



LAUREA
AMMATTIKORKEAKOULU
Yhdessä enemmän

Mobiilisovelluksen uudelleenkehitys

Pasanen, Perttu

2017 Laurea



Laurea-ammattikorkeakoulu
Laurea Leppävaara

Mobiilisovelluksen uudelleenkehitys

Perttu Pasanen
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Kesäkuu, 2017

Tietojenkäsittelyn koulutusohjelma

Pasanen Perttu

Mobiilisovelluksen uudelleenkehitys

Vuosi 2017

Sivumäärä 25

Opinnäytetyön tarkoituksena oli selvittää, miten olemassa oleva näppäimistösovellus voidaan uudelleenkehittää iOS-alustalta Android-alustalle ja tutkia näppäimistöön liittyviä toiminnallisuksia. Työ toteutettiin toimeksiantona MPaja Oy:lle, joka oli kehittänyt Viskey-nimisen esteettömän virtuaalinäppäimistön iPhoneille.

Viskey ja siihen opinnäytetyössä vertailut sovellukset on suunniteltu esteettömiksi näkövammaisille käyttäjille. Työssä selvitettiin olemassa olevien Google Play kaupan virtuaalinäppäimistöjen ominaisuuksia. Viskey:n esteettömiä ominaisuuksia vertailtiin sen tuleviin kilpailijoihin.

Työssä laadittiin Viskey-näppäimistösovelluksesta Android alustalle prototyyppi, jota voidaan käyttää jatkokehityksessä. Opinnäytetyössä käytiin läpi näppäimistösovellusprototyypin toteutukseen käytettyjä työkaluja, joista tärkein oli Android Studio. Kehitysympäristönä se mahdollisti koodin kääntämisen ja kokoamisen. Prototyypin kehittämiseen käytettiin Java -ohjelmointikieltä.

Pasanen Perttu

Recreation of a mobile application

Year	2017	Pages	25
------	------	-------	----

The aim of this thesis was to examine the process of recreating a keyboard application from the iOS platform to the Android platform and study the functionalities of this application. The thesis was commissioned and produced for MPaja Oy, who had created an accessible virtual keyboard named Viskey for the iPhone.

Viskey and the applications it was compared to are designed to be accessible for visually impaired users. The thesis explored popular virtual keyboard applications from Google Play Store and their properties. Viskeys accessible properties were compared to these future competitors.

The goal was to create a prototype application for the Android platform, which could be used to further develop Viskey for Android. The thesis also explored the tools used in creating the prototype of the keyboard application. The most important tool was Android Studio, a development environment that provided easy code compiling. Java programming language was used to create the prototype.

Keywords: mobile application, Android, virtual keyboard

Sisällys

1	Johdanto	6
2	Virtuaalinäppäimistöt	7
2.1	Korvattavissa kolmannen osapuolen näppäimistöillä	8
2.2	Suosituimmat näppäimistöt Google Play:ssä	8
3	Viskey-näppäimistö	11
3.1	Ominaisuudet	12
3.2	Esteettömyys	13
3.3	Viskey Androidille	13
3.4	Kilpailijat	13
4	Toteutustavan valinta	14
4.1	Java	14
4.2	Android Studio	15
4.3	Prototyypin toteutus Android Studiossa	16
4.4	Ulkoasu	19
4.5	Oikoluku	19
4.6	Esteettömyys	20
5	Sovelluksen testaus	20
6	Tulokset ja analyysit	21
7	Yhteenveto	22
	Lähteet	23
	Kuviot	24
	Taulukot	25

1 Johdanto

Mpaja Oy:n on pieni IT-ohjelmistoja ja -sovelluksia kehittävä yritys, joka on erikoistunut mobiilisovellusten kehitykseen. Mpaja Oy on kehittänyt Viskey-virtuaalinäppäimistösovelluksen iPhoneen iOS järjestelmälle. Opinnäytetyön tavoitteena on tutkia, mitä Viskey-sovelluksen uudelleenkehitys Android-käyttöjärjestelmän kanssa yhteensopivaksi vaatisi. Tarkoituksena on myös luoda prototyyppi Android-yhteensopivasta näppäimistösovelluksesta. Prototyypin pohjalta Mpaja Oy voi jatkaa sovelluksen kehitystä.

Alkuperäinen Viskey-virtuaalinäppäimistö on suunniteltu esteettömäksi, mahdollisimman selkeäksi ja helppokäyttöiseksi. Sen käyttöominaisuudet on suunniteltu näkövammaisia ja heikkonäköisiä ihmisiä ajatellen. Näppäimistössä on hyvä kontrasti ja isot kirjaimet, jotta yksittäiset kirjaimet ja symbolit olisi helppo erottaa.

MPaja Oy haluaa kehittää Viskey:n Android-versiosta mahdollisimman samanlaisen kuin iOS-versiosta. Tästä johtuen Android-järjestelmälle kehitettävän näppäimistön tulee sisältää nämä samat ominaisuudet esteettömyyden ja käytettävyyden suhteen.

Näppäimistön ulkoasu ja värit ovat jo valmiita, joten opinnäytetyössä ei tarvitse puuttua ulkoasuun tai näppäinten asetelmaan. Esteettömyyden kannalta on tärkeää selvittää Android-laitteisiin esiasennetun TalkBack-palvelun toimintaa yhdessä näppäimistön kanssa. Opinnäytetyössä käydään läpi uuden vaihtoehdoisen näppäimistön kehittämiseen liittyviä seikkoja.

Opinnäytetyössä käydään ensin läpi työn kannalta keskeisiä käsitteitä.

Keskeiset käsitteet: Android, TalkBack, VoiceOver, API taso:

Android

Mobiilikäyttöjärjestelmä Android julkaistiin vuonna 2008, jolloin se nousi haastamaan olemassa olevat järjestelmät kuten iOS:n, ja Windows Phone 7:n ja Blackberry OS:n. Siitä lähtien Androidin käyttö on lisääntynyt huomattavasti. (Zechner, M. & Green, R. 2012, 2). On tärkeää, että laajalti käytössä olevalle Androidille on tarjolla esteettömiä ratkaisuja. Android on kuitenkin enemmän kuin vain käyttöjärjestelmä, sillä se on mobiililaitteille kehitetty avoimeen lähdekoodiin perustuva kattava ohjelmistopino (Gargenta & Nakamura 2014, 1). Tämän opinnäytetyön kannalta yleinen merkitys Androidista käyttöjärjestelmänä on kuitenkin riittävä.

TalkBack

TalkBack on esteettömyyspalvelu Android-mobiililaitteille. Sovellus hyödyntää puhuttua ääntä, värinää ja muita auditiivisia keinoja antaakseen käyttäjälleen tietoa siitä, mitä näytöllä on, missä kohtaa sormi on näytöllä ja mitä tapahtuu siitä painettaessa. (Android central 2017). TalkBack toimii kaikissa Android-laitteissa versio 4.0:sta lähtien, ja se aktivoidaan laitteen helppokäyttö-, esteettömyys- tai saavutettavuusasetuksissa.

Kun puhelimen käyttäjän sormi osuu mobiililaitteen näytössä olevaan elementtiin TalkBackin ollessa aktiivinen, laite kertoo käyttäjälle täsmälleen mitä siinä lukee. Jos taas elementtiä painamalla voi suorittaa laitteella jonkin toiminnon, TalkBack kertoo mitä kohtaa käyttäjä on painanut. Tällöin TalkBack antaa joko suorittaa toiminnon loppuun kaksoisnapautuksella tai siirtyä toiseen elementtiin yksinkertaisella näpäytyksellä. (Android central 2017). TalkBackin avulla mobiililaitteella pystyy siis navigoimaan kosketuksen avulla, vaikka ei näkisikään tekstiä.

VoiceOver

VoiceOver on elepohjainen näyttölukija, jonka avulla käyttäjä kuulee kuvauksen kaikesta, mitä mobiililaitteen näytöllä tapahtuu. (Apple 2017) VoiceOver siis palvelee käyttäjää samantapaisesti kuin TalkBack ja mahdollistaa mobiililaitteen käytön ilman visuaalista ulottuvuutta.

API (application program interface) sovellusliittymätaso

Android-käyttöjärjestelmän versiot perustuvat API-tasoihin. API-taso määrittää sisäänrakennetut toiminnot ja funktiot, joiden avulla kommunikoida laitteen kanssa. API-tason noustessa toiminnallisuus kasvaa. Sovelluksilla ja laitteilla on omat API-tasot. Sovellukset eivät toimi laitteilla, joiden API-taso on niiden omaa API-tasoa pienempi. (Android Developers 2017) Android-version API-taso yksilöi sen, mutta samalla se myös estää ylempien API tason omaavien sovellusten asentamisen. Tällä pyritään varmistamaan sovellusten yhteensopivuus mobiililaitteen kanssa ja sitä kautta sovelluksen toimivuus. (Developer Xamarin 2017)

2 Virtuaalinäppäimistöt

Älypuhelimissa on fyysisen näppäimistön sijaan kosketusnäyttö, johon näppäimistö toteutetaan sovellusten avulla. Näppäimistösovellus luo näkymän mobiililaitteen näytölle. Kun näytössä näkyviä näppäimiä painetaan, sovellus muuntaa painallukset painetuiksi merkeiksi. Tämän jälkeen merkit välittyvät aktiiviseen tekstikenttään. Virtuaalinäppäimistössä voi olla monia eri nappeja, jotka eivät mahtuisi fyysiseen näppäimistöön. Virtuaaliseen näppäimistöön on

mahdollista sisällyttää esimerkiksi hymiöitä, kaikenlaisia erikoismerkkejä ja muiden kielten kirjaimia.

2.1 Korvattavissa kolmannen osapuolen näppäimistöillä

Näppäimistöä voi vaihtaa lataamalla sovelluskaupasta eri näppäimistösovelluksia. On olemassa sekä ilmaisia että maksullisia kolmannen osapuolen kehittämiä näppäimistöjä, ja eri näppäimistöjä löytyy suuri valikoima. Näppäimistöjä löytyy kaikenlaisia; jotkut niistä muuttavat teemaa, toiset taas ovat visuaalisesti samankaltaisia, mutta omaavat eri toimintoja. Jotkut näppäimistösovellukset vaihtavat tapaa, jolla näppäimistön näppäimet on aseteltu. Näppäimistösovellukset voivat myös lisätä uusia merkkejä, kuten esimerkiksi hymiöitä, joita löytyykin monia erilaisia.

2.2 Suosituimmat näppäimistöt Google Play:ssä

Viskey on tarkoitus uudelleenkehittää Android-alustalle. MPaja Oy pyysi selvittämään näppäimistösovellusten tilanteen Google Play -kaupassa. Aluksi kartoitus keskittyi Google Play -kaupan suosituimpiin näppäimistöihin ja niiden ominaisuuksiin. Seuraavaksi tutustuttiin Google Play -kaupasta löytyviin esteettämiin näppäimistöihin ja vertailtiin niiden ominaisuuksia Viskey:n iOS versioon.

Mobiililaitteissa on jo valmiiksi asennettu oletusnäppäimistösovellus, joka vaihtelee laitteen valmistajasta riippuen. Samsung A3 -mallin älypuhelimessa esiasennettu oletusnäppäimistö oli Samsungin itse kehittämä Samsung Keyboard, kun taas Asus:ksen tabletissa oletusnäppäimistönä oli Google-näppäimistö. Nämä olivat opinnäytetyössä käytetyt testilaitteet.

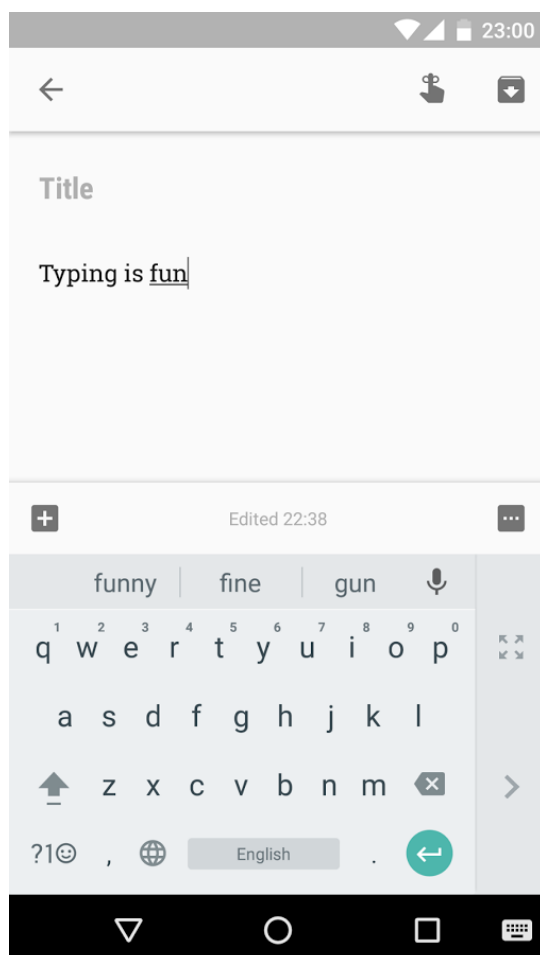
Seuraavat näppäimistöt olivat ensimmäisinä tuloksia haettaessa näppäimistösovelluksia Google Play -sovelluskaupassa. Esimerkit ovat neljä ensimmäistä hakutulosta. Niitä oli ladattu eniten ja niiden keskimääräinen käyttäjäarvio oli yli neljä tähteä viidestä. Näitä näppäimistöjä onkin ladattu huomattava määrä. (Google Play 2017.)

Latauksien määrä (Lähde: Google Play, 2017):

- Go Keyboard: 4 167 396
- SwiftKey: 2 291 006
- Emoji Kika Pro: 1 051 126
- Google Keyboard: 1 330 501 (esiasennettu oletusnäppäimistö suuressa osassa laitteista)

	Piirtokirjoitus	Puhekirjoitus	Sanojen ennakointi	Sanojen korjaus	Teeman vaihto	Hymiöt
Google-näppäimistö	X	X	X	X		
Go Näppäimistö	X			X	X	
SwiftKey	X		X	X	X	X
Emoji Keyboard Pro Kika Free	X		X	X	X	X

Taulukko 1 Google Play:n Näppäimistöjen ominaisuudet



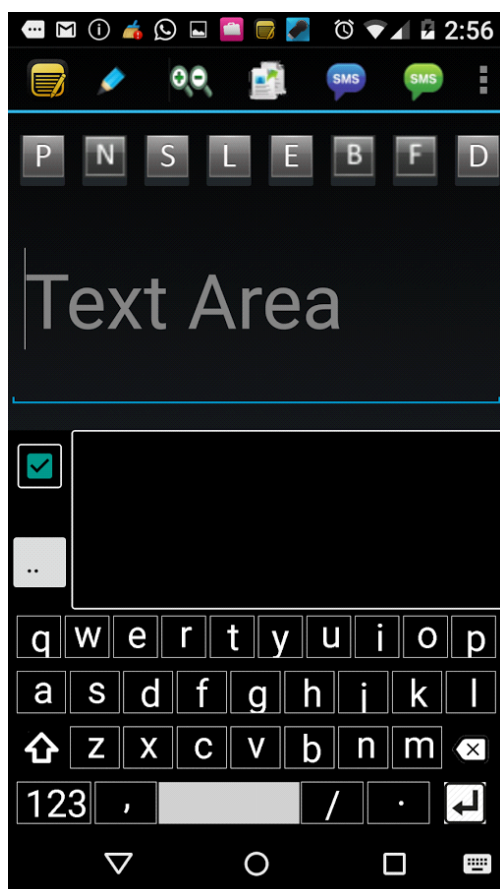
Kuvio 1: Google Keyboard

Seuraavaksi sovellusten kartoitus keskittyi esteettömiin näppäimistösovelluksiin, jolloin hakutulokset olivat aivan erilaisia kuin haettaessa generistä näppäimistösovellusta. Useimmat

tulokset eivät kuitenkaan olleet esteettömiä esimerkkejä eli heikkonäköisille tai sokeille ihmisille suunniteltua näppäimistöjä. Esimerkeiksi valittiin vain sovellukset, joiden nimessä esiintyi termi ”accessible” tai jotka olivat muilta ominaisuuksiltaan suunniteltu huononäköisille. Hakutuloksien vähydestä ja epätarkkuudesta voimme päätellä, ettei esteettömiä näppäimistöjä ole kovin monta. Taulukkoon 2 valittiin tarkasteltavaksi kolme muuta esteettömäksi suunniteltua näppäimistöä Viskey:n lisäksi.

	TalkBack yhteensopiva	Näkemistä helpotettava Värikontrasti	Suurentava sanojen esikatselu
Accessible Keyboard	X		X
Blind Accessibility Keyboard		X	
Eyes-Free Keyboard	X		
Viskey (iOS versio)	X (VoiceOver yhteensopiva)	X	X

Taulukko 2 Google Play:n esteettömien näppäimistöjen ominaisuudet ja Viskey

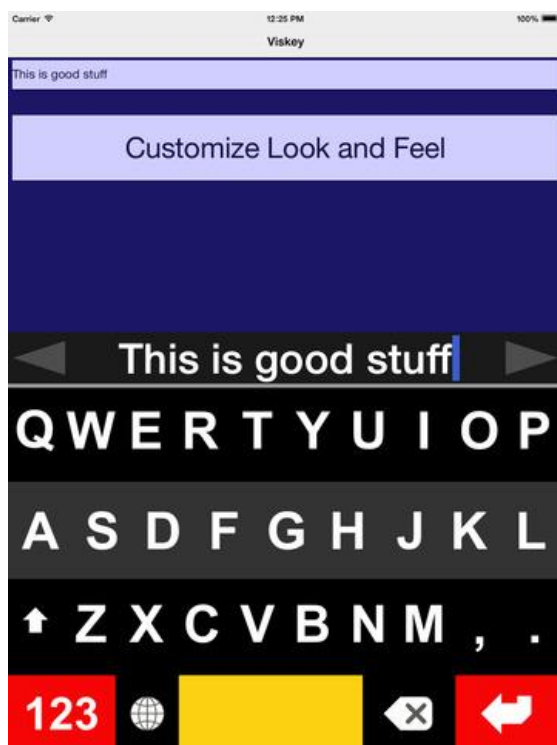


Kuvio 2: Accessible Keyboard

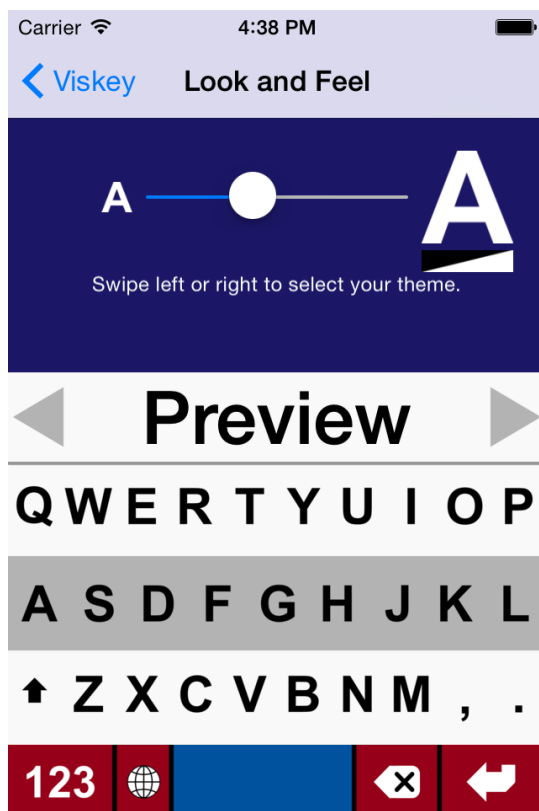
3 Viskey-näppäimistö

Viskey-näppäimistö on suunniteltu esteettömäksi, mahdollisimman selkeäksi ja helppokäyttöiseksi. Sen käyttöominaisuudet on suunniteltu näkövammaisia ja heikkonäköisiä ihmisiä ajatellen. Näppäimistön kirjaimet ovat valkoisia isokokoisia suuraakkosia mustalla taustalla. Toinen kirjainrivi on tummansinisellä taustalla, mikä tekee näppäinten erottamisesta vielä helpompaa. Näppäimet on aseteltu normaalin qwerty-näppäimistön tavoin. Alarivissä on punaisella taustalla numeronäppäin, jota painamalla numeronäppäimistö vaihtuu kirjainten tilalle. Numeronäppäimen oikealla puolella on maapallonappi, jolla näppäimistösovelluksen voi vaihtaa toiseen. Näiden lisäksi näppäimistössä on keltainen välilyöntinäppäin, jonka vieressä on backspace-näppäin. Backspace-näppäimen oikealla puolella on enter-näppäin punaisella taustalla. (Kuvio 3)

Näppäimistön värit on mahdollista vaihtaa, jolloin värit muuttuvat käänteisiksi. (Kuvio 4) Tällöin kirjaimet ovat mustia, tausta on valkoinen ja keskirivin tausta on vaaleansininen. Alarivin nappien tausta myös vaihtuu helpommin havaittavaksi. Niiden väriä voi vaihtaa erikseen. Heikkonäköisten auttamiseksi itse näppäimistön yläpuolella näkyy suurella fontilla kirjoitettu teksti, jonka kokoa voi muuttaa näkyvyyden parantamiseksi. Tekstiä on mahdollista selata vasemmalle ja oikealle osoittavilla nuolilla, jotka vaihtavat kirjoituskohtaa yhdellä merkillä osoitettuun suuntaan.



Kuvio 3: Viskey-näppäimistö



Kuvio 4: Viskey-näppäimistön vaihtoehtoinen väritys

3.1 Ominaisuudet

Viskey-näppäimistössä on self-voicing -toiminto, joka lausuu sanat ja merkit. Tämä self-voicing -toiminto käyttää laitteen omaa TTS (Text To Speech) -järjestelmää. Käyttäjä voi valita, mitkä kaikki toiminnot generoivat ääniefektin. Näihin valintoihin sisältyy yksittäisten kirjainten ja kokonaisten sanojen lausunta.

Viskey toimii myös iOS-järjestelmän VoiceOverin kanssa. Kun VoiceOver on päällä, Viskey antaa sen hoitaa puheen generoimisen myös silloin kun Viskey:n omat äänet ovat käytössä. Tällöin vältetään ongelmat äänien päällekkäisyyden kanssa. Näppäimistöä voi kuitenkin luonnollisesti käyttää myös ilman äänipalautetta.

Viskey hyödyntää iOS-järjestelmän sanakirjakokoelmaa, minkä ansiosta kaikille yleisimmille puhutuille kielille löytyy oikeinkirjoituksen tuki. Viskey:n asetuksista voi vaihtaa oikeinkirjoituksen asetuksia, kääntää oikeinkirjoituksen tarkistuksen pois päältä, alleviivata väärinkirjoitetun sanan tai piipata väärin kirjoitetun sanan kohdalla.

3.2 Esteettömyys

Viskey-näppäimistö pyrkii helppokäyttöisyyteen sisältäen ainoastaan kirjain- ja numeronäppäimiä sekä yleisimmät merkinäppäimet. Kun näppäinten määrää on rajoitettu, yksittäisten näppäinten koko voi olla suurempi. Myös näppäimissä käytetty fontti on selkeä. Havaitessaan käyttäjän tekemän kirjoitusvirheen, Viskey antaa visuaalisen merkin lisäksi myös äänimerkin. Tavoitteena on, että käyttäjä varmasti huomaa virheen ja voi palata takaisin sen korjatakseen.

Viskey:tä on mahdollista käyttää myös pelkkien äänien perusteella, jolloin sovellus lausuu painetut kirjaimet ja sanat ääneen. Viskey:tä ei kuitenkaan ole suunniteltu toimimaan pelkästään äänien avulla, vaan äänten on vain tarkoitus tukea Viskey:n käyttöä. MPaja Oy on kehittänyt myös kokonaan ääniin pohjautuvan erillisen näppäimistösovelluksen, MBraille:n.

3.3 Viskey Androidille

Ulkoasun kehittämistyö on tehty, joten tavoitteena on saada Android-käyttöjärjestelmälle toteutettavan Viskey:n toiminnallisuus samankaltaiseksi kuin alkuperäisessä näppäimistösovelluksessa. Applen mobiililaitteisiin yhteensopiva Viskey-näppäimistö on toteutettu eri ohjelmointikielellä ja se hyödyntää iOS-ohjelmistokehitysketyn sisäänrakennettuja funktioita. Tämän takia onnistunut Android-yhteensopivuus vaatii, että näppäimistön koodi kirjoitetaan kokonaan uudelleen toisella ohjelmointikielellä eli Javalla. Uudelleenkehittämisessä tulee kuitenkin koko ajan ottaa huomioon alkuperäisen näppäimistön ulkoasu ja käyttöominaisuudet.

3.4 Kilpailijat

Viskey-näppäimistö ei tule kilpailemaan suosituimpien näppäimistöjen, kuten Google Keyboard:in ja SwiftKey:n kanssa. Toisin kuin Viskey, niitä ei ole suunniteltu alusta lähtien esteettömiksi, vaikka ne toimivatkin yhdessä TalkBackin kanssa. Viskey:n kilpailijoita Android-laitteilla tulevat olemaan muut alusta alkaen esteettömiksi kehitetyt näppäimistöt.

Esteettömiä näppäimistöjä on olemassa, mutta niitä ei ole kovinkaan monta. Google Play-kaupasta esteettömiä näppäimistöjä haettaessa vain osa hakutuloksista oli esteettömiksi suunniteltuja näppäimistöjä. Esteettömien näppäimistöjen vähyydestä johtuen hakutulosten joukossa oli myös paljon tavallisia näppäimistöjä. Lisäksi jotkut hakutuloksissa esiintyvistä esteettömistä näppäimistöistä eivät toimineet halutulla tavalla tai ne olivat hyvin keskeneräisiä.

Joissakin esteettömissä näppäimistösovelluksissa oli samankaltaisia ominaisuuksia kuin Viskey:ssä. Kirjoitettu teksti näkyy erikseen suurella fontilla myös Accessible Keyboard-sovelluksessa. TalkBackin kanssa yhteensopivuus toimii myös Blind Accessibility Keyboardissa ja Accessible Keyboardissa. Lisäksi Blind Accessibility Keyboardissa on muokattava värikontrasti-asetus, joka auttaa erottamaan merkit taustasta. Vain Viskeyssä oli sekä näkemistä helpottava värikontrasti että suurentava sanojen esikatselu yhdessä TalkBack-yhteensopivuuden kanssa. Muissa sovelluksissa kaikki esteettömät ratkaisut eivät esiintyneet samassa sovelluksessa, mikä saattaa antaa Viskey:lle etulyöntiaseman kilpailijoihinsa verrattuna.

4 Toteutustavan valinta

Työssä kehitettiin prototyyppi näppäimistösovelluksesta. Sovelluksen kehitykseen löytyi ohjeita viralliselta Android-kehityssivustolta ja Android-kehittäjien suosimilta foorumisivustoilta. Itse kehitys toteutui Android Studiossa käyttäen Java-ohjelmointikieltä. Mpaja Oy ei ole aiemmin kehittänyt Android-alustalle virtuaalinäppäimistöä, joten prototyyppi tarjoaa heille jatkokehityspohjan näppäimistösovellukselle. Lisäksi kehitysprosessin yhteydessä he saavat hyödyllistä tietoa esimerkiksi Android Studion käytöstä.

4.1 Java

Java on Sun Microsystemsin vuonna 1995 julkaistu ohjelmointikieli. Java on maailmanlaajuisen standardi sulautettujen ohjelmistojen, mobiilisovellusten, pelien, web-pohjaisen sisällön ja yritysohjelmistojen kehityksessä. Javalla on yli 9 miljoonaa kehittäjää ja noin 3 miljardia matkapuhelinta käyttää Javaa.

Javaa on testattu, paranneltu ja laajennettu suuren Java-kehittäjäyhteisön toimesta. Java on suunniteltu mahdollistamaan korkean suorituskyvyn kannettavien sovellusten toiminnan eri käyttöympäristöissä. Yhdessä käyttöjärjestelmässä kehitettyä ohjelmaa voi siis suorittaa myös lähes kaikissa muissakin ympäristöissä. (Java 2017.)

Javan kanssa työskennellessä kirjoitetaan Java-lähdetiedosto, joka käännetään Java-tavukoodiksi Java-kääntäjällä. Tämä tavukoodi viimein suoritetaan JVM:llä eli Java Virtual Machine:lla. Dalvik taas on virtuaalikone, joka kehitettiin nimenomaan Android-mobiililaitteille. Mobiililaitteilla on rajallinen akun kesto ja niiden suorituskyky on paljon pienempi kuin pöytäkoneilla. Dalvik rakennettiin alusta asti näitä rajoituksia ajatellen. (Gargenta & Nakamura 2014, 36-37.)

Androidissa käytetty Dalvik on optimoitu mobiililaitteille. Toisin kuin JVM, Dalvik lataa sovelluksen luokkatiedoston viittaamalla sen sijaintiin levyllä. Tällä tavalla muistin käyttö pyritään

minimoimaan ja se on nopeampaa, koska mobiililaitteet käyttävät SSD-muistia kiintolevyn sijaan. Dalvik kehitettiin suorittamaan Java-tavukoodia, minkä takia ohjelmien kehitys kannattaa toteuttaa Java-ohjelmointikielellä virheiden välttämiseksi. (Gargenta & Nakamura 2014, 36-37.) Lisäksi myös Android Studio on suunniteltu käytettäväksi Javan kanssa. Android Studio ehdottaa ja tuo tarvittavat Java-luokat ohjelman koodia kirjoittaessa.

4.2 Android Studio

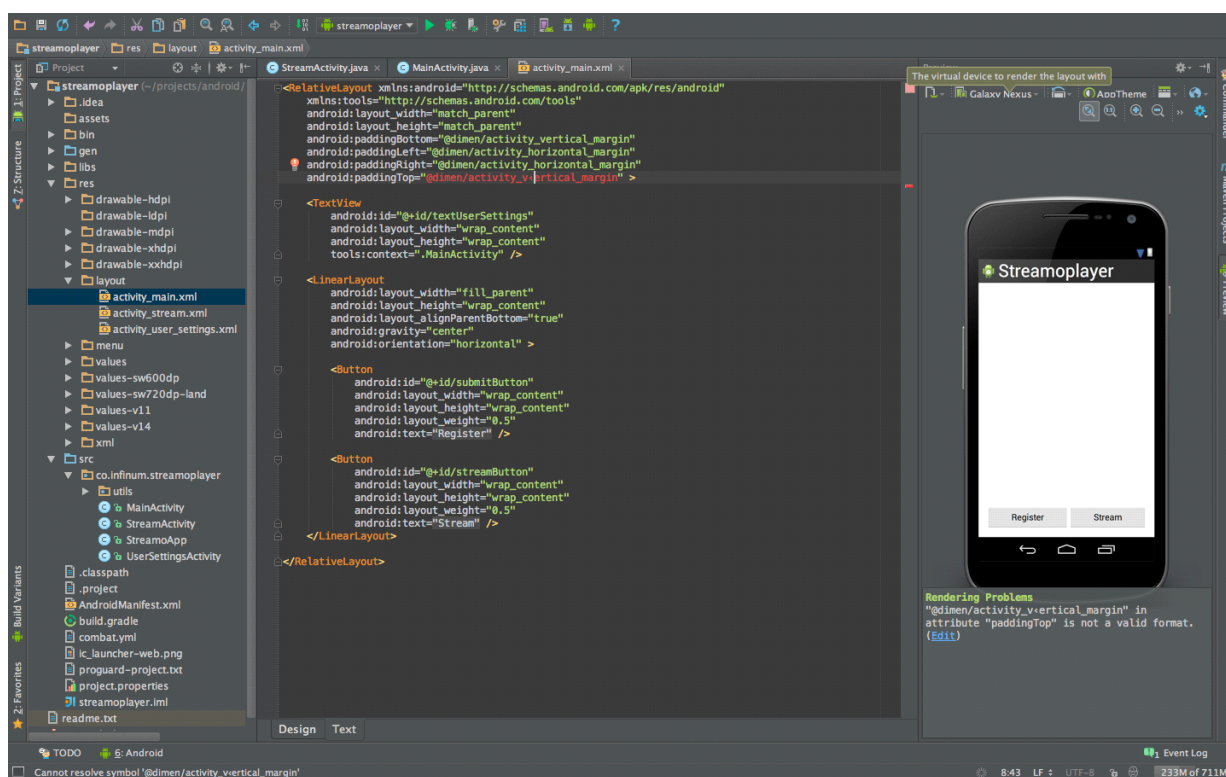
Näppäimistösovellusta kehitetään Androidin virallisen Android Studio-kehitysympäristön avulla. Aiemmin virallisena kehitysympäristönä käytettiin Eclipseä. Android Studio on kuitenkin uudempi ja monipuolisempi ympäristö, joka tukee myös projektien tuomista Eclipsestä. On suositeltavaa käyttää virallista ja uusimpaa kehitysympäristöä. Android Studio on todella kattava kehitysympäristö: se sisältää Android SDK:n (Software development kit), ADB:n (*Android Debugging Tools*) ja Android laite-emulaattorin, jonka avulla voi testata sovellusten toimivuutta ilman fyysistä laitetta. (Developer Android 2017.) Sovellusta voi myös ajaa USB-liitännällä kytketyssä laitteessa. Fyysinen laite vastaa paremmin käyttäjän toimintoihin kuin emulaattori.

Android Studio helpottaa sovellusten kehittämistä huomattavasti. Siinä on helppo määritellä API-taso, jolle sovellus tullaan julkaisemaan. Vanhemmat API:t eivät tue joitain uusia ominaisuuksia, jotka tekevät sovellusten tekemisestä helpompaa. API-tason määrittämisessä tuli ottaa huomioon käytettävyyden eri Android-versioilla. Ihannetilanteessa sovellus toimisi kaikilla Android-versioilla, mutta se ei käytännössä aina ole mahdollista. Tämä johtuu siitä, että jotkut uudet funktiot ja toiminnot toimivat vain tietyistä API-tasoista lähtien. Android Studio kuitenkin osaa kertoa sovelluksen kehittäjälle sillä hetkellä suositellun API-tason. Valitsemalla suositellun API-tason voi sekä tavoittaa mahdollisimman suuren osan käyttäjistä että hyödyntää käytetyimpiä funktioita sovelluksen kehityksessä.

Android Studio osaa ehdottaa ja tuoda sisäänrakennettuja funktioita, mikä auttaa sovellusten kehityksessä. Tämän lisäksi Android Studiolla on graafinen työkalu, jolla voi muotoilla sovelluksen ulkoasua. Elementtejä voi siirtää laitteen näytöllä, mikä auttaa reaaliaikaisessa ulkoasun kehityksessä.

Android Studio vaatii aina manifest-tiedoston, jossa on tiedot sovelluksen nimestä, tarkoituksesta ja sovelluksen vaatimista käyttöoikeuksista. Java-tiedostoon tulevat kaikki Java-lähdetiedostot, joita sovellus käyttää. Resurssi-tiedostoon on koottu alatiedostoihin eri arvoja; käyttöliittymät (layout), merkkijonot, kuvat, värit ja teemat. (Javaworld 2017.) On hyvä käytäntö määrittää elementtien nimet strings.xml tiedostossa, koska se antaa keskitetyn säi-

lytyspaikan eri teksteille. Tällöin niitä on helppo muokata myös jälkikäteen, sillä pitkiin nimiin voi viitata pelkän Id:n avulla.



Kuvio 5: Android Studioon käyttöliittymä

4.3 Prototyypin toteutus Android Studiossa

Aluksi kehitys seurasi ohjeita ja esimerkkejä, joita Android-kehittäjät olivat laatineet virtuaalinäppäimistön kehitykseen. Näppäimistösovellus on ominaisuuksiltaan IME eli Input Method Editor. IME-sovellus mahdollistaa tekstin syötön ja editoinnin. IME tosin vaatii, että sovellus luokitellaan service-sovellukseksi. Android Studiossa service-sovellus tarkoittaa sitä, että näppäimistö on käynnissä taustalla muiden sovellusten ollessa etusijalla. Tällainen näppäimistösovellus ei ole suorituskyvyn suhteen vaativa, koska mitään monimutkaisia toimintoja ei ole. Prosessorin ja muistin kannalta sovellus tulee toimimaan myös halvemmissä ja heikkotehokkaissa mobiililaitteissa. Sovellus määritetään Manifest.xml ja method.xml tiedostoissa IME ja service sovellukseksi. Manifest-tiedosto myös kertoo sovellusta suorittavalle laiteelle sovelluksen pyytämät käyttöoikeudet, näppäimistösovellus ei kuitenkaan pyydä tai tarvitse erityisiä oikeuksia.

Android Studioon uuden projektin luonnin yhteydessä myös ”luo aktiviteetti” kohta poistettiin, koska näppäimistösovellus ei hyödynnä aktiviteettia. Aktiviteetti Android-sovelluksessa varaa

näytön näkymän omaan käyttöönsä. Tässä opinnäytetyössä tavoitteena oli kehittää service-sovellus, joka toimii muiden sovellusten taustalla.

API-tasot määrittävät, mitkä kaikki toiminnot ovat hyödynnettävissä sovelluksen toiminnassa. Uusien API-tasojen myötä tulee uusia funktioita, joita pystyy hyödyntämään sovellusten kehityksessä. Suuremmat API-tasot ovat aina yhteensopivia pienempien API-tasojen kanssa, mutta ei toisinpäin. Sovellusten kehitykseen liittyen API-tason valinnassa tulisi ottaa huomioon se, kuinka monta mobiililaitetta tukee tiettyä API-tasoa. Jos valitsee alemman API-tason, sovellus on yhteensopiva useamman mobiililaitteen kanssa, mutta tällöin sen kehitys ja käytettävyys voi vaikeutua merkittävästi.

Kehitysprosessin alussa API taso määräytyi Android Studion ehdotuksen mukaiseksi. Ehdotettu taso oli 21, joka oli sillä hetkellä API-tasoista melkein uusin. Sen valitsemisen myötä sovellus olisi yhteensopiva 83 prosentissa kaikista Android-mobiililaitteista. Ehdotus perustui reaaliaikaiseen arvioon maailman kaikista Android-käyttöjärjestelmällä toimivista mobiililaitteista.

Projektit Android Studiossa sisältävät ”res”-resurssikansion, joka sisältää kaikki sovelluksen käyttämät arvot, nimet, kuvat, tyylit. Nämä kaikki tiedostot ovat .xml-muotoa. Näppäimistösovelluksen yksi merkittävä resurssitiedosto on tyylitiedosto, joka määrittää käyttäjälle esitettävän näkymän. Tämä tyylitiedosto sisältää näppäinten koon, taustan värin ja näppäimistön sijainnin näytöllä. (Kuvio 6)

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:background="#ffff00"
    android:textStyle="bold"
    android:textSize="30sp"
    >
</TextView>
```

Kuvio 6: Tyylitiedoston sisältämä koodi

Toisessa resurssitiedostossa (keyboard.xml) määriteltiin näppäimistön sisältämien eri näppäinten lisäksi myös niiden järjestys ja etäisyys toisiinsa nähden. Jokaiselle näppäimistöissä olevalle näppäimelle voi antaa tiettyjä arvoja. Näkyvän ja näkymättömän arvon määrittäminen kullekin näppäimelle on pakollista. Näkyvä arvo määrittää käyttäjän näkemän näppäimen merkin, kuten esimerkiksi "A". Näkymätön arvo taas vastaa näkyvän arvon merkkiä ("A"). Näppäimistösovellus hyödyntää näkymätöntä arvoa silloin, kun se siirtää käyttäjän painaman merkin tekstikenttään. (Kuvio 7)

```

<Keyboard xmlns:android="http://schemas.android.com/apk/res/android"
    android:keyWidth="10%p"
    android:horizontalGap="0px"
    android:verticalGap="0px"
    android:keyHeight="60dp"
>
    <Row>
        <Key android:codes="49" android:keyLabel="1" android:keyEdgeFlags="left"/>
        <Key android:codes="50" android:keyLabel="2"/>
        <Key android:codes="51" android:keyLabel="3"/>
        <Key android:codes="52" android:keyLabel="4"/>
        <Key android:codes="53" android:keyLabel="5"/>
    </Row>
</Keyboard>

```

Kuvio 7: Keyboard.xml-tiedoston sisältöä

Tarvittavien tausta-arvojen luomisen jälkeen sovellus tarvitsee .java-kooditiedoston määrittämään sovelluksen toiminnan. Kooditiedosto sisältää sovelluksen toimintaan tarvittavan koodin. Arvot ja tyyli saadaan resurssitiedostoihin viittaamalla. Kooditiedostossa luodaan funktioita eri tarkoituksiin. Esimerkiksi yksi funktio tuo näppäimistön esiin käyttäjän näytölle sekä tarkkailee, jos käyttäjä painaa näppäimistön nappeja.

Toisen funktion (onKey) tehtävä on tarkkailla käyttäjän toimintoja ja välittää käyttäjän painaman näppäimen arvo tekstikenttään tai suorittaa muu näppäimeen liittyvä toiminto. Normaali numero- tai kirjainpainallus välittyy tekstikenttään, mutta erikoisnäppäinten painallus saa aikaan uniikin toiminnon. Tämä sama funktio myös kutsuu tarvittaessa muita funktioita näppäimiä painaessa, eli tässä tapauksessa funktion, joka generoi äänen näppäimiä painettaessa. (Kuvio 8)

```

public void onKey(int primaryCode, int[] keyCodes) {
    InputConnection ic = getCurrentInputConnection();
    playClick(primaryCode);
    switch(primaryCode){
        case Keyboard.KEYCODE_DELETE :
            ic.deleteSurroundingText(1, 0);
            break;
        case Keyboard.KEYCODE_SHIFT:
            caps = !caps;
            keyboard.setShifted(caps);
            kv.invalidateAllKeys();
            break;
        case Keyboard.KEYCODE_DONE:
            ic.sendKeyEvent(new KeyEvent(KeyEvent.ACTION_DOWN, KeyEvent.KEYCODE_ENTER));
            break;
        default:
            char code = (char)primaryCode;
            if(Character.isLetter(code) && caps){
                code = Character.toUpperCase(code);
            }
            ic.commitText(String.valueOf(code),1);
    }
}

```

Kuvio 8: onKey-funktion sisältö

Nämä olivat keskeisimmät tiedostot ja funktiot, joita tarvitaan toimivan näppäimistösovelluksen kehittämiseen. Niiden avulla näppäimistösovelluksen on mahdollista suoriutua ensisijaisesta tehtävästään, joka on käyttäjän tekemien painallusten välittäminen tekstikenttään.

Luomalla uusia funktioita ja muuttamalla niiden käyttämiä arvoja, sovelluksesta voidaan tehdä monipuolisempi muuttamalla esimerkiksi sen ulkoasua tai toimintaa.

4.4 Ulkoasu

Viskey:n iOS-version ulkoasu on valmis. Ulkoasu perustuu taustakuvaan eli kuvaan näppäimistöä, jonka päälle oli luotu näkymättömiä näppäimiä. Niitä painaessa sovellus välitti painalluksen ja lasi sen arvon, kääntäen sen painetuksi merkiksi. Taustakuvaa ja muita Viskey:n kuvia voi hyödyntää sellaisinaan uutta Viskey-näppäimistöä kehitettäessä.

Näppäimistön ulkoasu tulee olemaan samanlainen kaikissa mobiililaitteissa näytön koosta riippumatta. Tämä johtuu siitä, että näppäinten koot voidaan ja ne suositellaankin määrittämään prosentuaalisesti, jolloin näppäinten koko skaalautuu riippuen mobiililaitteen näytön koosta ja tarkkuudesta. Näppäimistön ulkoasu säilyy siis samana, mutta haasteena on sen toimintojen uudelleenluonti. Näppäimistön tulee sekä näyttää samalta että toimia samalla tavalla. Sen siis tulee ottaa käyttäjältä vastaan erilaisia komentoja, kuten painalluksia ja pyyhkäisyjä, samalla tavalla kuin alkuperäisessä näppäimistössä.

4.5 Oikoluku

Kaikissa iOS-laitteissa on sanakirjakokoelma eli järjestelmän sisäinen sanakirja, jolla käyttäjä voi tarkistaa sanan oikeinkirjoituksen. Lisäksi, kehitettäessä sovellusta iOS-järjestelmälle, on mahdollista hyödyntää iOS:n sisäänrakennettua UITextChecker-funktiota. Tämä funktio osaa tarkistaa oikeinkirjoituksen tekstistä sanakirjakokoelmaa käyttäen. Sovelluksen kehittäjän ei siis itse tarvitse kehittää omaa oikolukufunktiota, vaan hän voi hyödyntää iOS:n sisäänrakennettuja ominaisuuksia. (Developer Apple 2017.) Myös Viskey-näppäimistö hyödyntää tätä iOS-järjestelmän valmista oikolukufunktiota.

Android Studiossa on valmis oikolukufunktio, jota pystyisi hyödyntämään sanojen tarkistuksessa. Toisin kuin iOS-järjestelmässä, Android-käyttöjärjestelmässä ei kuitenkaan ole yleisesti käytettävää sanakirjakokoelmaa. Sanakirjakokoelman puuttuessa olisi pitänyt rakentaa tällainen alusta tai vaihtoehtoisesti löytää vapaassa käytössä oleva sanakirjakokoelma. Android-laitteissa jokaisella näppäimistöllä pitää olla oma sanakirjakokoelmansa, jotta sanojen oikoluku olisi mahdollista.

4.6 Esteettömyys

Esteettömän näppäimistöratkaisun luominen heikkonäköisiä ajatellen jakautuu kahteen tilanteeseen. Ensimmäisessä tapauksessa käyttäjä näkee, mutta heikosti. Tällöin näppäimistön käytettävyyttä voi helpottaa suurentamalla sovelluksen fonttikokoa. Lisäksi kirjainten ja taustan välinen kontrasti on mahdollista toteuttaa niin, että eri näppäimet ovat helposti hahmotettavissa. Tämä onnistuu muokkaamalla värejä ja elementtien kokoa. Toisessa tapauksessa käyttäjä ei näe ollenkaan, jolloin visuaalisia keinoja ei voi käyttää. Kuitenkin esimerkiksi näppäimen koon suurentaminen voi helpottaa oikean näppäimen painallusta.

Viskey ja muut esteettömät näppäimistöt ovat yhteensopivia mobiililaitteiden omien TTS-toimintojen kanssa tai kehittäneet omansa. Näiden puhesynteesitoimintojen käyttö mahdollistaa näppäimistön käytön pelkän kosketuksen ja äänen välityksellä. TTS-toimintojen avulla näppäimistösovellus osaa lausua ääneen käyttäjän painaman merkin. Tällöin käyttäjä siis tietää kuulon perusteella, mitä merkkiä hän painoi.

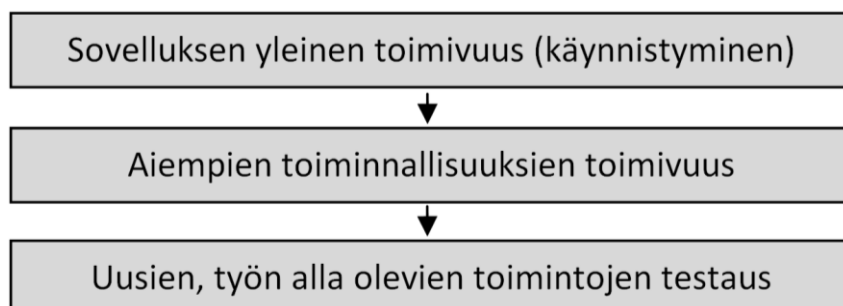
TalkBack on valmis ja laaja esteettömyyspalvelu, jota monet esteettömät sovellukset hyödyntävät. Vaikeuksia tuli yritettäessä saada TalkBack toimimaan optimaalisesti yhdessä näppäimistön kanssa. Jotta TalkBack toimisi oikein, sille täytyy luoda sovelluksen sisälle sellaisia arvoja, joita se osaa hyödyntää. Yksi lisäarvo, jonka pystyy määrittämään elementeille Android Studioissa, on contentdescription-arvo. Sen määrittäminen antaa TalkBackille tekstin, joka lausua näppäintä painaessa. Käytännössä tämä menettelytapa ei kuitenkaan toimi, sillä TalkBack ei lausu näppäinten contentdescription-arvoja.

Toteutettaessa näppäimet napeilla (button), niiden contentdescription-arvo on hyödynnettävissä TalkBack-toiminnolla. Tällöin näppäimistö on mahdollista toteuttaa siten, että TalkBack toimii yhdessä napeista koostuvan näppäimistön kanssa. Ongelma tämän lähestymistavan kanssa on, ettei IME-tyyppinen service osaa hyödyntää toista funktiota findViewById, jonka avulla se osaisi "löytää" oikean napin ja hakea tämän arvon. Viskey:ssä käytetty kehitystapa, jossa taustakuvan päälle luodaan läpinäkyviä näppäimiä, ei siis ole suoraan yhteensopiva TalkBackin kanssa.

5 Sovelluksen testaus

Sovellusta tulee testata kehityksen aikana säännöllisin väliajoin, jotta viimeisimpien sovellukseen tehtyjen muutosten toiminnallisuus ja vaikutukset selviävät käytännössä. Pieniä muutoksia testattiin emulaattorissa, mutta suurempien muutosten jälkeen on hyvä ajaa sovellusta myös fyysisessä laitteessa toiminnallisuuden varmistamiseksi.

Tässä opinnäytetyössä testauksessa oli kolme vaihetta, ja ne suoritettiin aina samalla tavalla. Aluksi testattiin sovelluksen yleistä toimivuutta, kuten käynnistymistä. Sen jälkeen testattiin, ovatko sovelluksen aiemmat toiminnallisuudet edelleen toimivia. Uusia, sillä hetkellä työn alla olevia toimintoja testattiin vasta viimeiseksi. Kuvio 9 selventää testausprosessin etene- mistä tässä opinnäytetyössä.



Kuvio 9: Sovelluksen testausprosessi tässä opinnäytetyössä

6 Tulokset ja analyysit

Opinnäytetyön tarkoituksena oli perehtyä mobiilisovelluksen uudelleenkehitykseen uudelle alustalle liittyviin asioihin. Aiemmin selvitettiin, mitä samanlaisia sovelluksia jo löytyy Android-alustalta ja vertailtiin niiden toiminnallisuuksia jo olemassa olevaan Viskey-sovellukseen. Huomiota kiinnitettiin erityisesti kilpailevien näppäimistöjen esteettömiin ominaisuuksiin. Vertailluista näppäimistösovelluksista Viskey:llä oli kattavimmat esteettömyysratkaisut. (Taulukko 2) Jos Viskey-sovellus kehitetään loppuun Android-alustalle, sillä on hyvät mahdollisuudet menestyä.

Työssä käytiin myös läpi Android-alustalle kehittämiseen käytettyjä työkaluja: Java ohjelmointikieltä ja Android Studio kehittämisalustaa. Javaa käytettiin, koska se on Android Studioissa käytetty ohjelmointikieli. Android Studio valittiin kehittämisalustaksi, koska se on virallinen ja tuettu alusta Android-kehitykselle. Lisäksi se osoittautui helpoksi käyttää.

Yksinkertaisen näppäimistösovelluksen kehittäminen on varsin nopeaa. Kun sovellukselle määrittää riittävän uuden API-tason, kehityksessä voi hyödyntää monia Android Studioon sisäänrakennettuja funktioita. Viralliselta kehityssivustolta löytyy paljon ohjeita eri funktioiden hyödyntämisen. Näppäimistösovelluksen edistyneempien toimivuuksien kehitys oli kuitenkin haastavaa. Edistyneempien toimivuuksien käytöstä ei löytynyt juurikaan esimerkkejä. Lisäksi viralliset ohjeet olivat lyhyet ja olettivat lukijalta aiempaa tuntemusta muista funktioista ja niiden käytöstä. Vaikka erilaisia hyödynnettäviä funktioita on paljon, voi oikean funktion löytä-

minen ja käyttö olla vaikeaa. Lisäksi valmiit funktiot toimivat vain tietyllä tapaa, eivätkä ne välttämättä ole yhteensopivia toistensa kanssa.

7 Yhteenveto

Alun perin Mpaja:n toimeksianto sisälsi mobiilikehitykseen liittyvien tekijöiden tutkimisen lisäksi uuden Viskey-sovelluksen uudelleenkehityksen Android-alustalle. Kokonaisen sovelluksen uudelleenkehitys kuitenkin osoittautui liian haasteelliseksi, ja päätös pelkän prototyypin kehityksestä syntyi yhteisymmärryksessä.

Uusi toimeksianto toteutui ja Mpaja Oy sai tietoa virtuaalinäppäimistön kehityksestä. Play Storessa on jo monia näppäimistösovelluksia, mutta niistä esteettömäksi kehitettyjä on vain rajallinen määrä. Tämän opinnäytetyön yhteydessä Mpaja sai toimivan prototyypin, jota voi käyttää jatkokehitykseen. Lisäksi Mpaja sai tietoa Android Studion käytöstä.

Mpaja Oy:n pitää sovelluksen jatkokehityksessä keskittyä saamaan TalkBack toimimaan yhdessä IME:n kanssa. TalkBack on hyödyllinen ja tehokas esteettömyyspalvelu. Se vaatii kuitenkin rajapintaan spesifiä ohjekoodia toimiakseen yhdessä näppäimistösovelluksen kanssa. Ohjekoodia tarvitaan, jotta TalkBack osaa antaa halutun palautteen käyttäjälle eri tilanteissa.

Lähteet

Green, R. & Zechner, M. 2012. Beginning Android Games. Apress.

Gargenta, M. & Nakamura, M. 2014. Learning Android, 2nd Edition. O'Reilly Media.

Sähköiset lähteet:

Android Central. 2014. What is Google TalkBack?. Viitattu 20.1.2017.
<http://www.androidcentral.com/what-google-talk-back>

Android Developer. Api Guides. Viitattu 10.2.2017
<http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>

Android Developer. Introduction to Android. Viitattu 8.2.2017.
<https://developer.android.com/guide/index.html>

Android Studio. Meet Android Studio. Viitattu 1.3.2017.
<https://developer.android.com/studio/intro/index.html>

Apple Developer. UITextChecker. Viitattu 23.4.2017.
https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UITextChecker_Class/index.html

Apple iTunes. Viskey. Viitattu 23.1.2017.
<https://itunes.apple.com/us/app/viskey/id998451252?mt=8>

Apple. Vision Accessibility: VoiceOver. Viitattu 14.2.2017.
<https://www.apple.com/accessibility/iphone/vision/>

Google Play-kauppa. Haku sanaparilla "accessible keyboard". Viitattu 8.4.2017.
<https://play.google.com/store/search?q=accessible%20keyboard&c=apps&hl=eng>

Java. Java Powers Our Digital World. Viitattu 12.3.2017.
<https://www.java.com/en/about/>

JavaWorld. 2016. Android Studio for beginners, Part 1: Installation and setup. Viitattu 4.2.2017.
<http://www.javaworld.com/article/3095406/android/android-studio-for-beginners-part-1-installation-and-setup.html>

Mpaja. Featuring Viskey. Viitattu 4.2.2017
<http://mpaja.com/frontpage/Viskey>

Stackoverflow 2015. Android - Set TalkBack accessibility focus to a specific view. Viitattu 21.1.2017
<http://stackoverflow.com/questions/28472985/android-set-talkback-accessibility-focus-to-a-specific-view>

Tutsplus 2014. Create a Custom Keyboard on Android. 3.2.2017.
<https://code.tutsplus.com/tutorials/create-a-custom-keyboard-on-android--cms-22615>

Xamarin. Understanding Android API Levels. Viitattu 31.1.2017.
https://developer.xamarin.com/guides/android/application_fundamentals/understanding_android_api_levels/

Kuviot

Kuvio 1: Google Keyboard	9
Kuvio 2: Accessible Keyboard	10
Kuvio 3: Viskey-näppäimistö	11
Kuvio 4: Viskey-näppäimistön vaihtoehtoinen väriys	12
Kuvio 5: Android Studion käyttöliittymä	16
Kuvio 6: Tyylitiedoston sisältämä koodi	17
Kuvio 7: Keyboard.xml-tiedoston sisältöä	18
Kuvio 8: onKeyDown-funktion sisältö	18
Kuvio 9: Sovelluksen testausprosessi tässä opinnäytetyössä	21

Taulukot

Taulukko 1	Google Play:n Näppäimistöjen ominaisuudet.....	9
Taulukko 2	Google Play:n esteettömien näppäimistöjen ominaisuudet ja Viskey.....	10