aloitusta puhesynteesin äänelle määritellään käytettävä kieli, korkeus ja nopeus (kuva 14). Puhesynteesin toiminta voidaan keskeyttää, jatkaa tai lopettaa (kuva 15).

```
export function startSpeech(text){
  synth.cancel();

  if (text !== '') {
    var utterance = new SpeechSynthesisUtterance(text);
    utterance.lang = 'fi-FI'; // Finnish voice
    utterance.pitch = 1.0;
    utterance.rate = 0.9;
    synth.speak(utterance);
  }
}
```

KUVA 14. Puhesynteesin määritelmät

```
export function pauseSpeech(){
  synth.pause()
}

export function resumeSpeech(){
  synth.resume()
}

export function stopSpeech(){
  synth.cancel();
}
```

KUVA 15. Puhesynteesin komennot

4.2.4 Aikeen tunnistaminen

Kun puheentunnistus on suoritettu, tekstiksi muutettu puhe välitetään SpeechSearch-komponenttiin, joka välittää sen edelleen eri funktioon aikeiden tunnistamista varten. Tässä sovelluksessa aikeiden tunnistaminen tarkoittaa haetun lintulajin nimen tai tuntomerkkien tunnistamista tekstiksi muutetusta puheesta.

SpeechSearch-komponentilla on oma *getBirdInfo-funktio,* joka kutsuu moduulin *getBirdFromSpeech*-funktiota, ja välittää sille ominaisuutena tekstiksi muutetun puheen. Jos tässä funktiossa löydetään puheesta vastaavuus johonkin lintulajiin, lintulajin tiedot palautetaan *birdInfo*-muuttujaan. (Kuva 16.)

```
getBirdInfo: function(speech) {
  var birdInfo = getBirdFromSpeech(speech.toLowerCase());

  if(birdInfo) {
    // Lisätään saadut linnun tiedot sovelluksen näkymään
```

KUVA 16. SpeechSearch-komponentin oma funktio, joka kutsuu lintulajin tunnistamisen funktiota

Tämän sovelluksen käyttäjän aikeet olivat hyvin rajatut, joten tunnistaminen voitiin suorittaa melko yksinkertaisesti olemassa olevien aineistojen perusteella. Sovelluksessa puheesta tunnistettu teksti käydään läpi muutamalla eri tavalla, jotta käyttäjälle löydetään oikea tulos.

Jos puheentunnistuksesta saatuja sanoja on vain yksi, tarkistetaan sisältyykö se suomenkielisten lintulajien nimien listaan. Jos sanoja on enemmän kuin yksi, sanojen jono muutetaan ensin taulukkomuotoon ja taulukon alkiot käydään läpi verraten niitä yksitellen suomenkielisiin linnun nimiin. Ensimmäinen vastaava lajinimi tallennetaan tulokseksi. Lisäksi tallennetaan muuttujaan tieto, onko tekstiksi muutetussa puheessa sana ”kuuntele”.

Jos puheesta ei tunnistettu yhtään lintulajin nimeä, voidaan olettaa, että käyttäjä on antanut hakukriteerit tuntomerkkeinä. Tekstiksi muutettu puhe välitetään toiseen funktioon ja käsitellään siellä uudestaan. JSON-muotoinen lista lintulajeista sekä niiden tuntomerkeistä muutetaan taulukkomuotoon ja käydään läpi alkio kerrallaan. Jokaista haun sanaa verrataan jokaisen alkiossa olevan objektin tuntomerkkisanoihin. Lintulajit arvotetaan sen perusteella, kuinka monta tuntomerkkiä kustakin täsmää käyttäjän hakukriteereihin. Parhaan tuloksen lisäksi näytetään enintään viisi seuraavaksi parasta tulosta linkkeinä.

Tunnistetun lintulajin tieteellinen nimi välitetään erilliseen palautusfunktioon, joka hakee sen perusteella tarvittavat linnun tiedot JSON-tiedostoista ja palauttaa sitten kerralla kaiken tiedon sovelluksen SpeechSearch-komponenttiin, josta lintulajin tunnistamisen funktiota alun perin kutsuttiin. Samalla välitetään tieto mahdollisten parhaiden tuloksien listasta ja puhesynteesin automaattisesta kuuntelusta. Komponentissa tunnistetun lintulajin tiedot asetetaan sovelluksen näkymään käyttäjän nähtäville, ja kuvaustekstin luku puhesynteesillä aloitetaan, jos käyttäjä on sanonut ”kuuntele”.

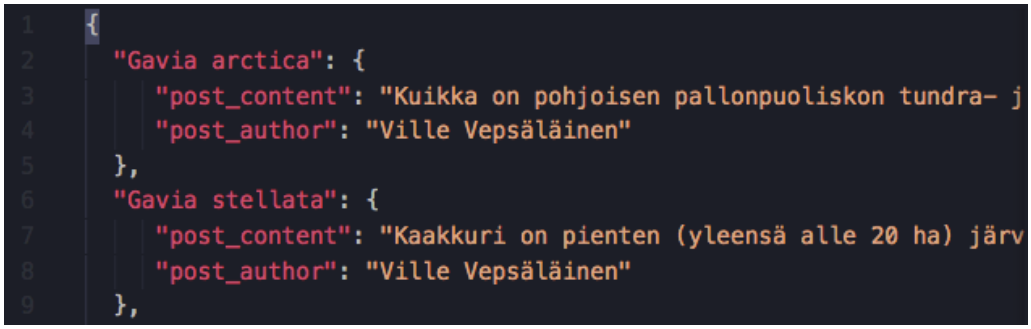
Lintulajien nimien ja tuntomerkkien tunnistaminen tekstiksi muutetusta puheesta on esitetty myös vuokaavioina liitteessä 1.

4.3 Käytetyt aineistot

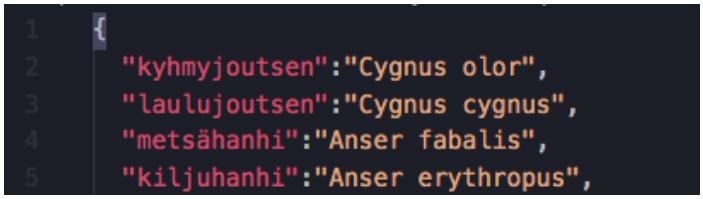
Sovelluksessa käytettiin lintulajien, kuvaustekstien ja kuvien näyttämiseen valmiita aineistoja, jotka olivat saatavilla Suomen Lintuatlaksen internetsivuilla avoimena datana. Sovelluksen näkymiin on

lisätty kuvien ja tekstien tekijänoikeudet sekä lisenssit vaaditulla tavalla. Lintuatlaksen mukaan käytetyt lajikuvat olivat Luonnontieteellisen keskusmuseon asiantuntijoiden tarkastamia, ja niitä oli mahdollista edelleen käyttää lisenssien ehtojen mukaisesti. (34.)

Aineistoja muokattiin omaan käyttöön sopiviksi huomioiden lisenssien ehdot. Tekstitiedostot muutettiin JSON-muotoon, jolloin niiden tietoja pystyi käsittelemään paremmin sovelluksessa (kuva 17). Aineistojen pohjalta muodostettiin myös oma jäsennelty tietoaineisto, joka yhdisti suomenkielisen lintulajin nimen sen tieteelliseen nimeen (kuva 18).



KUVA 17. Ote lajien kuvaustekstien JSON-tiedostosta



KUVA 18. Ote lajien nimien JSON-tiedostosta

Sovellukseen kokeiltiin myös mahdollisuutta hakea lintuja tuntomerkkien perusteella. Tähän tarkoitukseen ei ollut tarjolla valmiita avoimen datan lähteitä, joten testausta varten luotiin oma JSON-tiedosto, johon tallennettiin muutamien lintulajien tuntomerkkejä (kuva 19).

KUVA 19. Ote lajien tuntomerkkien JSON-tiedostosta

4.4 Käyttöliittymä

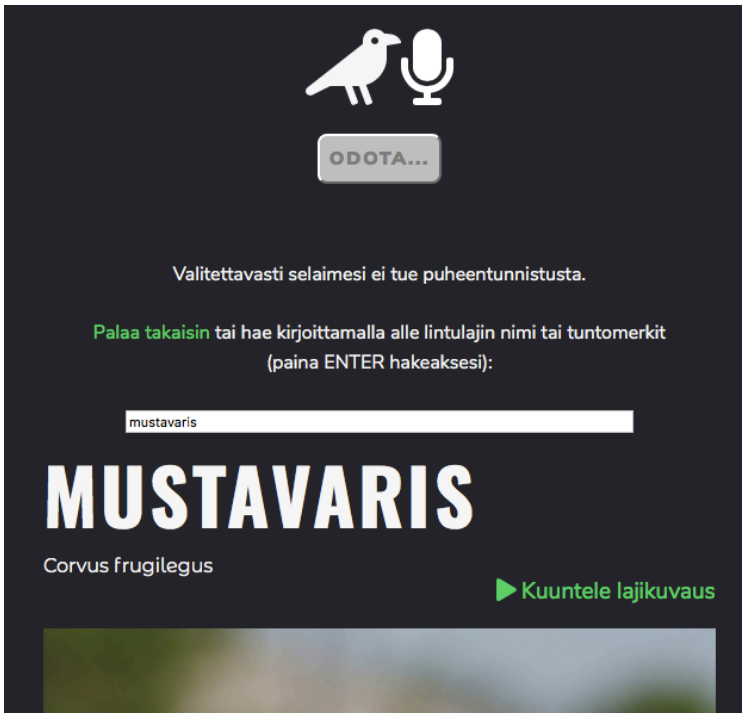
Tässä sovelluksessa visuaalinen käyttöliittymä tukee puhekäyttöliittymän käyttöä. Sovelluksen tarkoituksena oli esittää tietoja ja kuvia linnuista, jolloin visuaalinen näkymä oli tarpeellinen. Etenkään suomen kielellä puhetta tunnistavat web-sovellukset eivät ole vielä kovin yleisiä, joten oli hyödyllistä pyrkiä esittämään käyttäjälle mahdollisimman selkeät visuaaliset ohjeet puheentunnistuksen hyödyntämiseksi.

Puheohjauksella oli tarkoitus helpottaa niitä tilanteita, joissa käyttäjä joutuisi muuten kirjoittamaan tekstiä hakeakseen lintulajia. Erityisesti tällä tavoiteltiin mobiililaitteella tehtävien hakujen helpottamista, sillä puhelimen näppäimistöllä kirjoittaminen voi olla joissain tilanteissa vaikeaa.

Käyttöliittymässä pyrittiin huomioimaan sovelluksen selkeys, helppokäyttöisyys ja informatiivisuus. Sekaannusten välttämiseksi käyttäjän tuli nähdä selvästi, milloin puhetta kuunnellaan ja milloin puheentunnistus epäonnistui. Sovelluksessa puheentunnistus ilmaistiin animoidulla mikrofoni-kuvakkeella (kuva 20).

KUVA 20. Kuvakaappaus sovelluksen puheentunnistuksen näkymästä. Mikrofoni-kuvakkeen ympärillä oleva vihreä alue sykkii puheen kuuntelun merkkinä.

Puheentunnistukseen heikon selaintuen takia sovellukseen lisättiin mahdollisuus myös täysin visuaaliseen käyttöliittymään. Puuttuvan tuen selaimilla lintuja on mahdollista hakea tekstisyötteellä (kuva 21).



KUVA 21. Varasuunnitelma selaimille, jotka eivät tue Web Speech APIn puheentunnistusta

Tekstihaussa voitiin hyödyntää samoja lintulajin hakemisen funktioita kuin puheentunnistuksessa, mutta tekstisyötteestä tuleva teksti täytyy ensin käsitellä mahdollisten sopimattomien merkkien varalta (kuva 22).

```
submitText: function() {
  var text = this.$refs.notSupportedForm.value;
  var clearedText = text.replace(/[&\/\\#,+()$~%.'":*?<>{}]/g, '');
  this.getBirdInfo(clearedText.toLowerCase());
}
```

KUVA 22. Tekstisyötteen tarkistaminen ennen lintulajin tunnistusta

Sovellukseen pyrittiin lisäksi suunnittelemaan oikoteitä, jotka helpottaisivat puheohjauksella tehtäviä komentoja. Niiden avulla käyttäjän on mahdollista vähentää entisestään visuaalisen käyttöliittymän käyttöä käsin, kun haun lisäksi muitakin toimintoja voi suorittaa puheella. Sovellukseen lisättiin yksi tällainen oikotie. Hakiessaan lintulajia puheella käyttäjä voi sanoa hakukriteerien lisäksi sanan ”kuuntele”, jolloin puhesynteesi aloittaa kertomaan linnun kuvaustekstiä ääneen heti kun linnun tiedot on haettu.

Vaikka sovelluksesta ei suunniteltu täysin esteetöntä, otettiin siinä kuitenkin huomioon saavutettavuutta parantavia toimintoja. Esimerkiksi tietokoneella toimintoja pystyy käyttämään ilman hiirtä, liikkumalla tabulaattorin avulla.

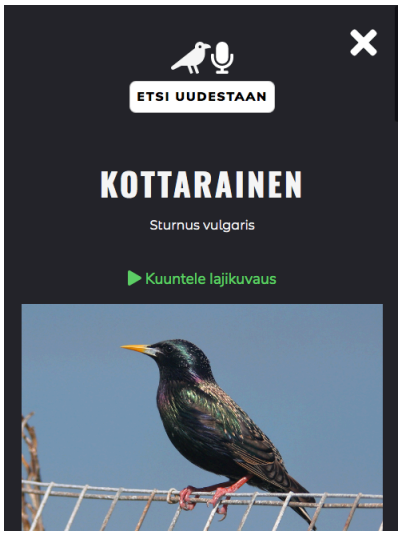
Yksi tavoite oli tehdä sovelluksesta hyvin mobiililaitteella käytettävä. Web-sovellus suunniteltiin responsiiviseksi, jotta se toimisi hyvin kaiken kokoisilla näytöillä. Responsiivinen näkymä voitiin määritellä HTML:llä meta-elementin viewport-ominaisuuden avulla, jonka jälkeen Less-tiedostossa voitiin määritellä CSS-tyylejä erikokoisille näytöille (kuva 23).

```
<meta name="viewport" content="width=device-width,initial-scale=1.0">
```

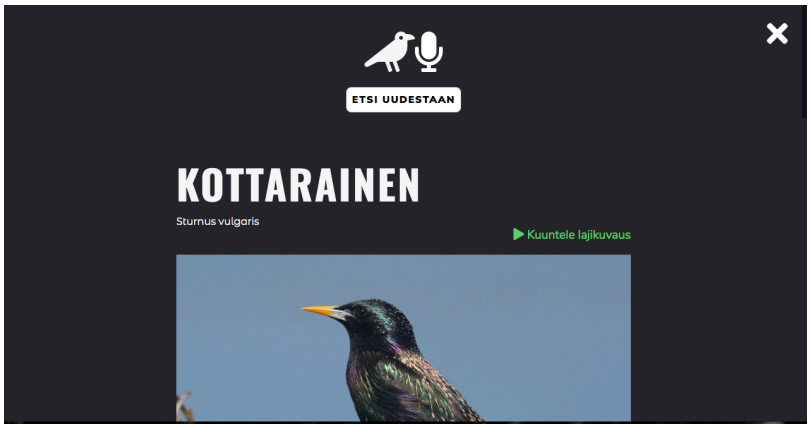
KUVA 23. Viewport-määrittely responsiivisen sovelluksen mahdollistamiseksi

```
.bird-name {
  @media only screen and (min-width: @tablet) {
    max-width: 70%;
  }

  &-title,
  &-scientific {
    @media only screen and (min-width: @tablet) {
      text-align: left;
    }
  }

  &-title {
    margin-bottom: 0;
  }
}
```

KUVA 24. Responsiivisia CSS-tyylimäärittelyjä Less-esiprosessorilla



KUVA 25. Kuvakaappaus sovelluksesta mobiililaitteen kokoisessa näkymässä



KUVA 26. Kuvakaappaus sovelluksesta työpöytäkokoisessa näkymässä

5 TULOKSET

Valmiista testisovelluksesta on katsottavissa esittelyvideo osoitteessa https://youtu.be/WGAiK-GvVzuM, ja liitteessä 2 on kuvakaappauksia sovelluksen näkymistä.

Sovelluksen toteutus onnistui suunnitellulla tavalla. Työn teoriaosuudessa selvisi, että web-sovelluksen puheentunnistustoiminnon toteuttamiseen oli olemassa eri vaihtoehtoja, mutta erityisesti testaustarkoitukseen sopi hyvin Web Speech API, jonka tarjoamien ominaisuuksien pohjalta sovelluksen puheentunnistusominaisuutta lähdettiin kehittämään.

Web Speech API aiheutti sovelluksen toimintaan ja testaamiseen joitain rajoituksiakin. Kyseiselle rajapinnalle ei ole täydellistä tukea selaimissa, ja osa selaimista ei tue kaikkia sen ominaisuuksia ollenkaan. Sovelluksen puheentunnistuksen testaaminen lokaalisti tietokoneen Chrome-selaimella onnistui silti ongelmitta. Tavoitteena oli tehdä hyvin mobiililaitteella toimiva sovellus, joten testaaminen mobiiliselaimella oli tärkeä vaihe. Ongelmaksi muodostui Chrome Android -selaimen kyvyttömyys tehdä rajapintakyselyitä Web Speech rajapintaan, kun sovellusta testattiin lokaalisti ilman TLS-suojattua osoitetta. Sovelluksen julkaiseminen TLS-suojattuun URL-osoitteeseen GitHub Pages- ja Netlify-palveluiden avulla ratkaisi ongelman, jonka jälkeen sovelluksen testaus onnistui myös Android-puhelimen Chrome -selaimella.

Web Speech APIn SpeechSynthesis-ohjaimen resume- ja pause-komennot eivät toimineet mobiililaitteessa Android Chrome -selaimella, jolloin puhesynteesiä ei voinut keskeyttää ja jatkaa uudestaan. Toiminto toimii kuitenkin Chromen työpöytäversiossa. Kyseinen ongelma on raportoitu bugiksi (35). Muuten sovellus toimi moitteettomasti mobiililaitteellakin.

Puheohjattavan käyttöliittymän suunnittelu ja rakentaminen oli kiehtovaa ja auttoi ymmärtämään paremmin puheohjattavien web-sovellusten tekniikkaa sekä käyttömahdollisuuksia. Käyttöliittymän suunnittelun yksi tärkein tavoite oli helppokäyttöisyys, erityisesti mobiililaitteiden näkökulmasta. Se toteutettiin suurilla, selkeillä painikkeilla, mahdollisimman yksinkertaisilla näkymillä ja mahdollisuudella kuunnella lajikuvauksia. Toinen tavoite oli informatiivisuus, jota huomioitiin visuaalisissa elementeissä ja näytettävissä teksteissä, kuten ilmaisemalla, milloin puheentunnistustoiminto kuuntelee aktiivisesti ja milloin puheentunnistus päättyy.

Mielenkiintoinen lisäominaisuus sovelluksen käyttötarkoituksen kannalta olisi ollut mahdollisuus kuunnella lintujen ääniä, ja toimintoa koitettiinkin kehittää. Kokeilujen jälkeen kuitenkin selvisi, että lintujen ääniä tarjoava rajapinta (36) ei hyväksynyt rajapintakyselyitä selainpohjaisesta sovelluksesta.

Web Speech APIsta löytyi kattavasti tietoa W3C:n ja MDN:n dokumenteista (29, 30). Selainkohtaista tietoa kuitenkin löytyi todella vähän, ja joidenkin selainten Web Speech API:n toiminnallisuus jäi mysteeriksi. Osassa lähteistä kerrottiin mm. Microsoftin Edge-selaimen puheentunnistusohjaimen tuesta, mutta testatessa Edgellä ohjain ei kuitenkaan toiminut.

Web Speech API oli silti hyvä keino testata puheentunnistusta web-sovelluksissa. Toistaiseksi sen puheentunnistustoiminnot selaimissa ovat epävakaita, ja jatkossa rajapintaan voi tulla muutoksia. Tämän vuoksi tekniikan hyödyntäminen tuotannon sovelluksissa ei toistaiseksi ole kannattavaa, sillä kokeelliset ominaisuudet voivat muuttua ja kehittyä yllättäen.

Toistaiseksi tarjolla ei ole vielä kovin paljon muita vaihtoehtoja web-sovellusten puheentunnistustoimintojen tekemiseksi. On huomioitava, että nykyisiä teknologiajättien tarjoamia puheentunnistusrajapintoja hyödynnettäessä kaikki puhedata siirtyy kolmannen osapuolen palvelimelle. Nykypäivänä tietojen kerääminen ja Big Datan hyödyntäminen on yleistä, mutta sitä myös kammoksutaan. Ongelma on se, että emme tiedä, mihin kaikkeen tätä tallennettua dataa käytetään.

Toinen huomioitava asia puheohjattavien sovellusten suunnittelussa on tietosuoja ja käytön mielekkyys. On asiayhteyksiä ja tilanteita, joihin puheohjaus ei ole sopiva menetelmä. Esimerkiksi julkisessa tilassa verkkopankkiin kirjautuminen ja tilisaldon kysyminen ääniohjauksella voisi olla haitallista, sillä arkaluontoiset tiedot kantautuisivat herkästi ulkopuolisten korviin. Sama koskee salasanoja ja muita salassa pidettäviä tietoja. Lintulajien haku puheella on kuitenkin mainio esimerkki käyttötarkoituksesta, johon puheentunnistus sopii hyvin.

6 YHTEENVETO

Puheentunnistuksen teknologiaan tutustuminen oli mielenkiintoista. Puheentunnistus on varsin kiehtova tutkimuksen ala, sillä sen kehityksessä on otettu ja otetaan edelleen paljon vaikutteita ihmiskehon toiminnasta. Mielenkiintoista oli myös huomata, että monen vuosikymmenen kehityksen jälkeenkin tämä teknologia tuottaa edelleen haasteita ja pohdittavaa kehittäjille ja tutkijoille monella alalla. Onneksi haasteista huolimatta puheentunnistuksen teknologiaa voidaan hyödyntää jo todella hyvin tässäkin kehityksen vaiheessa.

Puheohjaus mahdollistaa aivan uudenlaisia käyttöliittymiä niin älylaitteissa, mobiilisovelluksissa, verkkosivuilla kuin web-sovelluksissakin. On jännittävää seurata, mihin suuntaan puheohjaus johdattaa nykyajan käyttöliittymiä. Voisiko esimerkiksi tulevaisuudessa ohjelmistokehittäjän arkipäivää olla puheella koodaaminen? Tällä säästyttäisiin ainakin monilta istumatyön ja käsien rasituksen vaivoilta, sillä puheohjaus mahdollistaisi työn tekemisen ilman näppäimistöä ja hiirtä, sekä joissain työtehtävissä myös ilman näyttölaitetta. Puhekäyttöliittymät vaativat ihmisiltäkin paljon sopeutumista, sillä olemme niin tottuneita näppäimistöihin ja näyttöihin.

Puheohjauksen yleistymistä web-sovelluksissa ja verkkosivustoilla vielä kuitenkin odotetaan. Todennäköisesti suurempi läpimurto vaatii ensin avoimen lähdekoodin ja puhedatan kehittymistä, parempaa puheentunnistuksen tukea eri verkkoselaimille sekä ihmisten käyttöliittymätottumusten muuttumista.

LÄHTEET

1. Rekiaro, Ilkka – Kähönen, Antti 1984. Puhu tietokoneelle! – ja kuuntele, kun se vastaa. Tekniikan Maailma, nro 7. S. 58–60.

2. Kerkkänen, Tuomas 2016. Kännykkä tunnistaa puheesi pian yhä paremmin – Savon murre silti liian kova pala. Yle. Saatavissa: https://yle.fi/uutiset/3-9013931. Hakupäivä 1.5.2020.

3. Number of smartphone users worldwide from 2016 to 2021. 2020. Statista. Saatavissa: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. Hakupäivä 18.4.2020.

4. Percentage of all global web pages served to mobile phones from 2009 to 2018. 2020. Statista. Saatavissa: https://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/. Hakupäivä 18.4.2020.

5. Gävert, Hugo – Luoma, Kristian 2020. Puheentunnistus ja OP Puhe. OP Tech (podcast). Saatavissa: https://open.spotify.com/episode/0ZidyZhbAxZKHXvNLcopfY. Hakupäivä 12.4.2020.

6. Kurimo, Mikko 2009. Puheentunnistus. Teoksessa Aaltonen, Olli (toim.). Puhuva ihminen. Puhetieteiden perusteet. Helsinki: Otava. S.336–343.

7. Kurimo, Mikko 2017. Automatic recognition of human speech. Aalto University. Saatavissa: https://www.youtube.com/watch?v=96KTgK_GqY0. Hakupäivä 27.4.2020.

8. Suomi, Kari – Toivanen, Juhani – Ylitalo, Riikka 2006. Fonetiikan ja suomen äänneopin perusteet. Helsinki: Gaudeamus.

9. Kurimo, Mikko 2008. Puheentunnistus. Puhe ja Kieli nro 2. S. 73–83. Saatavissa: https://journal.fi/pk/article/view/5112. Hakupäivä 19.4.2020.

10. Kielitieto. Kielet. Kotimaisten kielten keskus. Saatavissa: https://www.kotus.fi/kielitieto/kielet. Hakupäivä 24.4.2020.

11. Elsea, Peter 1996. Microphones. UCSC Electronic Music Studios. Saatavissa: http://art-sites.ucsc.edu/EMS/Music/tech_background/TE-20/teces_20.html Hakupäivä 7.4.2020.

12. Smit, Peter 2019. Modern subword-based models for automatic speech recognition. Väitöskirja. Aalto University publication series, Doctoral Dissertations 97/2019. Helsinki: Aalto Yliopisto, signaalinkäsittelyn ja akustiikan laitos. Saatavissa: https://aalto-doc.aalto.fi/bitstream/handle/123456789/38073/isbn9789526085661.pdf. Hakupäivä 5.5.2020.

13. Haikonen, Pentti O. A. 2017. Tietoisuus, tekoäly ja robotit. Helsinki: Art House.

14. Sukthankar, Gita – Geib, Christopher – Hung Hai Bui – Pynadath, David V. – Goldman, Robert P. 2014. Plan, activity and intent recognition (E-kirja). Burlington: Morgan Kauffman.

15. Speech-to-Text. Google Cloud. Saatavissa: https://cloud.google.com/speech-to-text. Hakupäivä 4.4.2020.

16. Dialogflow. Google Cloud. Saatavissa: https://cloud.google.com/dialogflow. Hakupäivä 4.4.2020.

17. Onerva-bot voisi soittaa kaikki ikäihmiset läpi ja varmistaa heidän voinnin. Onerva. Saatavissa: https://onervahoiva.fi/onerva-bot-voisi-soittaa-kaikki-ikaihmiset-lapi-ja-varmistaa-heidan-voinnin/. Hakupäivä 1.5.2020.

18. Suomenkielinen tekoälyn kehittäminen etenee (Digi-Lönnrot, osa 2). Onerva. Saatavissa: https://onervahoiva.fi/suomenkielinen-tekoalyn-kehittaminen-etenee-digi-lonnrot-osa-2/. Hakupäivä 1.5.2020.

19. Speechnotes. Saatavissa: https://speechnotes.co/. Hakupäivä 8.5.2020.

20. Ruokonen, Janne 2019. Volyymit kaakkoon: Puhe on tekoälyn käytetyimpiä sovelluksia 2019. Tulos. Saatavissa: https://www.tulos.fi/artikkelit/volyymit-kaakkoon-puhe-tekoalyn-kaytetyimpia-sovelluksia-2019/. Hakupäivä 1.4.2020.

21. Mikkola, Heli 2019. Puheentunnistus avuksi kuuroille. Lääkärilehti nro 17. Saatavissa: https://www.laakarilehti.fi/ajassa/ajankohtaista/puheentunnistus-avuksi-kuuroille/. Hakupäivä 29.4.2020.

22. Amunwa, Jason. The UX of Voice: The Invisible Interface. Telepathy. Saatavissa: https://www.dtelepathy.com/blog/design/the-ux-of-voice-the-invisible-interface. Hakupäivä 1.4.2020.

23. Suni, Antti 2008. Puhesynteesi ja lausepaino. Puhe ja Kieli nro 2. S. 57–72. Saatavissa: https://journal.fi/pk/article/view/5111. Hakupäivä 5.5.2020.

24. Web Application. 2014. Tech Terms. Saatavissa: https://techterms.com/definition/web_application. Hakupäivä 26.4.2020.

25. MacPherson, Mary 2019. Websites vs. Web App: What's the difference? Medium. Saatavissa: https://medium.com/@essentialdesign/website-vs-web-app-whats-the-difference-e499b18b60b4. Hakupäivä 3.5.2020.

26. Haluaisitko puhua nettisivullesi? Rakenna web-sovellus. 2018. Red & Blue. Saatavissa: https://redandblue.fi/fi/haluaisitko-puhua-nettisivullesi-rakenna-web-sovellus/. Hakupäivä 26.4.2020.

27. Halme, Anu 2018. Mikä on Single Page App ja mihin sitä käytetään? City Dev Labs. Saatavissa: https://citydevlabs.fi/single-page-app/. Hakupäivä 1.5.2020.

28. Speech To Text. Microsoft Azure. Saatavissa: https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/. Hakupäivä 4.4.2020.

29. Natal, André – Shires, Glen – Cáceres, Marcos – Jägenstedt, Philip – Wennbordg, Hans 2020. Web Speech API. Draft Community Group Report. Saatavissa: https://wicg.github.io/speech-api/. Hakupäivä: 2.4.2020.

30. SpeechRecognition. MDN Web Docs. Mozilla. Saatavissa: https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition. Hakupäivä: 10.4.2020.

31. Can I Use. Hakusana Web Speech API. Saatavissa: https://caniuse.com/#search=web%20speech%20api. Hakupäivä 26.4.2020.

32. Common Voice. Mozilla. Saatavissa: https://voice.mozilla.org/fi. Hakupäivä 2.4.2020.

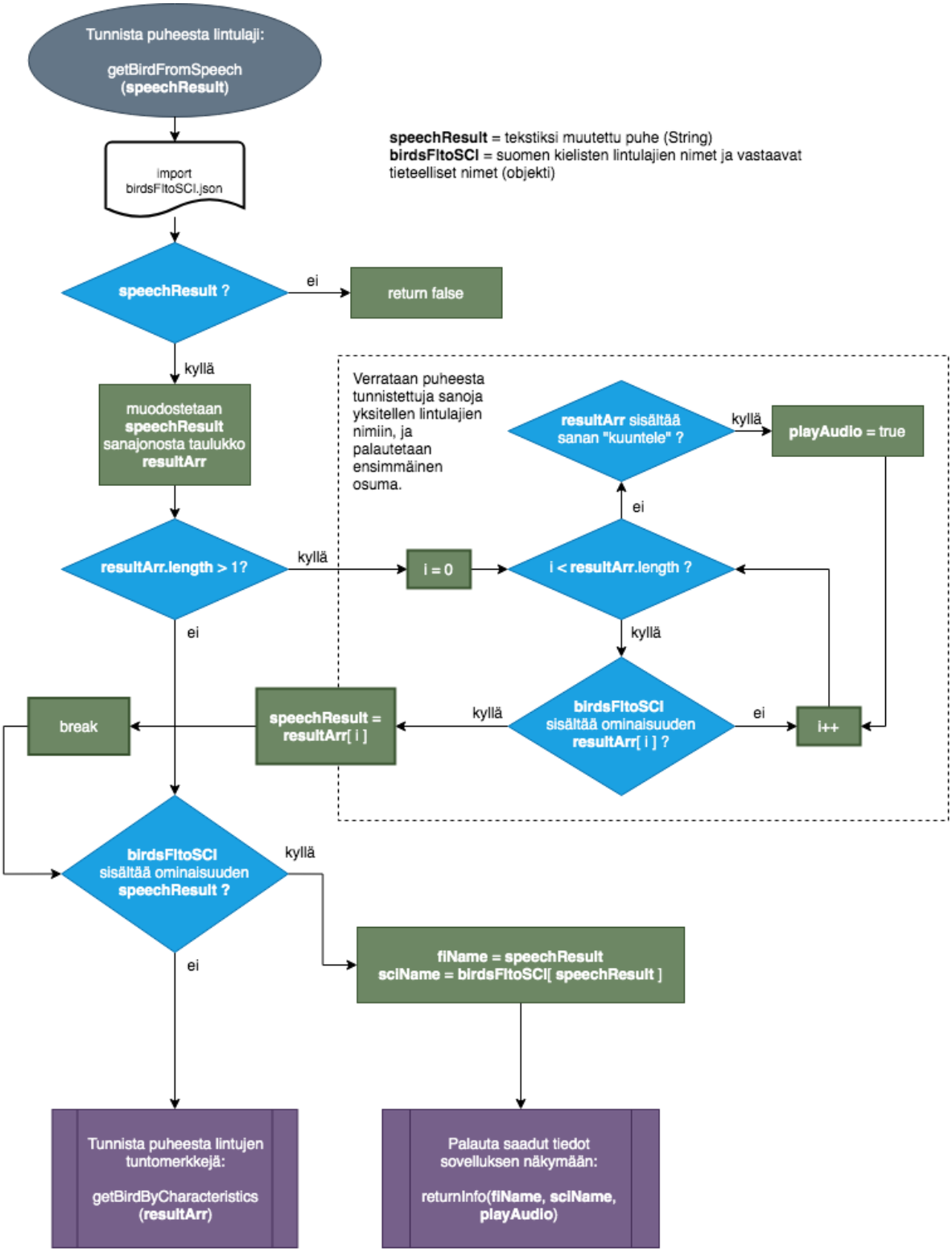
33. Web Speech API – Speech Recognition. Mozilla Wiki. Saatavissa: https://wiki.mozilla.org/Web_Speech_API_-_Speech_Recognition#Are_you_adding_voice_commands_to_Firefox.3F. Hakupäivä 2.4.2020.

34. Lintuatlas aineisto avoimena datana. Suomen Lintuatlas. Saatavissa: http://atlas3.lintuatlas.fi/taustaa/kaytto. Hakupäivä 20.3.2020

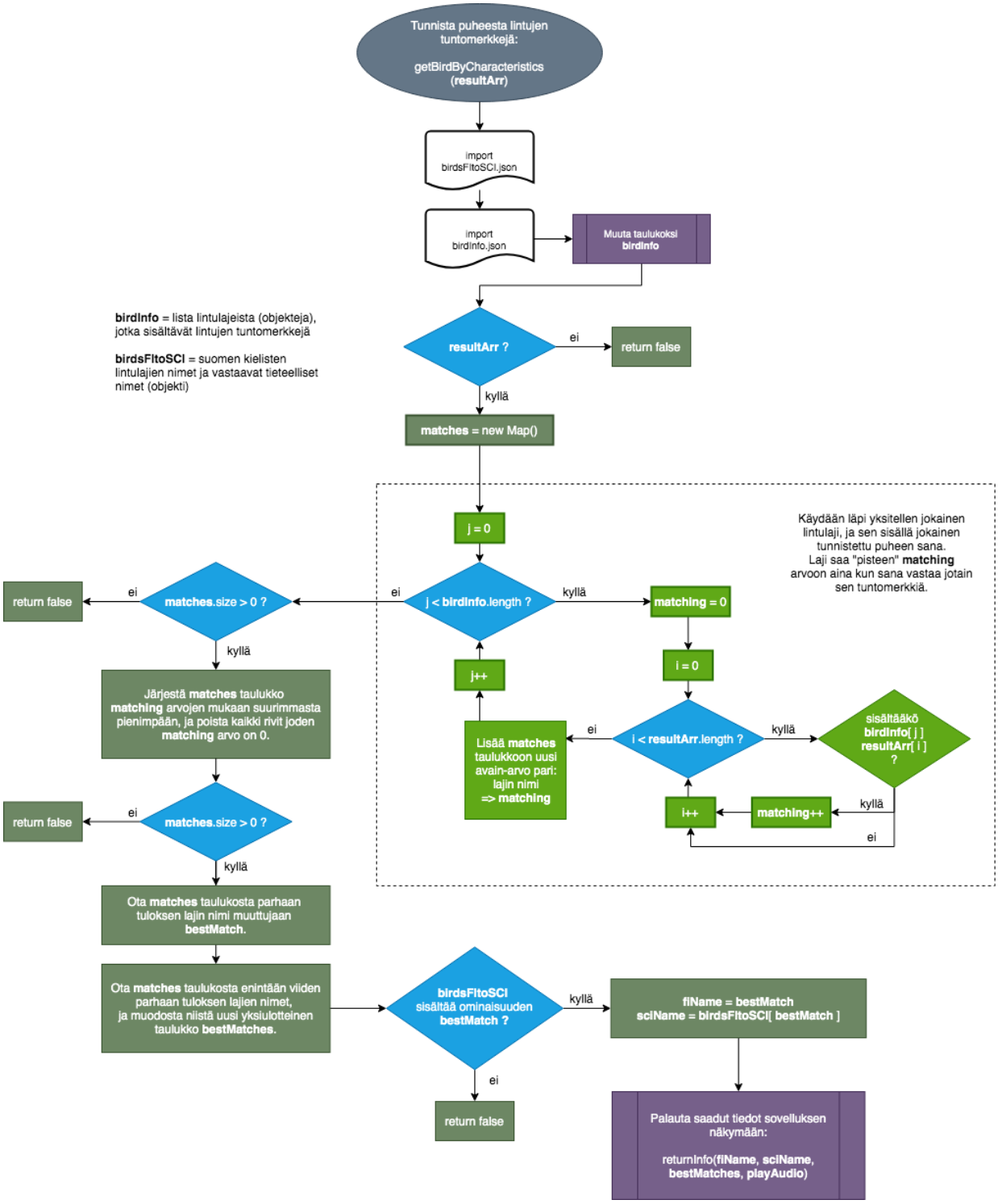
35. Issue: SpeechSynthesis resume() and pause(). Github MDN/browser-compat-data. Saatavissa: https://github.com/mdn/browser-compat-data/issues/4500. Hakupäivä 10.5.2020.

36. Sharing bird sounds from around the world. Xeno-canto. Saatavissa: https://www.xeno-canto.org/. Hakupäivä 14.4.2020.

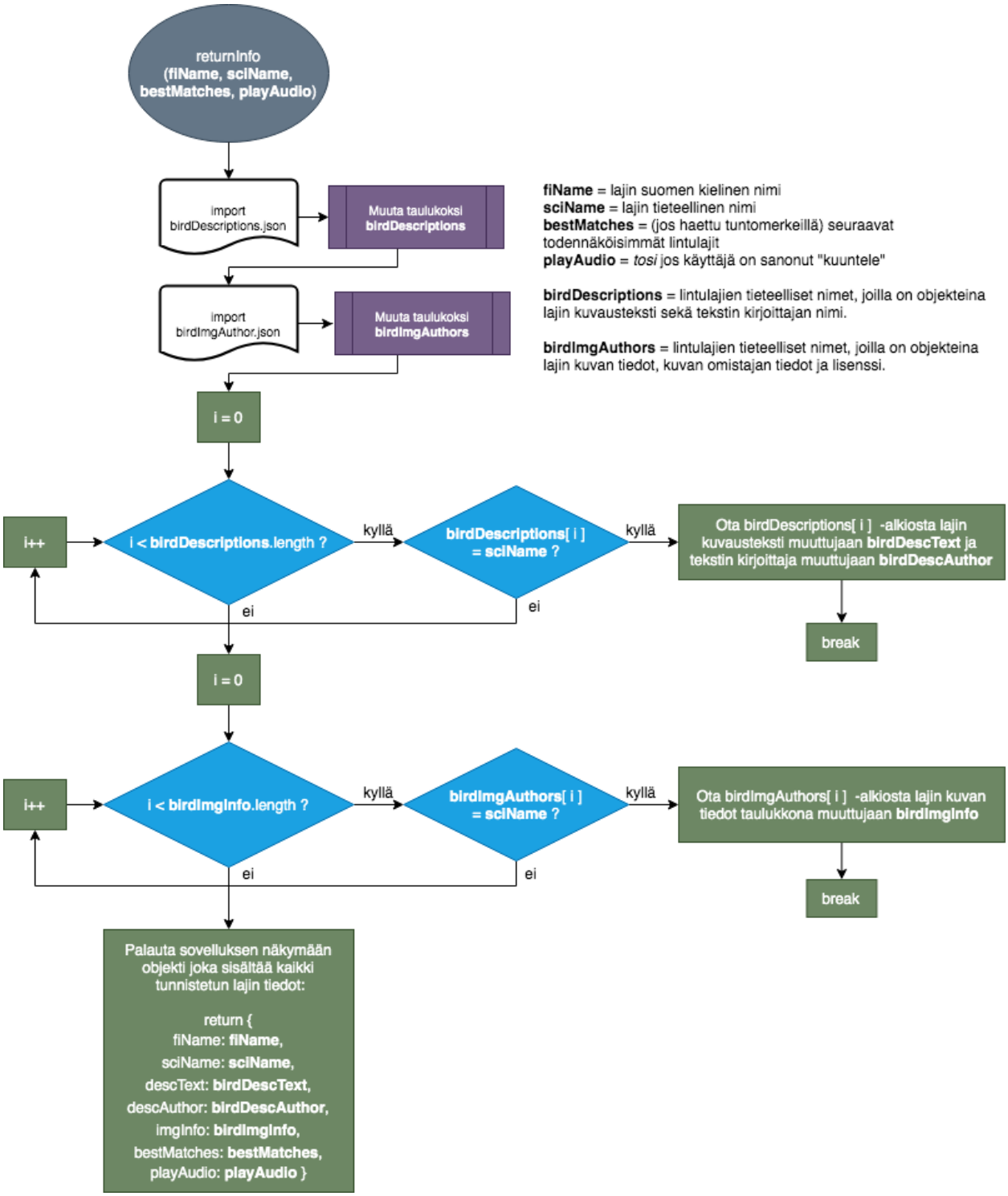
Funktio, joka etsii lintulajin tekstiksi muutetusta puheesta suomenkielisten lintulajien listaan vertaamalla.

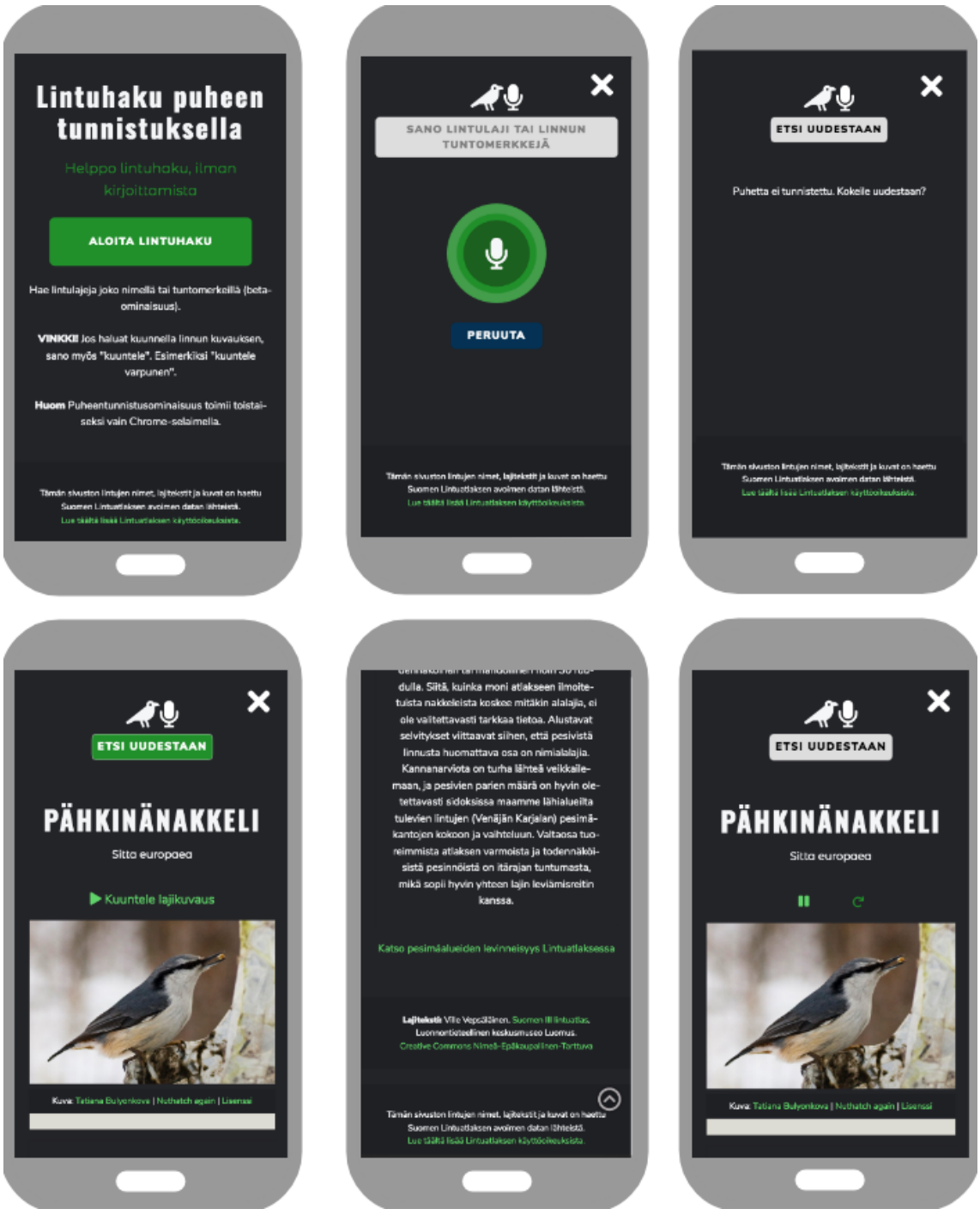


Funktio *getBirdByCharacteristics*, joka etsii lintulajin tekstiksi muutetusta puheesta vertaamalla käyttäjän antamia hakusanoja lintulajien tuntomerkkien listaan.



Funktio *returnInfo*, joka hakee annetun lintulajin tieteellisen nimen perusteella sen lajikuvausteks-tin, kuvan tiedot ja näiden tekijänoikeustiedot sovelluksen JSON-tiedostoista sekä palauttaa kaikki linnun tiedot funktiota kutsuneelle komponentille.




Lintuhaku puheen tunnistuksella
Helppo lintuhaku, ilman kirjoittamista
ALOITA LINTUHAKU
Hae lintulajeja joko nimellä tai tuntomerkeillä (beta-ominaisuus).
VINKKI! Jos haluat kuunnella linnun kuvauksen, sano myös "kuuntele". Esimerkiksi "kuuntele varpunen".
Huom Puheentunnistusominaisuus toimii toistaiseksi vain Chrome-selaimella.
SANO LINTULAJI TAI LINNUN TUNTOMERKKEJÄ
PERUUTA
ETSI UUDESTAAN
Puhetta ei tunnistettu. Kokeile uudestaan?
ETSI UUDESTAAN
PÄHKINÄNAKKELI
Sitta europaea
Kuuntele lajikuvaus
Katso pesimäalueiden levinneisyys Lintuatlaksessa
ETSI UUDESTAAN
PÄHKINÄNAKKELI
Sitta europaea