

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Laura Leisberg 179330IAIB

**BRAUSERI LAIENDUSE LOOMINE
EESTIKEELSE TEKSTI DIKTEERIMISEKS
KÕNETUVASTUSE ABIL**

Bakalaureusetöö

Juhendaja: Tanel Alumäe
PhD

Tallinn 2020

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Laura Leisberg

11.05.2020

Annotatsioon

Käesoleva töö eesmärgiks on luua veebilehitsejale Google Chrome'i laiendus eestikeelse teksti dikteerimiseks kõnetuvastuse abil, millega kirjutada eelkõige Gmailis emaile. Töö käigus sai laiendus rakendust ka teistel lehtedel, kuhu on võimalik teksti sisestada kõnetuvastusega. Loodav laiendus kasutab kõne tekstiks transkribeerimiseks TTÜ keeletehnoloogia laboris välja töötatud serveripõhist kõnetuvastustarkvara.

Laienduse ülesehitamisel lähtuti Chrome'i poolt pakutavast dokumentatsioonist. Programmeerimiskeeleks kujunes JavaScript ning kasutust sai ka Vue raamistik kasutajaliidese loomisel. Töö esimeses pooles uuritakse olemasolevaid lahendusi, kõnetuvastuse rakendusi ja kasutajaliidese disaini. Teises pooles kirjeldatakse pikemalt, kuidas laiendus valmis ning analüüsitakse valminud laienduse tagasiside tulemusi.

Bakalaureusetöö tulemuseks valmis laiendus nimega „Kõnetuvastus“, mida on võimalik allalaadida igal huvilisel Chrome'i veebipoest (veebipoe link saadaval Lisa 2 lehel).

Lõputöö on kirjutatud eest keeles ning sisaldab teksti 22 leheküljel, 8 peatükki, 14 joonist, 0 tabelit.

Abstract

Browser extension for dictating Estonian text using speech recognition

The purpose of this thesis is to create a Google Chrome extension for dictating Estonian text using speech recognition, mainly to be used for writing emails in Gmail. During the development of the extension it was found to be available to be used on other webpages, where it is possible to insert text. The extension is going to use server-based speech recognition software developed by The Laboratory of Language Technology at TTÜ.

The extension was built by following the instructions of the documentation provided by Chrome. The programming language was JavaScript and the Vue framework was used for the user interface. The first part of the thesis analyzes existing solutions, speech recognition usages and the design of speech recognition. The second part explains how the extension was made and an analysis of the extension feedback from test subjects.

The result of the thesis is a working speech-to-text extension named „Kõnetuvastus“, that is publicly available to download from the Chrome web store for everyone interested. (link available in Lisa 2).

The thesis is in Estonian and contains 22 pages of text, 8 chapters, 14 figures, 0 tables.

Lühendite ja mõistete sõnastik

<code><div></code>	Silt määratleb HTML-dokumendis jaotuse või jaotise
<code><script></code>	Silti kasutatakse käivitatava koodi manustamiseks või viitamiseks, tavaliselt on selleks JavaScript koodi
Android	Tarkvarakomplekt elektroonikaseadmetele, mis hõlmab operatsioonisüsteemi, vahetarkvara ja peamisi rakendusi
API	<i>Application Programming Interface</i> , Rakendusliides
CSP	<i>Content Security Policy</i> , sisu turvapoliitika
DOM	<i>Document Object Model</i> , Dokumendi objektimudel
GUI	<i>Graphical user interface</i> , Graafiline kasutajaliides
HTTP	<i>Hypertext Transfer Protocol</i> , hüpertexti edastusprotokoll
iOS	Apple toodete operatsioonisüsteem mobiilsetele seadmetele
JSON	<i>JavaScript Object Notation</i> , JavaScripti objektide notatsioon
WebSocket	Arvutisideprotokoll, mis pakub täisduplekseid suhtluskanaleid üle TCP-ühenduse kaudu
XSS	<i>Cross-site scripting</i> , murdskriptimine

Sisukord

1 Sissejuhatus	8
2 Taust ja olemasolevad lahendused	9
2.1 Dikteeri veebileht	10
2.2 Kõnele.....	11
2.3 Voice Recognition	12
2.4 Speech Recognition Anywhere	12
3 Kõnetuvastuse rakendusi ja kasutajaliides	13
3.1 Kõnetuvastuse tüüpilised kasutusvaldkonnad	14
3.2 Kõnetuvastuse kasutajaliidese disain.....	15
4 Tehnoloogiate valik	17
4.1 Chrome'i laienduse arhitektuur ja üldised nõudmised	17
5 Brauseri laienduse loomine	20
5.1 Skriptid	20
5.2 Kasutajaliides.....	22
6 Laienduse valideerimine.....	24
6.1 Laienduse kasutamine.....	25
6.2 Laienduse mugavus ja kujundus.....	26
7 Võimalikud edasiarendused.....	28
8 Kokkuvõte	29
Kasutatud kirjandus	30
Lisa 1 – Manifest.json fail	31
Lisa 2 – „Kõnetuvastus“ laienduse veebipoe link	32

Jooniste loetelu

Joonis 1. Google'i häälsisestamine	9
Joonis 2. Dikteeri veebilehe kõnesisendi kuvatõmmis.....	10
Joonis 3. Kõnele klaviatuuri kuvatõmmised	11
Joonis 4. Tüüpilise kõnetuvastus rakenduse arhitektuur [8]	14
Joonis 5. Microsoft Agent tegelased (vasakult: Peedy, Merlin, Genie and Robby) [10]	15
Joonis 6. Chrome'i laienduse arhitektuur [13]	18
Joonis 7. Loodava brauseri laienduse arhitektuur	20
Joonis 8. Hüpikmenüü valik „Dikteeri tekst antud tekstivälja“	21
Joonis 9. Kõnetuvastuse hüpikaken.....	22
Joonis 10. Kõnetuvastuse hüpikaken dikteeritud teksti ja animatsiooniga	23
Joonis 11. Laienduse kasutamise arusaadavus	25
Joonis 12. Tööriistariba peale laienduse installimist.....	25
Joonis 13. Laienduse kasutamise mugavus	26
Joonis 14. Laienduse kujunduse hinnang	27

1 Sissejuhatus

Kõnetuvastuse tehnoloogia kogub üha enam populaarsust üle maailma. Tuntuim kõnetuvastus tarkvara leidub nutitelefonides, mis võimaldab kasutajatel suhelda oma seadmetega ning anda seadmetele täitmiseks erinevaid ülesandeid. iOS nutitelefonides on kasutusel Siri, Android nutitelefonides Google Assistant, Windows Phone'ides Cortana [2]. Valdavalt on kõnetuvastus aga inglisekeelne ning väiksemate keelte tugi puudub, sealhulgas eestikeelene tugi, kuna arendamine konkreetsele keelele on ressursimahukas. Näiteks, Google plaanib luua kõnetuvastust kõikidele keeltele, millel on üle miljoni kõneleja, aga kahjuks jääb eesti keele kõnelejate arv (~900 000) sellele alla. Seetõttu on eestikeelse kõnetuvastustehnoloogia väljatöötamine eestlaste endi kätes [1].

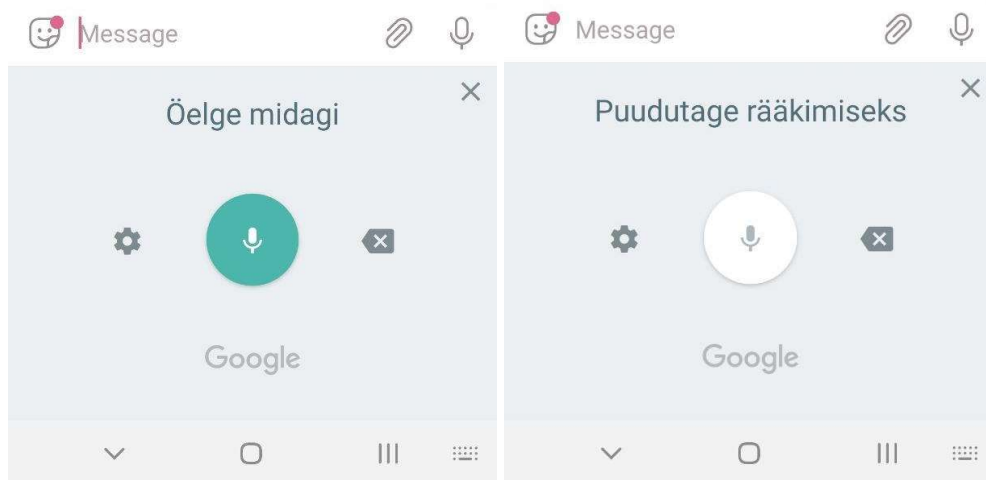
TTÜ keeletehnoloogia labor tegeleb eestikeelse kõnetuvastuse arendamisega. Nende poolt on väljatöötatud rakendus, mis saadab reaaliajaliselt kõne TTÜ keeletehnoloogia labori serverisse, kus sisend muudetakse tekstiks ning saadetakse brauserisse väljund tagasi [1]. Käesoleva bakalaureusetöö põhieesmärgiks on antud projekti kasutajakogemuse lihtsustamine arendades juurde Google Chrome'i laiendus.

Esialgul oli bakalaureusetöö eesmärgiks Chrome'i laiendus arendada ainult Google'i Gmailide kirjutamise jaoks. Töö käigus sai seda eesmärki laiendada igale veebilehele, kus on olemas tekstiväljad. Valminud laiendus „Kõnetuvastus“ on saadaval alla laadimiseks igale huvilisele Chrome'i veebipoest (vaata Lisa 2).

2 Taust ja olemasolevad lahendused

Hetkel on avalikult kättesaadavad eestikeelse kõnetuvastuse lahendused olemas androidi platvormil (rakendused „Kõnele“ ja „Arvutaja“) ja brauseris dikteerimise veebileht. Mõlemad kasutavad kõne tekstiks transkribeerimiseks TTÜ keeletehnoloogia laboris välja töötatud serveripõhist kõnetuvastustarkvara [2].

Teadagi ei toeta Google brauseris eestikeelset kõnetuvastust, aga Android telefonides on siiski võimalik klaviatuur vahetada Google'i häälsisestamise režiimi peale, mis võtab vastu eestikeelset teksti (joonis 1). Kahjuks pole nende lahendus nii täpne kui TTÜ keeletehnoloogia laboris välja töötatud kõnetuvastustarkvara.



Joonis 1. Google'i häälsisestamine

Samuti vaadeldakse selles peatükis populaarsemaid ingliskeelseid kõnetuvastus laiendusi, mida pakub Chrome'i veebipood.

2.1 Dikteeri veebileht

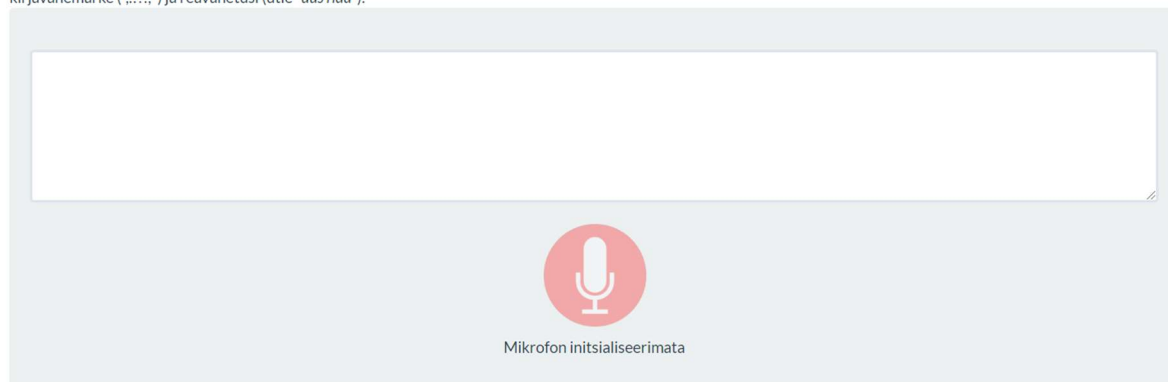
Hetkel on avalikult kättesaadav lahendus veebilehel <https://bark.phon.ioc.ee/dikteeri/>. Veebibrauser saadab reaalaajaliselt kõne TTÜ keeletehnoloogia labori serverisse, kus kõne muudetakse tekstiks ning saadetakse brauserisse tagasi. Enne kui tekst veebilehel kuvatakse, tuvastatakse dikteeritud tekstis kirjavahemärgid ja õigekeelsus reeglid [4].

Veebileht on kujunduse poolest minimalistlik, kergesti arusaadav (joonis 2). Lehel on suur tekstiväli suure nupuga. Antud nupp suudab kõnetuvastust käivitada, lõpetada või katkestada. Lisaks väljendab nupp kasutajale informatsiooni selle kohta, kas mikrofon on initsialiseeritud. Nupu lisafunktsiooniks on oskus häälsisendi järgi ennast animeerida, et anda kasutajale infot, et mikrofon kuulab teda.

Veebilehe kasutamise miinus on siiski selle ebamugavus. Peale dikteerimist on vaja kopeerida ja kleepida dikteeritud tekst soovitud tekstiväljadele. Nii peab kasutaja palju rohkem hiirt kasutama.

Rakendus valmis riikliku programmi Eesti Keeletehnoloogia (2011-2017) projekti Kõnetuvastus raames TTÜ keeletehnoloogia laboris. Rakenduse loojad on Tanel Alumäe (serveritarkvara, kõnetuvastus) ja Kaarel Kaljurand (serveriga suhtlev JavaScripti moodul) [3].

Parima tulemuse saamiseks tuleks dikteerida selge häälega ja mõõdukas tempos (umbes nagu raadiodiktor). Sõnade vahel pause ei pea tegema. Dikteerida saab ka kirjavahemärke (",!?:") ja reavahetusi (ütle "uus rida").



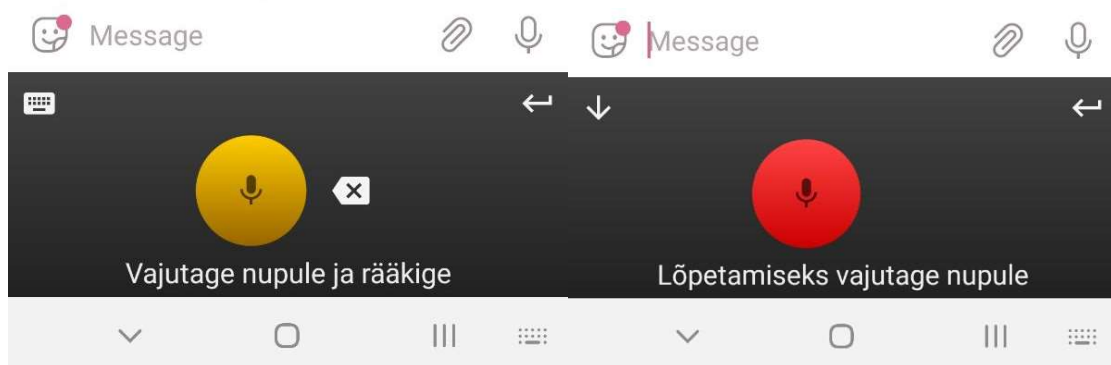
Joonis 2. Dikteeri veebilehe kõnesisendi kuvatõmmis.

2.2 Kõnele

Kõnele on kõnetuvastusteenus Androidi rakendustele, mille töötas välja Kaarel Kaljurand. See on avalikult kättesaadav kõigile Google Play poes. Sellel on 213 kasutajat ja kesmine hinnang on 4/5. Kõnele võimaldab saata emaile, sooritada infootsinguid, kirjutada märkmeid ja anda käske kõne abil [5].

Kõnele on samuti disaini poolest minimalistlik (joonis 3). See võtab ekraanil vähem ruumi ja on kasutajale arusaadav. Rakenduse klaviatuur koosneb ainult ühest nupust, mis võimaldab kõnetuvastust käivitada, lõpetada ja katkestada ning nupu animatsioon reageerib sisendkõnele. Kõnele klaviatuur toetab ka järgmisi operatsioone[5]:

- 1) svaip vasakule/paremale muudab kursori asukohta
- 2) svaipimine pärast pikka vajutust valib teksti
- 3) topeltvajutus lisab tühiku
- 4) lühike vajutus klaviatuuriikoonile lülitab eelmisele klaviatuurile
- 5) pikk vajutus klaviatuuriikoonile lülitab järgmisele klaviatuurile
- 6) vajutus otsinguikoonile käivitab otsingu (ainult otsingureal)



Joonis 3. Kõnele klaviatuuri kuvatõmmised

2.3 Voice Recognition

Kõige populaarsem Google Chrome'i ingliskeelne kõnetuvastuse laiendus on *Voice Recognition*, mis kasutab Chrome'i enda kõnetuvastuse API't. Sellel on üle 400 000 kasutaja ning keskmine hinnanguga on 3.2/5. Laiendus on tasuta kasutamiseks [6].

Funktionaalsuse poolest on see väga sarnane eelnevalt vaadeldud Dikteeri veebilehele. See laiendus ei paku võimalust sisestada otse tekstiväljadele kõnesisendit, vaid laienduse avades avatase eraldi veebileht, kus võetakse kõnesisend vastu. Veebileht on saadaval <https://dictation.io> [6].

2.4 Speech Recognition Anywhere

Speech Recognition Anywhere laiendus on palju kasulikum ning palju sarnasem lõputöö eesmärgile. See laiendus kasutab samuti Chrome'i enda kõnetuvastuse API't. Sellel on üle 10 000 kasutaja ning keskmine hinnang on 4/5. See laiendus on tasuline ja maksab 29.99€. Saadaval on tasuta prooviversioon [7].

Antud laiendus on väga kasutajasõbralik. Suureks plussiks selle laienduse puhul on see, et kõnet pole vaja kopeerida ja kleepida teksti sisestusväljadele. Räägitava kõnega on võimalik häälkäsklustega veebilehti juhtida ilma ühegi hiire või klaviatuuri nupu vajutuseta. Näiteks on võimalik häälkäsklustega linke avada [7].

Laienduse tööle panemiseks on vaja vajutada brauseris laienduste real rakenduse ikooni, mis avab laienduse pealehe, kus on tekstiväli, kuhu hakkavad dikteeritud sõnad ilmnema. Pealeht peab olema laienduse töötamise ajal kogu aeg vahelehel lahti. Kui pealeht on avatud saab kasutaja hakata kasutama ka häälkäsklusi. Kui kasutaja palub pealehelt minna uuele lehele hakkavad dikteeritud sõnad ilmnema uuel veebilehele kollastes kastides. Laiendus suudab automaatselt tuvastada tekstivälju ning paigutada kollaseid kaste selle lähedusse, et kasutaja saaks aru, kuhu dikteeritud teksti hakatakse sisestama [7].

3 Kõnetuvastuse rakendusi ja kasutajaliides

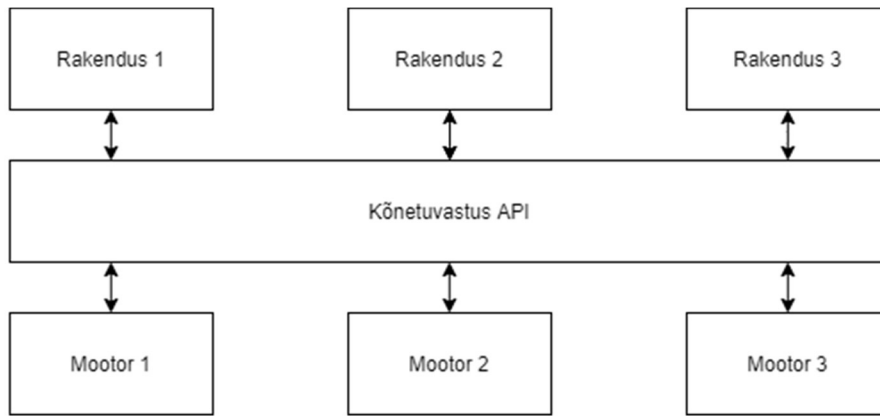
Kõnetuvastuse tehnoloogia abil leitakse automaatselt sõnad ja laused, mis kõige paremini vastavad sisendiks olevale inimese kõnele. Taoline tehnoloogia võimaldab luua mitmesuguseid rakendusi, kus kõnetuvastuse väljund võib olla lõpptulemuseks või sisendiks edasisele töötlusele [1].

Kõnetuvastuse rakendusi arendades tuleb läbimõelda, mis on kasutajaliidese ja rakenduse olemus. Enne arendama hakkamist on mõistlik veenduda, et loodav liides vastab sellele, kuidas kasutajad soovivad ülesannet täita [8].

Kõnetuvastuse rakenduste puhul on väljakutseks see, et kõne tuvastamine ja mõistmine pole ideaalne. Seetõttu on kriitilise tähtsusega see, et rakendused kasutaksid vajalikke vigade käsitlemise tehnikaid, et vigade mõju minimeerida. Rakenduste arendajad peavad mõistma põhjalikult kõnetehnoloogiate tugevaid ja nõrku külgi ning leidma sobiva koha kõnetehnoloogia tõhusaks kasutamiseks [8].

Traditsiooniline kõnetuvastus rakendus koosneb kolmest põhikomponendist nagu joonisel 3 illustreeritud [8]:

1. Mootorid, mis on kõnetuvastus süsteemid
2. API, mis suhtleb vahelülina mootorite ja rakendustega
3. Rakendused, mis kasutavad sama API't kõnetuvastus mootoriga suhtlemiseks



Joonis 4. Tüüpilise kõnetuvastus rakenduse arhitektuur[8]

3.1 Kõnetuvastuse tüüpilised kasutusvaldkonnad

Kõnetuvastuse rakendused saab liigitada kolme rühma, mis vajavad oma rühmale iseloomuliku kasutajaliidest [8]:

1. Kontor: töölaua rakendused nagu näiteks Microsoft Windows ja Microsoft Office.
2. Kodu: televiisorid ja köögimööbel. Iseloomulik on klaviatuuri või hiire puudumine ning traditsioonilist GUI'd pole võimalik kasutada.
3. Mobiil: mobiiltelefonid ja autosalongi elektroonika. Iseloomulik on ekraani väike füüsiline suurus, käed ja silmad on rohkem hõivatud kasutusajal. Traditsioonilist GUI'd võimalik kasutada.

Üks kõnetuvastuse varasemaid prototüüpe on käsuta ja juhi rakendused, mida kasutatakse peamiselt navigeerimiseks operatsioonsüsteemi liidest ja nende all töötavate rakenduste kaudu [8]. Näiteks, Microsoft Agent tehnoloogia lõi aluse loomulikumatel viisidel, kuidas inimesed saaksid oma arvutitega suhelda. Microsoft Agent on tarkvarateenuste komplekt, mis võimaldab arendajatel integreerida interaktiivseid animeeritud tegelasi oma rakendustesse ja veebilehtedesse. Näiteks, joonisel 5 on näha nelja sellist tegelast. Ekraanile ilmuvad tegelased saavad rääkida ja isegi häälkäsklusi vastu võtta. Kahjuks on Microsoft Agent vananenud alates Windows 7 versioonist [9].



Joonis 5. Microsoft Agent tegelased (vasakult: Peedy, Merlin, Genie and Robby) [10]

Dikteerimine on inimeste jaoks atraktiivne, kuna üldjuhul on rääkimine kiirem kui tippimine, eriti Ida-Aasia keeltes, näiteks hiina keeles. Dikteerimise rakenduste puhul on väga oluline, et kõne tekstiks teisendamisel näeb kasutaja reaalselt kirjutatud formaadis oma dikteeritud kõne. Oluline on ka, et kirjavahemärgid on lisatud automaatselt ning suurtähtede ja homonüümide eristamine on olemas. Kuupäevad ja kellaajad on vaja teisendada tavapäraselt kasutatavasse vormi [8].

Dikteerimise puhul tuleb tihti ette argumente, et klaviatuuri ja hiire kasutamine on mugavam kui dikteerimine. Sellised argumendid peegeldavad tõsiasja, et kõnetuvastuse ja dikteerimise tehnoloogia pole veel nii väljaarenenud. Seetõttu on vaja teha kasutajatele vigade parandamise võimalikult lihtsaks. Samuti peab kasutajal lubama süsteemi koolitada, et see saaks kohaneda erinevate aktsentide ja keskkonnatingimustega [8].

Ligikaudu 10–15 protsendil elanikkonnast on puue, mis mõjutab nende võimet arvutit tõhusalt kasutada. Puuetega inimestel on rohkem kasutatavuse probleeme kui neil, kellel pole. Lahendus oleks selline, et inimesed, kes ei saa klaviatuuri või hiirt tõhusalt kasutada, saavad arvuti juhtimiseks kasutada kõnekäsklusi ja teksti sisestamiseks dikteerimis-tarkvara [8].

3.2 Kõnetuvastuse kasutajaliidese disain

Kasutajaliidese kujundamisel pole seni välja kujunenud ühiseid reegleid, aga praktikud on postuleerinud üldised põhimõtteid, mida arendajad saaksid järgida. Kõige olulisemad kriteeriumid, mida jälgida tähtsuse järjekorras on [8]:

1. Tõhusus
2. Praktilisus
3. Kasutajate rahulolu

Ennekõike on tähtis see, et loodav kasutajaliides töötab ning kasutaja saab soovitud töö tehud. Peale seda on oluline, et kasutajaliides täidaks oma ülesanded võimalikult produktiivselt ning kui see on ka rahuldatud saab keskenduda kasutaja rahulolul parandamisele, mida mõõdetakse küsimustike ja kasutaja tagasiside abil [8].

4 Tehnoloogiate valik

Suures osas kasutati töö tegemisel JavaScripti, kuna Chrome'i laienduse loomisel on spetsiifilised JavaScripti failid olulised. Kasutati Vue raamistikku laienduse kasutajaliidese loomiseks. Lisaks kasutati webpacki, et pakkida kokku kõik vajalikud failid.

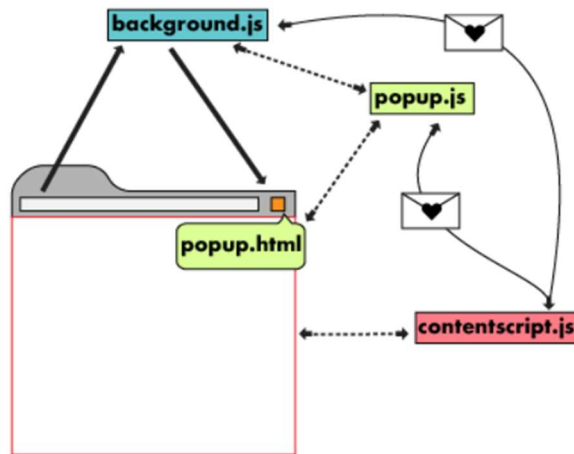
4.1 Chrome'i laienduse arhitektuur ja üldised nõudmised

Brauseri laiendus on väike rakendus, mida saab juurde lisada oma brauserile, näiteks Google Chrome'i brauserile või Firefoxile. Need väikesed rakendused võimaldavad meil muuta või/ja parandada brauseri funktsionaalsust [12].

Chrome'i laienduste puhul on üks miinus, et Google ei luba laienditel töötada spetsiaalsetel *chrome://* lehtedel ühilduvusprobleemide minimeerimiseks, mis võivad tekkida kui uued brauseri versioonid tulevad vana versioonide asemele. Seetõttu ei saa lõputöö raames valminud laienduse kasutajad oma hääle abil neil lehtedel laiendust kasutada. Kasutajaliides küll avaneb, aga teksti ei sisestata nendel lehtedel tekstiväljadele. See on Google'i kehtestatud piirang, mitte laienduse piirang [11].

Antud lõputöös keskenduti just Chrome'i brauseri laienduse arendamisele. Chrome'i laienduse arhitektuur oleneb rakenduse funktsionaalsusest, aga Chrome'i laienduse südamik koosneb kahest failist. *Manifest* failist, mis on JSON formaadis informatiivne fail, milles on näha laienduse versioon, ressursid ja load, mida laiendusel on vaja töötamiseks. Teine fail on *background.js* skript, mis on laienduse sündmuste käsitleja. Skript kuulab brauseris tehtavaid sündmusi, näiteks mingit nupu vajutust, ning täidab talle ette antud loogika [13].

Kui laienduse funktsionaalsus on suurem, siis lisanduvad arhitektuuri ka kasutajaliidese elemendid (*popup.js*, *popup.html*) ja sisuskriptid (*contentscript.js*). Joonisel 6 on sellise arhitektuuri illustreeriv joonis. Sisuskriptid saavad lugeda ja saavad muuta veebilehe DOMi, mida brauser hetkel külastab. Sisuskriptid saavad veel suhelda ja jagada sõnumeid eelnevalt mainitud *background.js* failiga ning kasutajaliidese skripti failiga [13].



Joonis 6. Chrome'i laienduse arhitektuur [13]

Kui on tahtmine oma laiendus avalikult kättesaadavaks teha on Chrome'i veebipoodi kehtestatud karmid reeglid, et kaitsta murdskriptimine, XSS, eest. Tänu sellele on Chrome võtnud vastu CSP kontseptsiooni, mis muudab laiendused vaikimisi turvalisemaks ja annab arendajatele võimaluse luua ja jõustada reegleid, mis reguleerivad seda tüüpi sisu, mida nende laiendused saavad laadida ja täita. Üldiselt töötab CSP laienduste laaditud või teostatud ressursside blokeerimise ja lubamise loendina [14].

Veebis on taolised reeglid kehtestatud HTTP päises või meta elemendis. Laienduse tegemisel me määrame reeglis *manifest.json* failis. Kuid võimalik on ka vaikimisi määrata CSP reeglid. Selleks on vaja määrata *manifest_version* versioon 2, mis automaatselt piirab laienduse arendamisel järgneva kasutust [14].

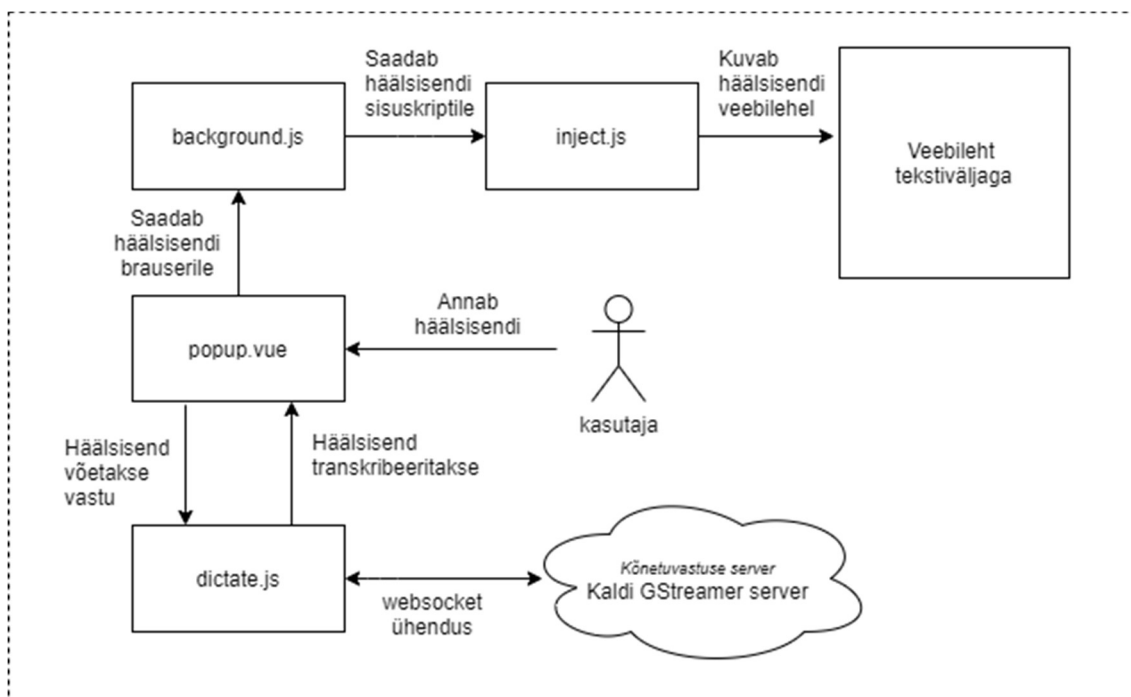
JavaScripti `eval()` funtsioon on rangelt keelatud, kuna see funktsioon käivitab koodi, mis on funtsiooni kutsuja õigustega [14]. Nimelt kui käivitada `eval()` sõnega, mida saab kolmas pahatahtlik osapool mõjutada, võib juhtuda, et laiendus hakkab jooksutama klientide arvutis pahatahlikku koodi. Sealhulgas on kolmandal osapoolel võimalik näha koodi skoopi, kust `eval()` funktsioon välja kutsuti, mis võib põhjustada rünnakuid. Sarnaselt on ka keelatud `Function` objekti kasutamine, mis on sarnaste turvariskidega nagu `eval()` funktsioonid [15].

Lisaks on keelatud tekstisisesed skriptid. Näiteks `<button onclick="...">` või tekstisisesed `<script>` sildid [14]. Selle nõudmise sai ära lahendatud antud projektis Vue raamistiku kasutamisega. Nimelt kui kasutada käitusfaasis kompileerimiseks webpacki ja vue-loaderit, siis Vue mallid eelkompileeritakse *render* funktsioonideks, mis töötavad ideaalselt CSP keskkondades [16].

Viimane nõue on see, et teegid ja skriptid tohivad olla ainult laaditud laienduse enda paketist [14]. Näiteks antud projektis on kasutusel Bootstrap, mis on eraldi salvestatud *bootstrap.min.css* failina. Samuti on eraldi failidena kõik teised kasutatavad teegid.

5 Brauseri laienduse loomine

Arhitektuur on loodaval laiendusel minimalistlik. Kasutusel on skript `background.js`, siseskript `inject.js`, kasutajaliidese `.vue` fail, kõnetuvastuse teek `dictate.js` ning kõnetuvastuse jaoks Kaldi GStreamer server, millel on WebSocket ühendus `dictate.js` klassiga, mida on võimalik näha joonisel 7.



Joonis 7. Loodava brauseri laienduse arhitektuur

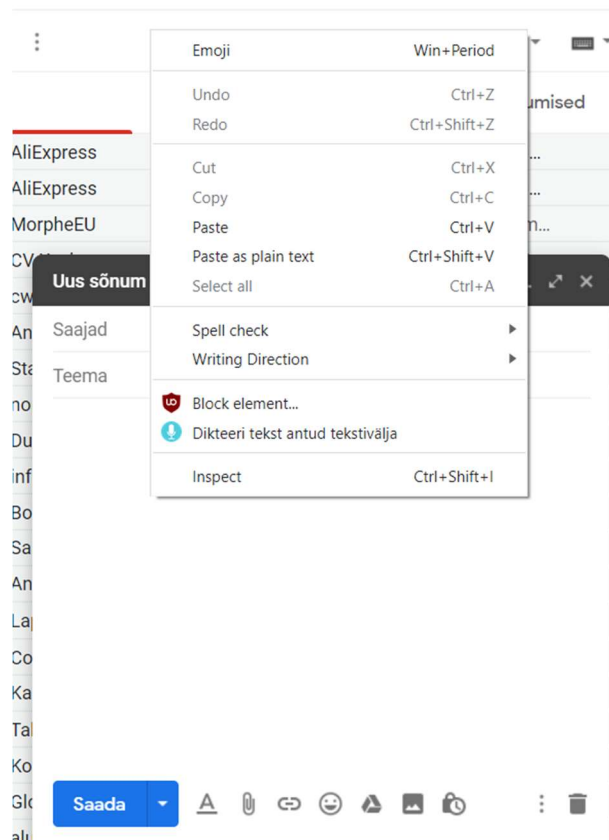
5.1 Skriptid

Eelnevas peatükis, tutvuti kuidas töötavad skriptid Chrome'i laienduse kasutamisel. Teame, et `background.js` kuulab brauseris sündmusi ning siseskript `inject.js` saab lugeda ja muuta veebilehe DOMi. Nende skriptidega paneme paika, kuidas laiendus hakkab funktsioneerima.

Laienduse laadimisel brauserisse registreeritakse veebilehe hüpikmenüüsse uus valik nimega „Dikteeri tekst antud tekstivälja“ (joonis 8). Hüpikmenüü uus valik tekib, aga ainult juhul kui paremklikitud element on kasutuskonteksti tüüpi *editable*. `Background.js`

kuulab lehel paremakliki sündmuse, kui sündmus on tuvastatud, siis background.js võtab brauserilt vastu veebilehe id, kus klikk tehti. Id on vajalik, et laadida inject.js sisuskript veebilehele ja saata dikteeritud tekst veebilehele. Kui kasutaja vajutab hüpikmenüüs loodud valikut, siis loob background.js uue hüpikakna arvuti ekraani vasakusse alumisse nurka. Avanenud hüpikaken on laienduse kasutajaliides. Kasutajaliidese kood on kõik popup.vue failis.

Loogiline oleks olnud inject.js fail laadida veebilehele enne kui paremklikk sooritatakse. Sellise lähenemise probleemiks on see, et Chrome'i käest tuleb seljuhl rohkem õiguseid küsida. Seetõttu laetakse inject.js veebilehele peale paremakliki valikut. Laiendusel on antud lahenduse puhul vaja Chrome'ilt küsida ainult 2 õigust: *contextMenus* – hüpikmenüü redigeerimise õigus ja *activeTab* – aktiivse vahelehe redigeerimise õigus. Õigused (*permissions*), määrati manifest.json failis, mida on võimalik näha Lisa 1 lehel.

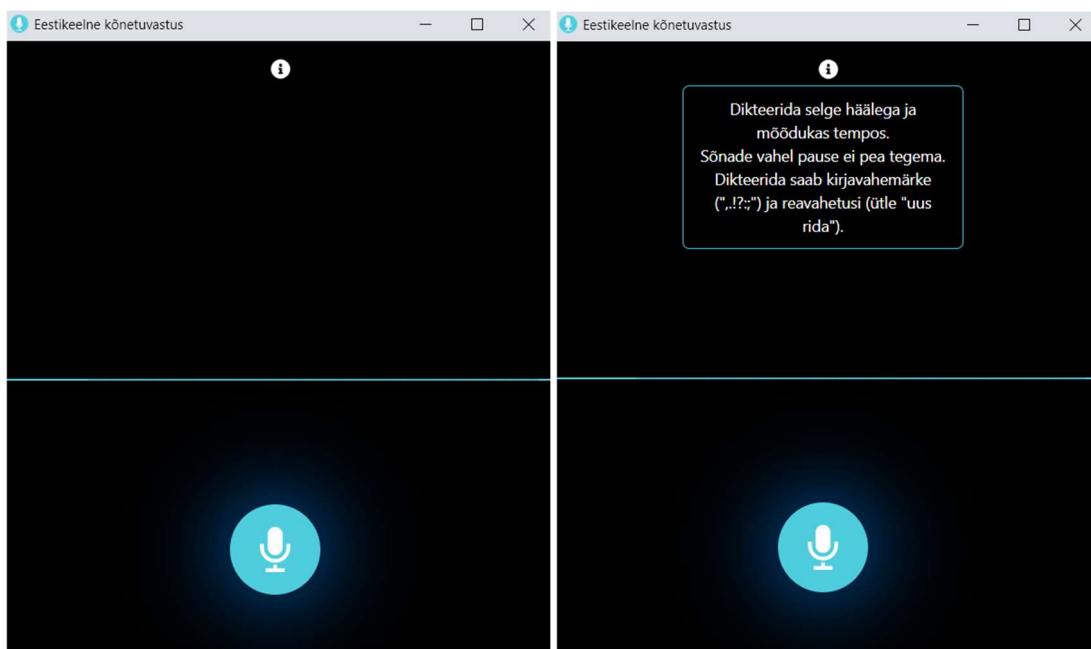


Joonis 8. Hüpikmenüü valik „Dikteeri tekst antud tekstivälja“

Kui Hüpikaken on võtnud kasutaja häälsisendi vastu ja hüpikaken on sulgenud saadab popup.vue fail transkribeeritud teksti background.js faili, mis omakorda saadab selle inject.js faili. Inject.js failis kontrollitakse kas veebilehel aktiivne element, kuhu teksti tahetakse sisestada on tekstiväli, sisend või *<div>* element. Sõltuvalt tüübist sisestab sisuskript transkribeeritud teksti veebilehe DOMi.

5.2 Kasutajaliides

Kasutajale avaneb peale paremakliki teostamist arvuti ekraanile vasakusse alumisse serva hüpikaken, millel on ainult üks vaade (joonis 9). See koosneb 2 nupust, info nupp ja sinine dikteerimise nupp. Kui info nupu peal hiirega hõljuda ilmub nupu alla kast, milles on juhised, kuidas kõige paremini häälsisendit anda. Lisaks on häälsisendit animeeriv sinine joon akna keskel. Dikteeritud tekst hakkab ilmuma vahetult peale mikrofoni nupu vajutust sinise joone ja info nupu vahele.



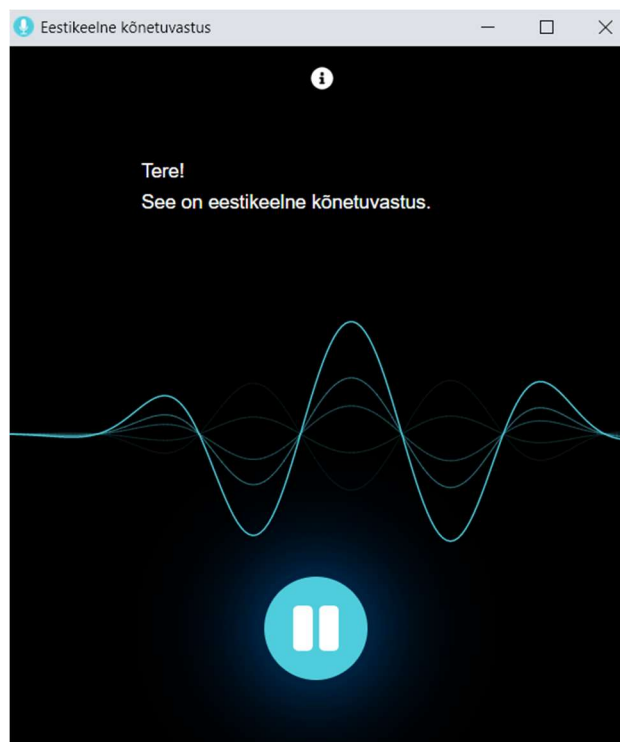
Joonis 9. Kõnetuvastuse hüpikaken

Kui hüpikaken avatakse, siis kasutajaliidese popup.vue failis luuakse dictate.js teegi objekt, millele on võimalik anda enda konfiguratsioon. Objekti konfiguratsiooni sai kirjutada õigekeelsus reegleid. Näiteks tuvastada suure algustähe reegleid ja

kirjavahemärk. Kahjuks ei saanud lisada reegleid, et ära tunda kirjamärke „@“ ja „com“, et emailide kirjutamine oleks mugavam.

Kasutaja kõne ei hakata kohe kuulama kui dictate object loodi. Esmalt küsitakse kasutajalt luba mikrofoni kasutada. Kui mikrofoni kasutuse luba on saadud saab kasutaja dikteerimise alustamiseks vajutada sinist dikteerimise nuppu. Kui nuppu vajutatakse, siis hakkab popup.vue failis loodud dictate.js teegi objekt kuulama hääsisendit. Dictate.js suhtleb WebSocket ühenduse abil reaalajaliselt Kaldi GStreamer serveriga, mis teostab kõnetuvastust. Serverist tagasisaadud tekstis tuvastatakse kirjavahemärgid ja õigekeelsus reeglid nii nagu ette antud dictate object konfiguratsiooni reeglites kirjas.

Samal ajal kui kasutaja dikteerib teksti on võimalik kasutajal näha, kuidas tema antud häälsisendile reageerib sinine joon (Joonis 10). Nimelt hakkab joon lainetama sõltuvalt hääletugevusele. Lainetuse jaoks kasutati vabavaralist SiriWave teeki.



Joonis 10. Kõnetuvastuse hüpinkaken dikteeritud teksti ja animatsiooniga

6 Laienduse valideerimine

Valideerimise katseisikuteks valiti inimesed, kelle emakeel on eesti keel. Testijate vanus oli 15-50 ning kokku oli katseisikuid 14. Valideerimine viidi läbi kasutades Google'i Vormid keskkonda. Katseisikutel saadeti küsitlus, milles paluti Chrome'i veebipoest allalaadida laiendus „Kõnetuvastus“ ning proovida häälsisestusega kirjutada emaili jaoks teksti ning proovida erinevatel veebilehtedel tekstiväljadel seda sama. Peale katsetuse lõppu paluti katseisikutel küsitluses vastata küsimustele:

1. 1-5 skaalal kui mugav on laiendust kasutada?
2. 1-5 skaalal kas laienduse kasutamine on arusaadav?
3. 1-5 skaalal kuidas hindad laienduse kujundust?
4. Kas sa kasutaksid seda laiendust edasipidi?
5. Mis arvamused teil laiendusest tekkisid? Mis meeldis mis ei meeldinud?
6. Kuidas laiendust saaks edasi arendada? Kas midagi peaks eemaldama või juurde lisama?

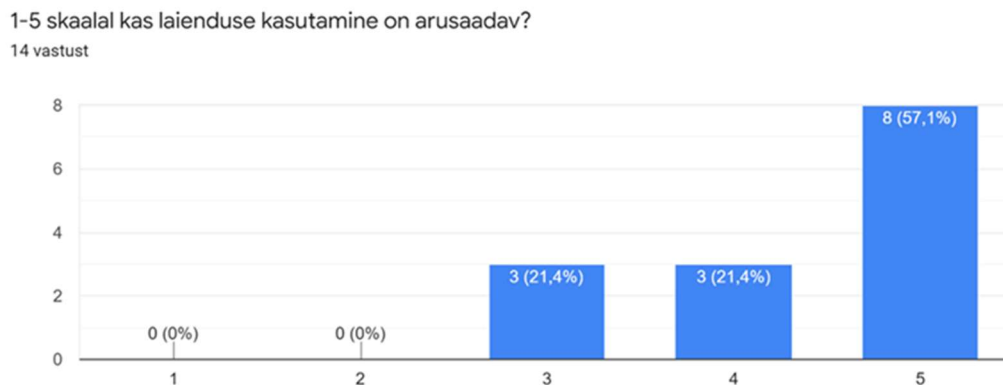
Üldine tagasiside oli positiivne ja laienduse kasutamine tekitas paljudes katseisikutes elevust. Katseisikud olid väga üllatunud kui täpselt laiendus dikteeritud teksti tuvastab.

Tagasiside andis mõista, et laiendust on võimalik veel palju edasi arendadaja mugavamaks teha. Samuti peegeldas tagasiside küsitlus peatükis 3 käsitletud kõnetuvastus rakenduse arendamise nõudmisi.

64.3% vastas, et nad kasutaksid seda laiendust edaspidi ka. Katseisikud, kes laiendust edaspidi ei kasuta andsid põhjenduseks selle, et nende meelest on klaviatuuril kirjutamine kiirem kui dikteerimine, aga nad tõdesid, et selline laiendus võib olla väga kasulik düsgraafikutele ja inimestele, kelle jaoks kirjutamine ei ole nii kerge.

6.1 Laienduse kasutamine

Suurim loodud laienduse puudus oli see, et laienduse kasutamine pole piisavalt arusaadav nagu näha joonisel 11.



Joonis 11. Laienduse kasutamise arusaadavus

Kasutamishüpsid olid Chrome'i veebipoe alla laadimise lehel olemas, aga tuleb välja, et kasutajad ei loe läbi kõiki detaile kui nad midagi internetist alla tõmbavad. Nimelt tekitab segadust brauseri tööriistariba tekkiv laienduse ikoon, mis on nähtav punases kastis joonisel 12.



Joonis 12. Tööriistariba peale laienduse installimist

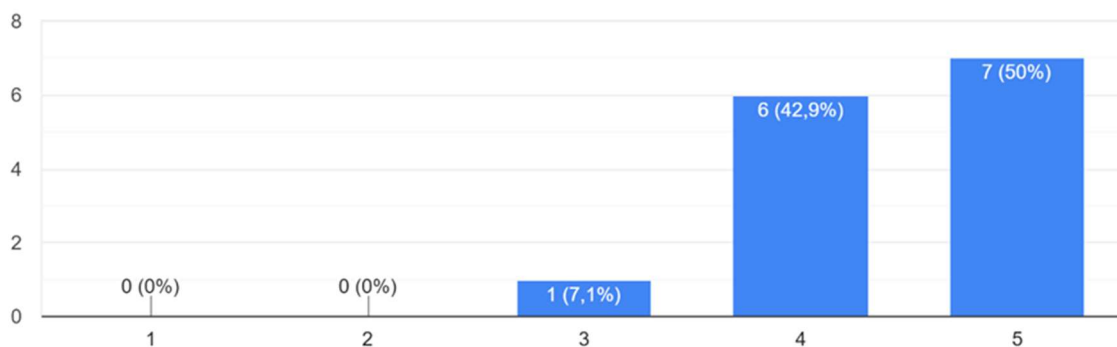
Osad katseisikud eeldasid, et laiendus hakkab kohe häält kuulama kui nad vajutavad tekkinud ikooni. Katseisikuid tabas segadus kui ikoonile vajutamine ei teinud midagi. Sellist käitumismustrit oleks pidanud ette nägema. Tulevikus tuleb laienduse edasi arendamisel kuvada ikooni vajutamisel väike hüpikaken, milles on juhised, kuidas laiendust kasutada.

6.2 Laienduse mugavus ja kujundus

Mugavuse statistika oli veidi parem nagu näha joonisel 13. Laienduse arendamisel oli selge, et paremkliki lahendus ei ole kõige mugavam viis, kuidas laiendust käivitada. Katseisikud tõid välja, et kõne tuvastamine võiks kohe alata kui vajutada mingit nuppu või tööriistareal olevat laienduse ikooni. Selline lahendus on sarnane eelnevalt mainitud tasulisele inglise keelse kõnetuvastus laiendusele *Speech Recognition Anywhere*. Selline lahendus on aga palju keerulisem ja nõuaks, et rohkem õigusi oleks Chrome'ilt küsitud.

1-5 skaalal kui mugav on laiendust kasutada?

14 vastust



Joonis 13. Laienduse kasutamise mugavus

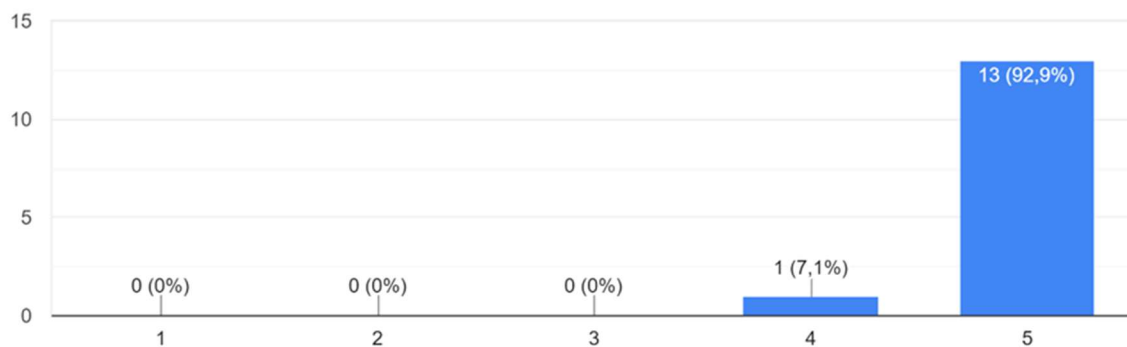
Soovitati ka seda, et hüpikakna avades hakkaks laiendus kohe kõne kuulama ning nupu asemel võiks olla käsklused alustamiseks ja lõpetamiseks. Dikteerimise ajal häiris katseisikuid ka see, et kui öelda „uus rida“ ei teki hüpikaknas mingit märget ja kasutaja ei saa aru kas uus rida tekkis või mitte. Ebamugavust tekitas ka see, et kui laiendus ei saanud sõnast aru. Sel juhul hakkab laiendus mõtlema ja ei kuva kasutajale mingit teadet ning kasutaja jääb ootama.

Probleemiks kujunes ka see, et kui dikteerida tekst tekstivälja, kuhu on juba teksti dikteeritud kaob eelenev tekst ära.

Kujunduse kohta oli väga positiivne tagasiside nagu näha joonisel 14 ning selle kohta ei saanud halba tagasisidet.

1-5 skaalal kuidas hindad laienduse kujundust?

14 vastust



Joonis 14. Laienduse kujunduse hinnang

7 Võimalikud edasiarendused

- 1) Arendada laiendusele lisaks Firefox ja Microsoft Edge brauserile toetus
- 2) Laienduse edasiarendused:
 - a. Eemaldada hüpikaken ja integreerida kõne tuvastamine lehe sisse ja/või aktiveerida kõne tuvastamine ühest nupust.
 - b. Lisada laiendusele häälkäsklused, kõnetuvastuse aktiveerimiseks ja lõpetamiseks.
 - c. Lisada võimalus häälkäsklustega veebilehti juhtida ilma ühegi hiire või klaviatuuri nupu vajutuseta.
 - d. Lisada võimalus lause ära lõpetada ning siis uuesti jätkata samast kohast dikteerimist.

8 Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli Google Chrome'i laiendus arendada Google'i Gmailide kirjutamise jaoks, aga seda eesmärki sai töökäigus laiendada igale veebilehele, kus on olemas tekstiväljad. Eesmärgi saavutamiseks analüüsi esmalt olemasolevaid lahendusi ja nende funktsionaalsust.

Lõputöö raames valmis laiendus, mis võimaldab dikteerida eestikeelset teksti vabalt valitud tekstivälja, mis kasutab kõne tekstiks transkribeerimiseks TTÜ keeletehnoloogia laboris välja töötatud serveripõhist kõnetuvastustarkvara.

Tulemuseks valmis laiendus nimega „Kõnetuvastus“, mis on avalikult Chrome'i veebipoes, mille aadress on leitav Lisa 2 lehel. Loodud laienduse jaoks koostati ka tagasiside küsitlus, mis kinnitas peatükis 3 käsitletud kõnetuvastus rakenduse arendamise nõudmisi.

Edasiarenduse järgmiseks etapiks oleks lisada laienduse toetus Firefox ja Microsoft Edge brauserile. Samuti saab laiendusele lisada lisafunktsionaalsusi ja parandada vigu, mis toodi välja kogutud tagasisides.

Kasutatud kirjandus

- [1] TTÜ keeletehnoloogia labor, „Kõnetuvastus“ [Võrgumaterjal]. Available: <https://www.keeletehnoloogia.ee/et/ekt-projektid/konetuvastus>. [Kasutatud 14 mai 2020].
- [2] TTÜ keeletehnoloogia labor, „Kõnetuvastus 2“ [Võrgumaterjal]. Available: <https://www.keeletehnoloogia.ee/et/ekt-projektid/konetuvastus-2> [Kasutatud 14 mai 2020].
- [3] TTÜ keeletehnoloogia labor, Dikteerimise veebileht brauseris [Võrgumaterjal]. Available: <http://bark.phon.ioc.ee/dikteeri/about.html>. [Kasutatud 10 aprill 2020].
- [4] TTÜ keeletehnoloogia labor, Dikteerimise veebileht brauseris [Võrgumaterjal]. Available: <http://bark.phon.ioc.ee/dikteeri/index.html>. [Kasutatud 10 aprill 2020].
- [5] K.Kaljulaid, „Kõnele“, [Võrgumaterjal]. Available: <https://play.google.com/store/apps/details?id=ee.ioc.phon.android.speak&hl=et>. [Kasutatud 10 aprill 2020].
- [6] Voice Recognition. [Võrgumaterjal]. Available: <https://chrome.google.com/webstore/detail/voice-recognition/ikjmfndklfaonkodbnidahohdfbdhkn>. [Kasutatud 10 aprill 2020].
- [7] Speech Recognition Anywhere. [Võrgumaterjal]. Available: <https://chrome.google.com/webstore/detail/speech-recognition-anywhere/kdnnmhpncakdilnofmllgcigkibjonof>. [Kasutatud 10 aprill 2020].
- [8] X.Huang, A.Acerro, H.Hon. (2001) „Spoken Language Processing: A Guide to Theory, Algorithm, and System Development“ - „Applications and User Interfaces“, 899-931, New Jersey.
- [9] Microsoft, „Microsoft Agent“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/windows/win32/lwef/microsoft-agent>. [Kasutatud 7 mai 2020].
- [10] Woo, Huay Lit & Wang, Qiyun. (2020). Designing an Agent-Based Learning Environment – A Preliminary Study on A Pre-service Teacher Course.
- [11] Google, „Can extensions modify chrome:// URLs?“. [Võrgumaterjal]. Available: <https://developer.chrome.com/extensions/faq#faq-dev-15>. [Kasutatud 10 aprill 2020].
- [12] Google, „What are extensions?“. [Võrgumaterjal]. Available: <https://developer.chrome.com/extensions>. [Kasutatud 10 aprill 2020].
- [13] Google, „Overview“. [Võrgumaterjal]. Available: <https://developer.chrome.com/extensions/overview>. [Kasutatud 10 aprill 2020].
- [14] Google, „Content Security Policy (CSP)“. [Võrgumaterjal]. Available: <https://developer.chrome.com/extensions/contentSecurityPolicy>. [Kasutatud 10 aprill 2020].
- [15] MDN web docs, „eval()“, [Võrgumaterjal]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval#Never_use_eval!. [Kasutatud 14 mai 2020].
- [16] Vue.js, „CSP environments“, [Võrgumaterjal]. Available: <https://vuejs.org/v2/guide/installation.html#CSP-environments>. [Kasutatud 20 mai 2020].

Lisa 1 – Manifest.json fail

```
{
  "name": "Kõnetuvastus",
  "description": "Brauseri laiendus eestikeelse teksti dikteerimiseks
kõnetuvastuse abil",
  "version": "2.0.0",
  "manifest_version": 2,
  "icons": {
    "48": "icons/icon_48.png",
    "128": "icons/icon_128.png"
  },
  "browser_action": {
    "default_title": "Kõnetuvastus"
  },
  "background": {
    "scripts": [
      "background.js"
    ],
    "persistent": true
  },
  "permissions": [
    "activeTab",
    "contextMenus"
  ]
}
```

Lisa 2 – „Kõnetuvastus“ laienduse veebipoe link

<https://chrome.google.com/webstore/detail/k%C3%B5netuvastus/flknjambpipjohjmnghlgnpcenaii>