



SAVONIA

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

LEIKKAUS- JA LABORATORIO- INSTRUMENTTIEN TUNNISTAMINEN

Android sovellus

TEKIJÄ/T: Kristian Saari

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä(t) Kristian Saari	
Työn nimi Leikkaus- ja laboratorioinstrumenttien tunnistaminen	
Päiväys 03.06.2020	Sivumäärä/Liitteet 29
Ohjaaja(t) Lehtori Sami Lahti, Lehtori Pasi Liimatainen	
Toimeksiantaja/Yhteistyökumppani(t) Future Technologies in Education -hanke	
<p>Tiivistelmä</p> <p>Opinnäytetyön toimeksiantaja oli Savonia-ammattikorkeakoulun vetämä Future Technologies in Education-hanke. Työn tarkoituksena oli tuottaa sosiaali- ja terveysalan opiskelijoille sovellus, jonka avulla he voivat tunnistaa leikkaus- ja laboratorioinstrumentteja. Sovelluksen tuli myös tarjota kattava tietokanta näistä instrumenteista. Tuotos tuli toteuttaa Android-sovelluksena, joka toimii puhelimissa ja tableteissa.</p> <p>Sovelluksen kehittämisessä käytettiin TensorFlow-koneoppimisjärjestelmää ja Teachable Machine-koneoppimis-malleja tuottavaa alustaa. Näiden kahden avulla luotiin sovellukseen kuvatunnistustoiminto, jossa esine pystytään tunnistamaan reaaliajassa otetulla kuvalla. Tämän avulla saadaan kyseisen instrumentin informaatio esille. Sovelluksen kehittämisessä käytettiin myös muita valmiita luokkakirjastoja, joiden avulla rakennettiin 2D-koodin tunnistus tilanteisiin, joissa kuvaaminen ei ole vaihtoehto. Sovellukseen luotiin myös kirjasto, jossa voidaan ilman tunnistusominaisuutta tutustua kaikkiin instrumentteihin, jotka löytyvät tietokannasta.</p> <p>Lopputuloksena oli Android-sovellus, joka toimii niin puhelimissa kuin myös tableteissa. Sovellus tarjoaa opettajille työhönsä hyvän ja helppokäyttöisen työkalun, joka on opettajan opetuksen ja opiskelijan oppimisen tukena. Kyseessä on toimiva ensimmäinen versio sovelluksesta, jossa on seuraavat kolme toimintoa: kuvantunnistus, 2D-koodin tunnistus sekä kirjasto. Sovellus ei ole vielä suoraan valmis käyttäjille, vaan sovellukselle pitää ensin tehdä testaukset ja tarvittavat jatkokehitykset.</p>	
Avainsanat Android, Sovellus, Mobiilisovellus, Sovelluskehitys, TensorFlow, Teachable Machine, 2D-koodi	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Kristian Saari			
Title of Thesis Surgery and laboratory instruments identification			
Date	03.06.2020	Pages/Appendices	29
Supervisor(s) Mr. Sami Lahti, Senior Lecturer, Mr. Pasi Liimatainen, Senior Lecturer			
Client Organisation /Partners Future Technologies in Education project			
<p>Abstract</p> <p>The thesis was commissioned by the Future Technologies in Education project led by Savonia University of Applied Sciences. The purpose of this thesis was to create an application for social and health care students that would allow them to identify surgical and laboratory instruments. The application should also offer a comprehensive database of these instruments. The output was to be executed as an Android application that works on both mobile phones and tablets.</p> <p>The application was created by using the TensorFlow machine learning framework and the Teachable Machine platform that generates machine learning models. With the help of these two, an image recognition function was created, where an object can be identified by a real-time image. This function enabled the display of instrument information in the application. Other ready-made class libraries were also utilised to build a 2D code recognition for situations where making an image is not an option. A library was also created in the application that gives the user access to all the instruments that are found in the database without an identification feature.</p> <p>The result was an Android application that works on both mobile phones and tablets. The application offers teachers a good and easy-to-use tool for their work that supports both the teaching and the learning of students. It is a workable first version of an application with the following three functions: image recognition, 2D code recognition and a library. The application is not yet directly ready for users, but it must first be tested and developed further insofar as needed.</p>			
<p>Keywords</p> <p>Android, Application, App, Mobile Application, Application Development, TensorFlow, Teachable Machine, 2D Code</p>			

ESIPUHE

Haluan kiittää Savonian FutureEdu-hanketta ja sen henkilökuntaa. Lisäksi haluan kiittää myös opettajaani Mikko Pääkköstä, jota ilman en olisi saanut harjoittelupaikkaa FutureEdu-hankkeessa. Ilman heitä minulla ei olisi ollut näin mielenkiintoista ja opettavaista opinnäytetyötä.

Haluan kiittää myös vanhempiani, varsinkin äitiäni, joka mahdollisti opiskeluni Kuopiossa ja oli tukena opinnäytetyöni aikana. Haluan myös kiittää siskojani, jotka auttoivat tämän opinnäytetyön viemisen loppuun.

Kuopiossa 03.06.2020.

Kristian Saari

SISÄLTÖ

1	JOHDANTO.....	7
1.1	Toimeksiantaja	7
1.2	Lyhenteet ja määritelmät.....	8
2	KÄYTETTY TEKNIIKAT	10
2.1	Android	10
2.2	Java	11
2.3	XML	11
2.4	Android Studio	12
2.5	SQLite	12
2.6	SQLite Browser	13
2.7	TensorFlow.....	14
2.8	Teachable Machine	14
3	KONEOPPIMINEN	16
3.1	Lyhyt historia.....	16
3.2	Koneoppiminen opinnäytetyössä	16
4	SUUNNITTELU JA MÄÄRITTELY	17
4.1	Sovellus.....	17
4.2	Tietokantasuunnitelma	18
5	SOVELLUS.....	20
5.1	Käyttöliittymä	20
5.2	Ominaisuudet	21
5.2.1	Kuvatunnistus	21
5.2.2	2D-koodi tunnistus	21
5.2.3	Kirjasto.....	22
5.3	Tietokanta	22
6	OPINNÄYTETYÖN TULOS JA JATKOKEHITYS.....	24
6.1	Opinnäytetyön tulos	24
6.2	Jatkokehitys	25
6.2.1	Käyttöliittymä.....	25
6.2.2	Ominaisuudet	26
6.2.3	Kuvatunnistus	26

6.2.4	Kirjasto.....	26
6.2.5	Muut kehityskohdat.....	26
7	POHDINTA	27
	LÄHDELUETTELO	28

1 JOHDANTO

Opinnäytetyön tavoitteena oli saada sosiaali- ja terveysalan opiskelijoille sovellus, jonka avulla he voisivat tunnistaa leikkaus- tai laboratorioinstrumentteja. Sovellus pystyy myös tarjoamaan kattavan tietokannan näistä instrumenteista. Tuotos toteutettiin Android sovelluksena, joka toimii niin puhelimissa, kuin myös tableteissa. Sovelluksella opiskelija pystyy tunnistamaan ja opiskelemaan instrumentit ilman kokoaikaista henkilökohtaista ohjausta. Sovellus tarjoaa opettajille työhönsä hyvän ja helppokäyttöisen työkalun, joka on opettajan, opetuksen ja opiskelijoiden oppimisen tukena.

Opinnäytetyö toteutettiin kehittämistyönä. Ojasalon, Moilasen ja Ritalahden (Ojasalon;Moilainen;& Ritalahti, 2014, ss. 17-19) mukaan kehittämistyössä yhdistyy usein tutkiminen, kehittäminen ja innovaatiotoiminta. Tutkimuksellinen kehittäminen on tieteellisen tutkimuksen ja arkiajatteluun perustuvan kehittämisen välimuoto. Tutkimuksellinen kehittäminen voi saada alkunsa erilaisista lähtökohdista, kuten organisaation kehittämistarpeista tai halusta saada aikaan jotakin muutosta. Tutkimukselliseen kehittämistyöhön kuuluu yleisesti käytännön ongelmien ratkaisu ja uusien ideoiden, käytäntöjen, tuotteiden tai palvelujen tuottaminen ja toteuttaminen.

1.1 Toimeksiantaja

”Savonia on yksi Suomen suurimmista ja monipuolisimmista ammattikorkeakouluista, jolla on kampuksia kolmella paikkakunnalla: Iisalmessa, Kuopiossa ja Varkaudessa. Savonia palvelee lähes 6000 opiskelijan lisäksi aktiivisesti ympäröivää elinkeinoelämää, joka mahdollistaa sekä opetuksen että oppimisen työelämäläheisyyden.” (Savonia, 2020).



KUVA 1. Savonia-ammattikorkeakoulu logo. (Savonia, 2020).

”Savonia kouluttaa sosiaali- ja terveysalan ammattilaisia ja tarjoaa elinikäisen osaamisen palveluita sekä aluekehitystä tukevaa tutkimus-, kehitys- ja innovaatiotoimintaa. Sosiaali- ja terveysalaa opiskelee noin 1 700 opiskelijaa Iisalmessa ja Kuopiossa.” (Savonia, 2020). Tämä työ tehtiin Kuopion kampuksella toimivaan toimipisteeseen.

Bioanalyttikko (Laboratoriohoitaja, bioanalyttikko on tutkintonimike) on kliinisen laboratoriotyön asiantuntija. Laboratoriotutkimukset ja niihin liittyvä näytteenotto ovat ydinosia osaamisalueita. Bioanalyttikko toimii näytteenoton asiantuntijana ja ohjaajana hoitoyhteisössä, sekä tuottaa näytteenottoon liittyvää ohjeistusmateriaalia. Työtehtäviin voi kuulua mm. asiakaspalvelua, näytteiden ottamista ja käsittelyä sekä laboratoriotutkimusten tekemistä ja tulosten raportointia. (Savonia, 2020).

Sairaanhoitaja on osaava ja ammattitaitoinen hoitotyön asiantuntija, jolla on laaja-alainen hoitotyön tuntemus, hallitsee hoitotyössä tarvittavat käsitteet, menetelmät ja tiedot, joiden avulla hän pystyy kriittisesti arvioimaan ja tekemään päätöksiä muuttuvissa ja ennakoimattomissa hoitotyön tilanteissa. Sairaanhoitaja voi toimia erilaisissa tehtävissä ja toimintaympäristöissä muun muassa perusterveydenhuollossa, sairaalassa, yksityisissä palveluissa tai kolmannen sektorin järjestöissä. (Savonia, 2020).

Bioanalyytikkojen ja sairaanhoitajien opintojen laajuus on 210 opintopistettä. Opetussuunnitelman mukaan yksi opiskeluvuosi vastaa 60 opintopistettä, mikä vastaa 1600 tuntia opiskelijan työtä. Opiskelijan työ koostuu mm. lähitunneista, etä- ja itsenäisestä opiskelusta, verkko-opiskelusta ja harjoittelusta. Lisäksi opiskeluun sisältyy työelämälähtöinen tutkimus- ja kehittämistoiminta. (Savonia, 2020); (Savonia, 2020).

Sovellus on toteutettu Savonia-ammattikorkeakoulun vetämään hankkeeseen Future Technologies in Education, jossa kehitetään uusia ratkaisuja koulutukseen ja perehdytykseen erityisesti sairaala-, laboratorio- ja leikkaussaliympäristöihin.

Hankkeen keskeisiä vaiheita ovat:

- 1) suunnitella ja toteuttaa kaksi työelämälähtöistä virtuaalista oppimisympäristöä
- 2) edistää itsenäistä opiskelua, harjoittelua ja perehdytyksiä tukevia prosesseja
- 3) kehittää satelliittikoulutusta, jossa digitaalisten työkalujen avulla voidaan opiskella eri paikkakunnilla ajasta ja paikasta riippumatta sekä teoriaa että simuloida käytännön työtä
- 4) kehittää ja testata virtuaalisten oppimisympäristöjen rakentamisen toimintamalli yhdessä kansainvälisten toimijoiden kanssa. (FutureEdu, 2020).

Hankkeen tuloksena syntyy:

- 1) työelämälähtöinen virtuaalinen oppimisympäristö laboratorioalan opiskelijoiden ja jo työelämässä olevien ja perehtyjien käyttöön
- 2) työelämälähtöinen virtuaalinen oppimisympäristö perioperatiivisen hoitotyön opiskelijoiden ja jo työelämässä olevien ja perehtyjien käyttöön
- 3) virtuaalisten oppimisympäristöjen rakentamisen toimintamalli
- 4) kansainvälisen yhteistyön tuotoksena syntynyt virtuaalinen opintokokonaisuus. (FutureEdu, 2020).

1.2 Lyhenteet ja määritelmät

2D-koodi	2D-koodi eroaa normaalista viivakoodista siten, että 2D-koodi koodataan kaksiulotteisesti eli pituus ja leveys suuntaisesti.
Dp	Density-independent pixels (dp) ovat joustavia yksiköitä, jotka skaalautuvat siten, että niiden mitat ovat yhtenäiset missä tahansa näytössä. Ne tarjoavat joustavan tavan mukauttaa malli eri alustoille. (Material Design, 2020).

FutureEdu	Lyhenne sanoista Future Technologies in Education.
Id	On lyhenne englannin kielen sanasta Identifier, suomeksi tunniste. Hyvä esimerkki tunnisteesta on henkilötunnus.
IDE	On lyhenne Integrated Development Environment, suomeksi ohjelmointiympäristö.
Käyttöjärjestelmä	Käyttöjärjestelmä on ohjelmisto, joka mahdollistaa tietokoneen kommunikation ja toimimisen sovellusten kanssa.
Latenssi	Latenssi on se aika, joka kuluu edestakaiseen matkaan lähettäjältä vastaanottajalle ja takaisin.
Luokkakirjasto	Luokkakirjasto on valmiiksi kirjoitettuja luokkia tai malleja, joita ohjelmoija voi käyttää kehittäessään sovellusta.
Mobiililaite	Mobiililaite on laite, jolla pääsee tietoverkkoon ajasta tai paikasta riippumatta. Mobiililaitteita on mm. matkapuhelimet, tabletit ja kannettavat tietokoneet.
PWA	Lyhenne sanoista Progressive Web App, suomeksi progressiivinen web-sovellus. Ne ovat suunniteltu siten, että ne tuntuvat käytössä kuten natiivisovellukset ja ovat myös asennettavissa puhelimiin.
Regressio	Regressio on tilastollinen menetelmä, jossa määritellään yhden muuttujan ja muiden muuttujien välinen suhde.
SGML	Lyhenne sanoista Standard Generalized Markup Language. Sen avulla voidaan määritellä asiakirjan rakenne ja sen osien looginen suhde.
Sulautettu laite	Sulautettu laite on erikoistarkoitukseen tarkoitettu laite. Se on yleensä integroitu suurempaan kokoonpanoon.
TF	Lyhenne sanasta TensorFlow.

2 KÄYTETYT TEKNIIKAT

2.1 Android

Vuonna 2003 perustettiin yritys nimeltään Android Inc., joka alkoi suunnittelemaan Android käyttöjärjestelmää. Alkuperäisenä tarkoituksena oli tehdä käyttöjärjestelmä ”älykameroille”, jotka pystytään kytkemään tietokoneeseen (Alabaster, 2013). Google osti Android Inc.-yrityksen vuonna 2005. Siitä muutama vuosi eteenpäin 2007 perustettiin 84 teknologia- ja langattomien yritysten yhteenliittymä, jotka alkoivat kehittämään Android-käyttöjärjestelmää yhdessä (Open Handset Alliance, 2007). Vuonna 2008 julkaistiin käyttöjärjestelmän ensimmäinen virallinen versio 1.0 (Android Developers, 2008). Alla olevassa kuvassa (KUVA 3) esitellään vuoden 2008 julkaistusta Android 1.0 vuoden 2016 julkaistuun 7.0 versioon. Tämän hetken viimeisin versio on vuonna 2019 julkaistu Android 10.0.



KUVA 2. Android robot logo. (Android Open Source Project, 2020).

	Platform	API Level	Codename	Features
2008	1.0	1		
	1.1	2		
	1.5	3	Cupcake	
2009	1.6	4	Donut	Maturing app market interface, better voice tools, 800x480
	2.0	5		Better user interface, more screen sizes, more camera functionality, Bluetooth 2.1 support, multi-touch support
	2.0.1	6	Eclair	
	2.1	7		
2010	2.2	8	Froyo	Better performance with just-in-time (JIT) compiler, USB tethering, 720p screen, ability to install apps to the SD card
	2.3	9	Gingerbread	System-wide copy/paste, multi-touch soft keyboard, better native code development, concurrent garbage collection
	2.3.3	10		
	3.0	11		
2011	3.1	12	Honeycomb	Designed for tablets, new soft keyboard, tabbed browsing, redesigned widgets, "holographic UI", interface fragments
	3.2	13		
	4.0	14	Ice Cream	Customizable launcher, screenshot capture, face unlock, Chrome browser, near-field communication, Roboto font
	4.0.3	15	Sandwich	
2012	4.1.2	16		
	4.2.2	17	Jelly Bean	Expandable notifications, Google Now, smoother drawing, improved voice search
2013	4.3	18		
	4.4	19	KitKat	Immersive mode for apps, WebViews based on Chromium, text messaging management, UI transitions framework
	2.3	9	Gingerbread	System-wide copy/paste, multi-touch soft keyboard, better native code development, concurrent garbage collection
	2.3.3	10		
	3.0	11		
2011	3.1	12	Honeycomb	Designed for tablets, new soft keyboard, tabbed browsing, redesigned widgets, "holographic UI", interface fragments
	3.2	13		
	4.0	14	Ice Cream	Customizable launcher, screenshot capture, face unlock, Chrome browser, near-field communication, Roboto font
	4.0.3	15	Sandwich	
2012	4.1.2	16		
	4.2.2	17	Jelly Bean	Expandable notifications, Google Now, smoother drawing, improved voice search
2013	4.3	18		
	4.4	19	KitKat	Immersive mode for apps, WebViews based on Chromium, text messaging management, UI transitions framework
2014	4.4W	20		API for wrist watches (Android Wear)
	5.0	21	Lollipop	Material Design has shadows and animations
	5.1	22		
2015				
	6.0	23	Marshmallow	New way of approving permissions, doze mode puts the device on standby to save power
2016				
	7.0	24	Nougat	Apps can share the screen, virtual reality support

KUVA 3. Android Historia 2008-2016. (Burd, 2016, s. 14).

Opinnäytetyön tuotos toteutettiin Android-käyttöliittymään ja alimmaksi versioksi, jossa sovellusta pystytään käyttämään, määriteltiin Android 7.0. Tämä johtui ainoastaan siitä, että yritettiin ottaa huomioon myös tietoturva ja mitä vanhempi versio, sitä suurempi on tietoturvariski. Parasta olisi ollut valita vielä tuoreempi versio, mutta kuten alla olevasta kuvasta selviää, Suomessa on vielä kohdallainen käyttäjäkunta (8,54 %) myös Android 7.0-versiolla.



KUVA 4. Android markkinaosuudet Suomessa huhtikuussa 2020. (StatCounter, 2020).

2.2 Java

Java on ohjelmointikieli jonka alun perin kehitti James Gosling kollegoidensa kanssa Sun Microsystemsissä 1990-luvun alkupuolella. Alun perin sitä kutsuttiin projektiksi ”Oak”, jonka toteutus oli samanlainen kuin C:n ja C++:n. Nimi Java valittiin myöhemmin aivoriihessä ja nimi perustuu espressopapuun, englanniksi bean. Java 1.0, ensimmäinen versio julkaistiin vuonna 1995 otsikolla ”write once, run anywhere” (”kirjoita kerran, aja missä vain”). (Shankar, 2020). Opinnäytetyössä tuotettiin Android-sovellus, jonka ohjelmointikielenä käytettiin Java-ohjelmointikieltä.



KUVA 5. Java logo. (Wikipedia, 2017).

2.3 XML

Extensible Markup Language (XML) on yksinkertainen, erittäin joustava tekstimuoto, joka on johdettu SGML:stä (ISO 8879). Se on alun perin suunniteltu vastaamaan suurten sähköisten julkaisujen haasteisiin. XML:llä on myös yhä tärkeämpi rooli monenlaisten tietojen vaihdossa verkossa ja muualla. (W3C, 2016).

XML:lä on kaksi pääkäyttöä tarkoitusta. Ensimmäinen on tapa esitellä matalan tason tietoja, esimerkiksi konfigurointitiedostoja. Toinen tapa on lisätä metatietoja asiakirjoihin. (Fawcett;Quin;& Ayers, 2012). Opinnäytetyössä XML:llä oli käytössä sovelluksen ulkoasun luonnissa.

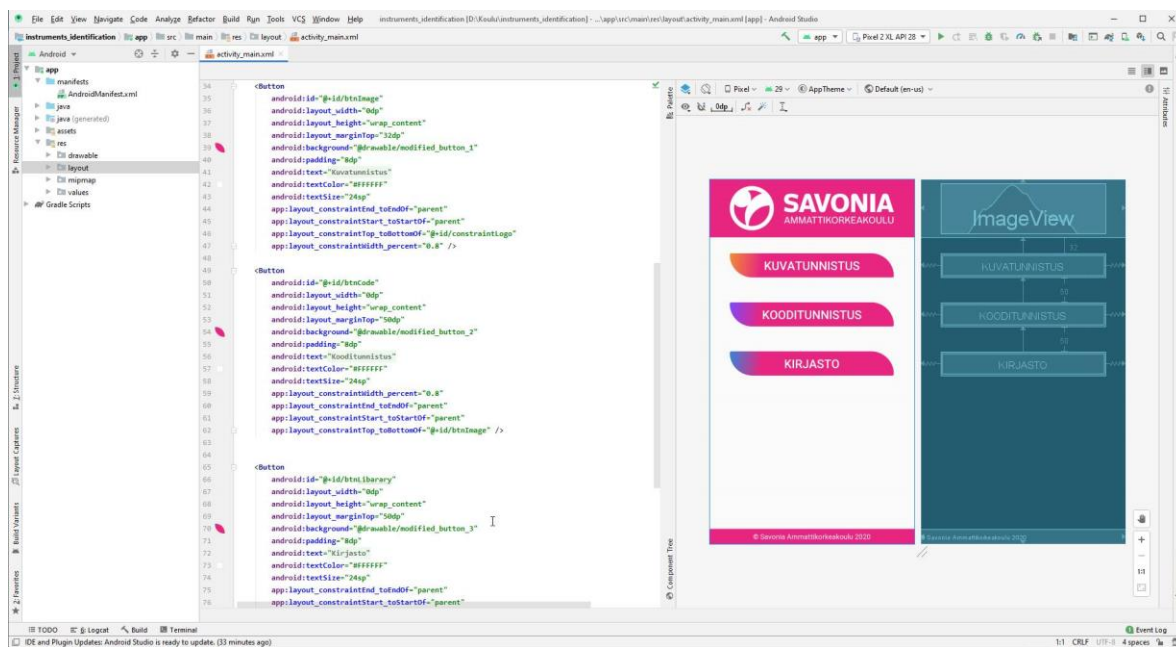
2.4 Android Studio

Android Studio on Androidin virallinen IDE. Se on suunniteltu Androidille nopeuttamaan kehitystä ja auttamaan luomaan korkealaatuisimmat sovellukset jokaiselle Android-laitteelle. (Android Developers, 2020). Android Studio 1.0 julkaistiin 8 joulukuuta 2014 (Eason, 2014).



KUVA 6. Android Studio logo. (Wikipedia Commons, 2020).

Opinnäytetyössä käytettiin koodaamisalustana Android Studiota, jotta sovelluksen kehittäminen oli nopeaa ja korkealaatuista. Alla olevassa kuvassa (KUVA 7) esitellään sovellusikkunaa, jossa näkyy sovelluksen ulkoasu.



KUVA 7. Android Studio.

2.5 SQLite

SQLite on C-kielen kirjasto, joka toteuttaa pienen, nopean, itsenäisen, erittäin luotettavan ja monipuolisen SQL-tietokantamoottorin. SQLite on eniten käytetty tietokantamoottori maailmassa. (SQLite, 2020). SQLite:n koodi on tekijänoikeudeton, joten sitä voidaan käyttää vapaasti mihin tahansa kaupalliseen tai yksityiseen tarkoitukseen. Toisin kuin useimmat muut SQL-tietokannat, SQLite:llä ei ole erillistä palvelinprosessia. SQLite lukee ja kirjoittaa suoraan tavallisiin levytiedostoihin. (SQLite, 2020).



KUVA 8. SQLite logo. (SQLite, 2020).

SQLite:n loi Richard Hipp. Hippin (Litovski & Maynard, 2010, s. 14) mukaan tietokannat olivat vaikeita asentaa ja hallita, joten siitä syntyi idea uudelle SQL-tietokantamoottorille, joka oli pieni, kevyt ja triviaalinen toiminnalle - nollahallinnollinen tietokantamoottori. Tästä ideasta tuli 'SQLite'. Ensimmäinen lähdekoodi kirjoitettiin 29. toukokuuta 2000.

2.6 SQLite Browser

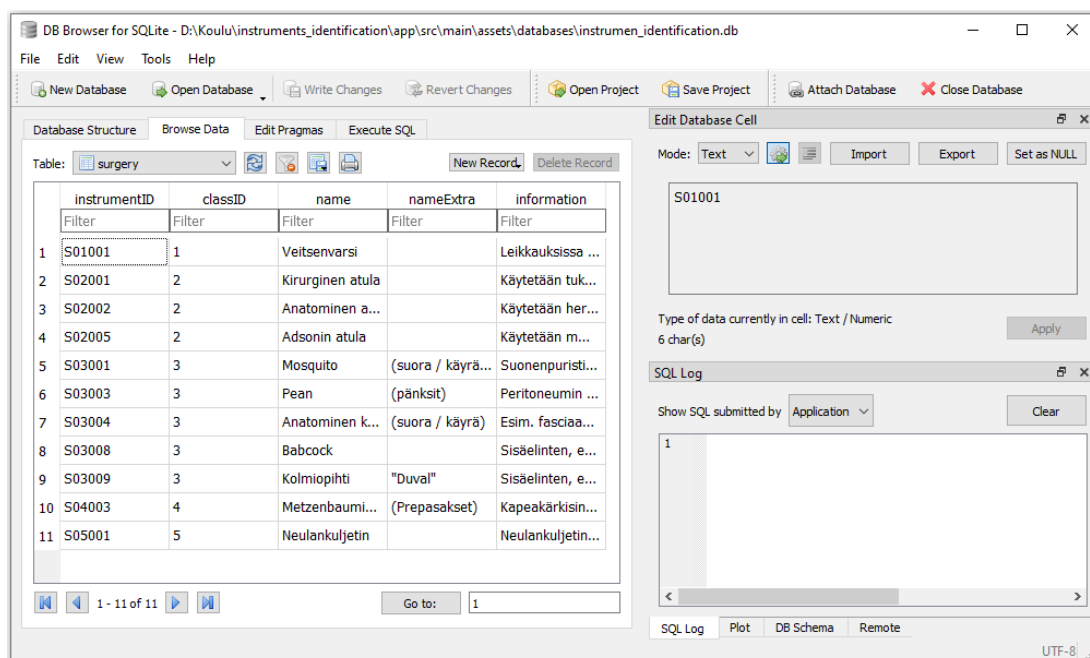
SQLite Browser on alun perin kehittänyt Tabiciro Producoes -yrityksen Mauricio Piacentini Arca-tietokantaselaimena. Alkuperäistä versiota käytettiin vapaana oheistyökaluna Arca Database Xtralle, kaupalliselle tuotteelle, joka upottaa SQLite-tietokantoja joihinkin muihin laajennuksiin pakattujen ja binääritietojen käsittelemiseksi. (SQLite Browser, 2020).

Alkuperäinen koodi leikattiin ja säädettiin yhteensopivaksi SQLite 2.x -standarditietokantojen kanssa vuonna 2014. Tuloksena oleva ohjelma nimettiin uudelleen SQLite Database Browser -sovellukseksi, ja Mauricio julkaisi sen vapaaseen yleiseen käyttöön. (SQLite Browser, 2020).



KUVA 9. SQLite Browser logo. (SQLite Browser, 2020).

Opinnäytetyössä SQLite Browseria käytettiin tietokantojen rakentamiseen ja ylläpitämiseen, johon se oli erittäin suorituskykyinen sovellus. Alla olevassa kuvassa (KUVA 10) esitellään sovellusikkunaa, jossa näkyy avonainen tietokanta.



KUVA 10. SQLite Browser.

2.7 TensorFlow

TensorFlow on Googlen avoimen lähdekoodin kehys, tieteelliseen ja numeeriseen laskentaan, joka perustuu TF-suoritusmallia edustaviin datavirtakaavioihin. TF:ssa käytetyt datavirtakaaviot auttavat hyödyntämään dataa syvällisemmin ja kehittyneemmin, mistä seuraa syväoppimista ja ennakoivia analytiikan malleja. (Karim, 2018, s. 8).



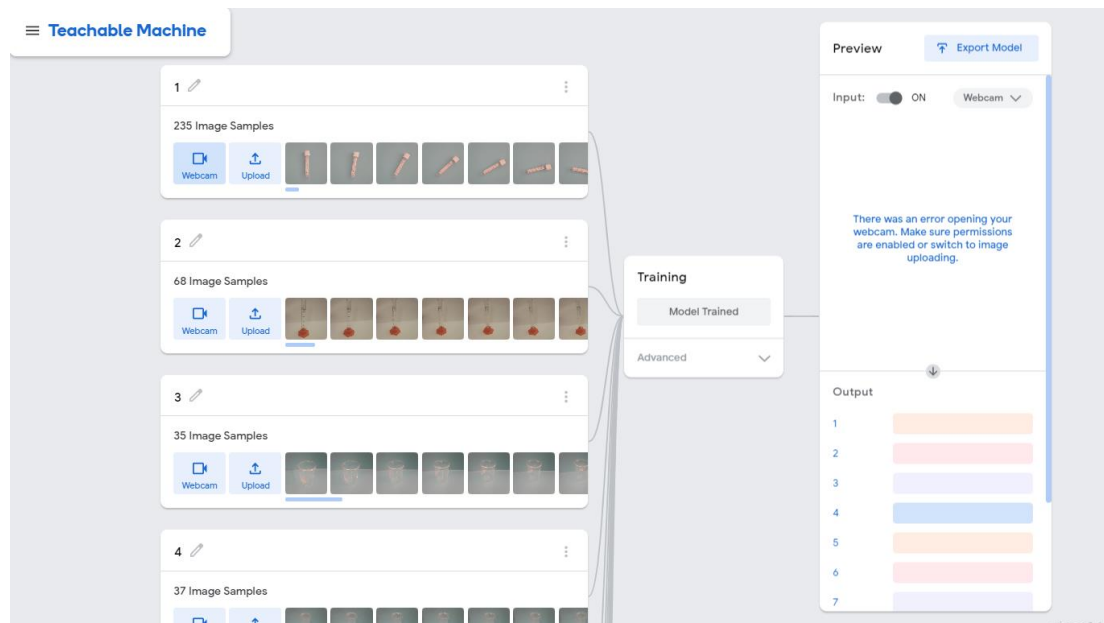
KUVA 11. TensorFlow logo. (TensorFlow, 2020).

TensorFlow Lite on TF:n kevyt ratkaisu mobiililaitteille ja sulautetuille laitteille. Sen avulla voit käyttää koneopetettuja malleja mobiililaitteilla, joilla on pieni latenssi. Voit hyödyntää niitä luokitukseen, regressioon tai muuhun, ilman edestakaista matkaa palvelimelle. (Moroney, 2018) Opinnäytetyössä tultiin käyttämään juuri TF Lite versiota. Vaikka nykyiset mobiililaitteet jaksaisivatkin pyörittää pääversiota, niin työssä otettiin huomioon myös vanhemman mallin mobiililaitteet.

2.8 Teachable Machine

Vuoden 2017 lokakuussa Googlen Creative Lab julkaisi ilmaisen verkkopohjaisen työkalun Teachable Machine, jonka avulla voit harjoitella koneoppimista selaimessasi käyttämällä web-kameraasi syöteenä oman koneoppimisen mallin kouluttamiseen. (Webster, 2018).

Teachable Machine:ssa tuotetaan mallit joko valmiiden kuvien avulla tai tietokoneen web-kameran avulla, joka tuottaa ohjelman avulla suoraan kuvat malliin. Opinnäytetyössä tuotiin valmiit kuvat mallin luontiin. Alla olevassa kuvassa (KUVA 12) esitellään mallin luontia Teachable Machine:ssa.



KUVA 12. Teachable Machine mallin luonti.

3 KONEOPPIMINEN

Koneoppimisella tarkoitetaan tiedon poimimista datasta. Se on leikkauspiste tilasto-, tekoäly- ja tietojenkäsittelytieteiden tutkimusalueilta ja se myös tunnetaan ennustavasta analyysistä ja tilastollisesta oppimisesta. Koneälyohjelmien käyttö on tullut viime vuosina yleiseksi arkielämässä. Automaattisista suosituksista, mitä elokuvia tulisi katsoa, mitä ruokia tulisi tilata tai mitä tuotteita tulisi ostaa, räätälöityihin online-radioihin ja ystävien tunnistamiseen valokuvista, monilla nykyaikaisilla verkkosivustoilla ja laitteilla on koneoppimisen algoritmeja ytimessään. Kun tarkastellaan monimutkaisia verkkosivustoja kuten Facebook, Amazon tai Netflix on erittäin todennäköistä, että jokainen sivuston osa sisältää useita koneoppimismalleja. (Guido & Müller, 2016, s. 1).

3.1 Lyhyt historia

Koneoppiminen perustuu osittain aivosolujen vuorovaikutuksen malliin. Mallin on luonut vuonna 1949 Donald Hebb, josta hän kertoo kirjassa "The Organization of Behavior". Kirja esittelee Hebb'in teorioita hermosolujen jännityksestä ja neuronien välisestä kommunikoinnista. (Foote, 2019).

Tekoälyn todellinen syntymä sellaisena kuin me sen tänään tiedämme, alkoi Alan Turingin julkaisusta "Computing Machinery and Intelligence" vuonna 1950. Tässä tutkimuksessa Turing tutki ajatusta siitä, kuinka voidaan määrittää, pystyvätkö koneet ajattelemaan. Tietenkin, tämä paperi johti jäljittelypeliin, johon osallistuu kolme pelaajaa. Pelaaja A on tietokone ja pelaaja B on ihminen. Kummankin on vakuutettava pelaaja C (ihminen, joka ei näe pelaajaa A tai pelaajaa B), että hän on ihminen. Jos pelaaja C ei pysty johdonmukaisesti selvittämään kuka on ihminen ja kuka ei ole, tietokone voittaa. (Mueller & Massaron, 2016, s. 12).

3.2 Koneoppiminen opinnäytetyössä

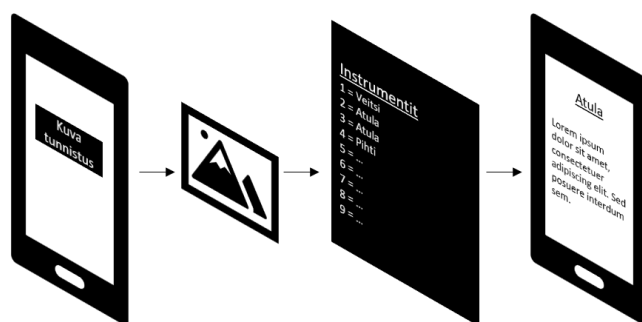
Kuva-analysointi ja tietokonenäkö ovat aina olleet tärkeitä teollisissa ja tieteellisissä sovelluksissa. Matkapuhelimissa ovat tehokkaat kamerat yleistyneet ja internet-yhteyksien myötä myös kuluttajat tuottavat kuvia yhä enemmän. Siksi on olemassa mahdollisuuksia hyödyntää tietokonenäköä paremman käyttökokemuksen tarjoamiseksi uusissa tilanteissa. (Coelho & Richert, 2015, s. 252).

Tässä opinnäytetyössä koneoppiminen oli osana TensorFlow:n hermoverkon luomisessa tietokonenäkölle. Lisäksi Teachable Machine:n verkkopohjaista työkalua käytettiin koneoppimismallien luontiin.

4 SUUNNITTELU JA MÄÄRITTELY

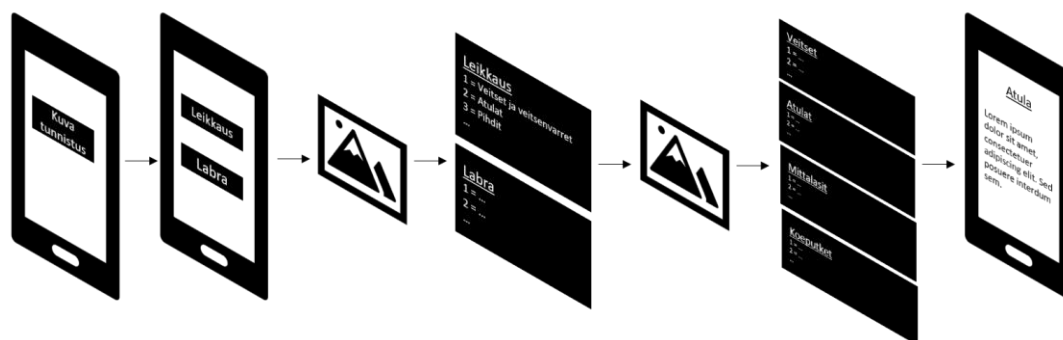
4.1 Sovellus

Ensimmäinen suunnitelma oli tehdä yksikertainen sovellus, jossa sen käynnistyessä voitaisiin valita kuvatunnistustoiminto, nappia painamalla. Tämän jälkeen olisi otettu kuva, joka olisi TF ja yhden mallin avulla tunnistanut instrumentin ja antanut tuloksen. Ongelmana tässä kuitenkin on, että instrumentit, joita tunnistetaan, olivat hyvin saman näköisiä. Alla olevassa kuvassa (KUVA 13) esiintyy ensimmäinen suunnitelma.



KUVA 13. Ensimmäinen sovellus suunnitelma.

Seuraavassa versiossa muokkasin suunnitelmaa lisäämällä kuvien ottamisen kahteen ja lisäsin malleja useampaan kappaleeseen. Nyt ohjelman toiminto menee niin, että ensimmäisessä kohdassa valitaan vieläkin kuvatunnistus-nappi, mutta tämän jälkeen ohjelma meneekin uuteen valintaan. Tässä valitaan joko leikkaussali- tai laboratoriopuoli ja tämän avulla saatiin tarkemmin jaoteltua seuraavan kohdan mallit. Kun ensimmäinen kuva on otettu, TF tunnistaa aikaisemmin valitun puolen perusteella pääluokan leikkauksesta tai laboratoriosta. Toisen kuvan oton tarkoitus on etsiä pääluokan alaisesta välineistöstä juuri oikea väline ja antaa tulos näkyville. Alla olevassa kuvassa (KUVA 14) esiintyy lopullinen suunnitelma kokonaisuudessaan.



KUVA 14. Lopullinen suunnitelma.

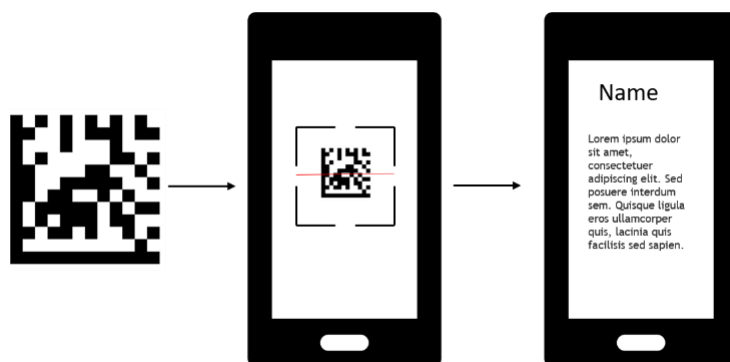
Päädyin valitsemaan TensorFlow:n, koska FutureEdu-hankkeella on yhteistyötä bulgarialaisen koulun kanssa, josta he olivat Savonian vieraina. Vierailun aikana minua pyydettiin esittelemään tulevaa

opinnäytetyötä heille. Heidän joukossaan oli opiskelija, joka oli jo aikaisemmin käyttänyt koneoppimiseen Teachable Machine:a ja suositteli tätä ohjelma kokonaisuutta minulle. Samalla keskustelimme, että voisimme tehdä tulevaisuudessa yhteistyötä ja tuottaa julkaisun tietotekniikan lehtiin.

Suunnitelmana oli myös ottaa selvää varajärjestelmästä, jos kuvantunnistus ei toimisi. Tähän jo silloin suunniteltiin QR-koodia, jota mietittiin laitettavaksi välineisiin. Mutta tiedonhaun avulla selvisi, että Data Matrix-koodi olisi parempi ja se toimii pienempänä kuin QR-koodi. Alla olevassa kuvassa nähdään Data Matrix:in ja QR-koodien eroavaisuudet. Alla olevasta kuvassa on silloinen suunnitelman luonnos.



KUVA 15. Data Matrix ja QR.



KUVA 16. 2D-koodin tunnistus suunnitelma.

Käyttöliittymän ulkoasun suunnittelun ja toteutuksen kanssa oli minulla vapaat kädet. Päätin ottaa sovelluksen värimaailmaan esimerkkiä Savonian logosta ja verkkosivuista. Halusin lisäksi pitää ulkoasun yksinkertaisena, että sovelluksesta ei tule liian sekava.

4.2 Tietokantasuunnitelma

Päätin, että tietokanta tulee olemaan hyvin yksinkertainen, kun sovelluksessa ei tulla tietokannasta hakemaan kuin instrumentin tiedot. Aluksi suunnitelmana oli tehdä vain yksi taulu, johon olisi tullut kaikki tiedot. Kuitenkin instrumentteja tulee sovelluksessa olemaan jopa satoja, niin se voisi olla liian sekava ja vaikeakäyttöinen. Lopulta päädyin tekemään kahden taulun tietokantasuunnitelman, niin että kummallekin, leikkaus- sekä laboratoriopuolelle, tulisi oma taulu.

Tämän jälkeen oli suunniteltava instrumenttien id-tunnisteet ja niiden toiminta. Instrumenteilla on yrityksien kautta omat tunnukset, mutta saattaa olla, että tunnukset voivat olla samoja eri instru-

menteilla, joten tunnuksina ei voinut käyttää instrumenteissa jo olevia tunnuksia. Lopulta loin jokaiselle instrumentille oman tunnuksen ja esittelen sen toteutuksen, jonka avulla id-tunnukset toteutettiin alla olevassa kuvassa (KUVA 17).



Toimiala
 S = Leikkaus (Surgery)
 L = Laboratorio (Laboratory)

Luokat

Leikkaus	Laboratorio
1 = Veitset ja veitsenvarret	
2 = Atulat	
3 = Pihdit	
4 = Sakset	
5 = Neulankuljettimet	
6 = Haat ja spaattelit	

Järjestysnumero

KUVA 17. Instrumenttien id toteutus.

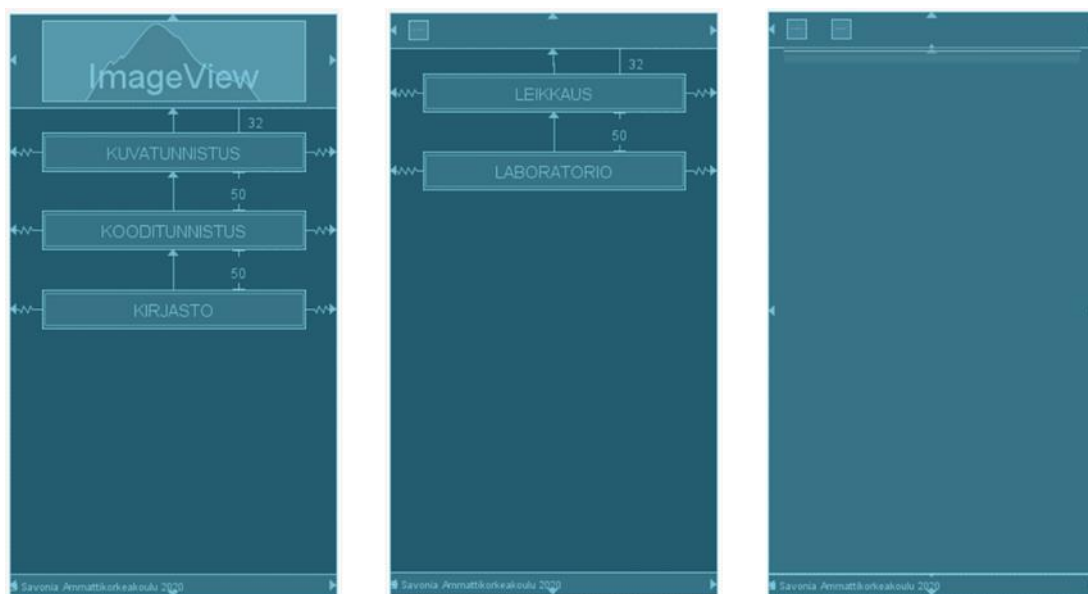
5 SOVELLUS

Sovellus rakentuu kolmesta eri alueesta. Kuvatunnistus, jossa puhelimen oman kameran kautta otettavan kuvan avulla tunnistetaan edessä oleva instrumentti. Toisena on, 2D-koodien avulla oleva tunnistus, tämäkin toimii puhelimen oman kameran avulla. Kolmas on kirjasto, jossa voidaan käydä katsomassa kaikki instrumentit, jotka ovat tietokannassa ja tutustua näihin. Sovellus rakentuu kummankin suunnitelman mukaan yksinkertaisemman ja monipuolisemman mallin mukaan.

5.1 Käyttöliittymä

Sovelluksen käyttöliittymän värimaailma pysyi Savonian väreinä, jota jo suunnitelma vaiheessakin mietittiin. Käyttöliittymästä saatiin tehtyä selkeä ja helppo kulkuinen, jotta sen käyttäminen olisi mahdollista jokaiselle käyttäjälle.

Käyttöliittymä on rakennettu niin, että sitä voisi käyttää puhelimilla ja tableteilla. Tämän takia jouduttiin ottamaan huomioon näyttöjen eri koot. Sovellus rakentuu niin, että esimerkiksi napit muuttuvat kuvaruudun koon mukaan eli niille syötetään koko prosentteilla ei kiinteällä numerolla. Samalla tavalla mukautuu myös yläpalkit, pääsivulla oleva logo. Sovelluksesta löytyy kiinteitä arvoja. Niitä ovat muun muassa nappien välinen etäisyys, sekä alapalkille on annettu kiinteä korkeus 30dp. Alla olevassa kuvassa (KUVA 18) on sovelluksen niin sanotut pohjapiirroksat. Näistä näkee ne arvot, jotka ovat kiinteitä arvoja ja ne on annettu dp-arvoina. Pääsivun yläpalkki on 16 prosenttia ruudun koosta ja logon on 80 prosenttia yläpalkin koosta. Muiden sivujen yläpalkin koko on 6 prosenttia ruudun koosta.



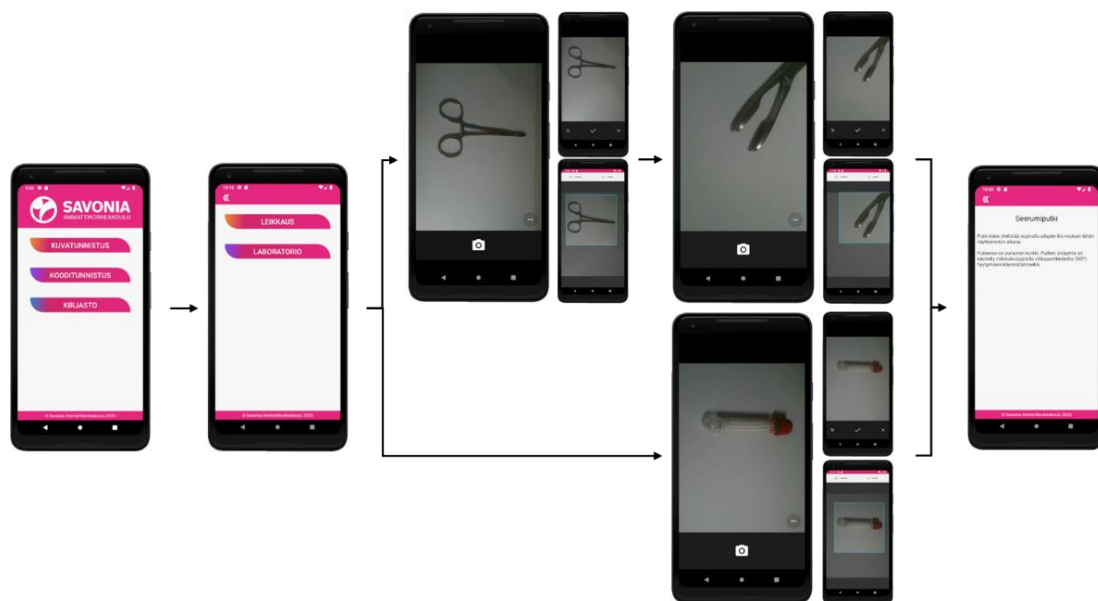
KUVA 18. Sovelluksen pohjapiirroksat.

5.2 Ominaisuudet

Ominaisuuksia sovelluksesta löytyy kolme: kuva- tai kuvilla ja 2D-koodilla toimivat instrumenttien tunnistukset sekä kirjasto, josta voi selata kaikkia instrumentteja.

5.2.1 Kuvatunnistus

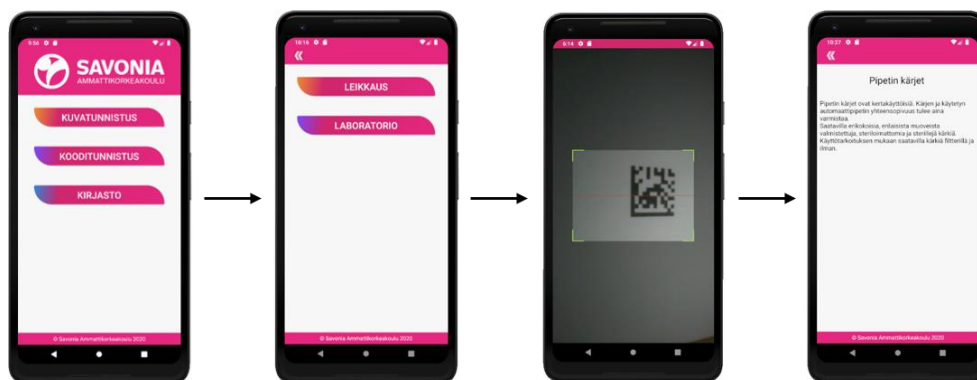
Kuvatunnistaminen on pääosin toteutettu TensorFlow Lite-luokkakirjaston avulla. Tähän sovellukseen tämä on toteutettu kahdella eri tavalla. Leikkauspuolen toteutuksessa otetaan instrumentista kaksi kertaa kuva. Ensimmäisessä kuvassa, kuvaa verrataan pääluokkien malliin ja sen avulla saadaan selville pääluokka. Toisessa kuvassa, kuvaa verrataan kaikkiin niihin instrumentteihin, jotka kuuluvat tähän pääluokkaan. Tämä toisen kuvan tarkoitus on helpottaa ja parantaa instrumenttien tunnistettavuutta. Laboratoriopuolen instrumentit on helpompi tunnistaa, joten tässä on päädytty vain yhteen kuvan ottamiseen.



KUVA 19. Kuvatunnistus.

5.2.2 2D-koodi tunnistus

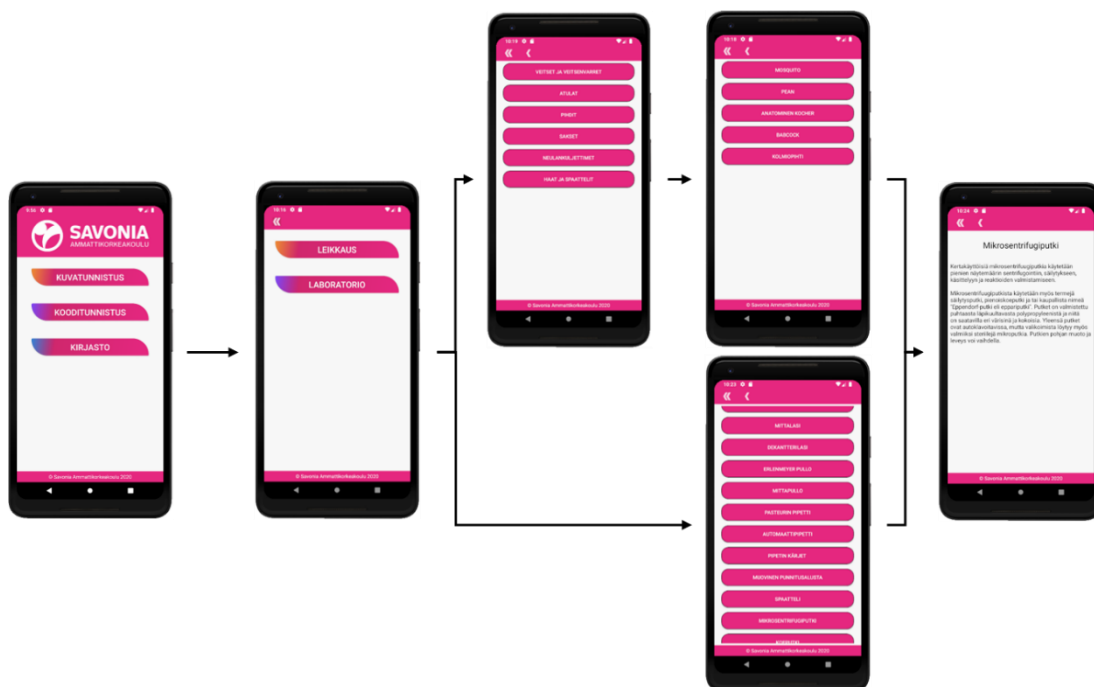
2D-koodi tunnistus on toteutettu käyttämällä kahta valmista luokkakirjastoa, ZXing Barcode Scanner (me.dm7.barcodescanner:zxing:1.9.13) ja Dexter (com.karumi:dexter:5.0.0). ZXing Barcode Scanner tuottaa itsessään tämän koodin tunnistuksen, jonka avulla tunnistetaan erilaiset 2D-koodit. Dexter on taas luokkakirjasto, jonka avulla yksinkertaistetaan käyttöoikeuksien pyytämistä, tässä tapauksessa kameran käyttöä.



KUVA 20. 2D-koodin tunnistus.

5.2.3 Kirjasto

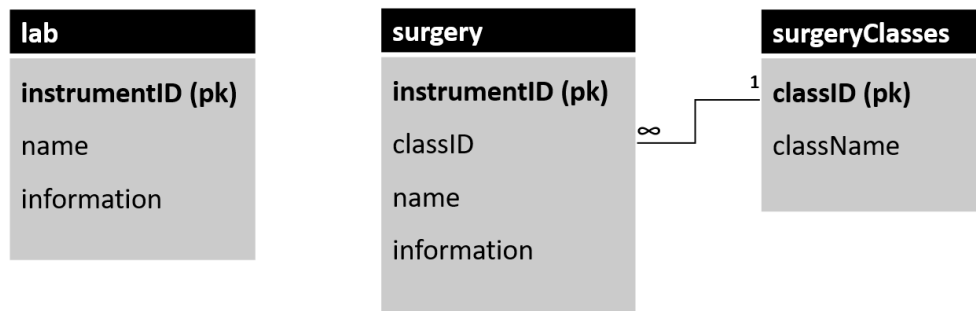
Kirjasto on hyvin yksinkertainen. Kirjastossa pystyy tutustumaan kaikkiin instrumentteihin alakohtaisesti. Kirjastossa on helppo liikkua. Ensimmäisessä kohdassa valitaan ala ja toinen kohta on vain leikkaus puolen instrumenteille, siinä valitaan instrumenttien pääluokka. Tämä löytyy vain leikkauspuolen instrumenteissa, koska leikkaus ja laboratorion puolen tietokanta on suunniteltu eri tavalla. Viimeisessä kohdassa valitaan se instrumentti, jota halutaan tarkastella lähemmin. Tämä tulostaa instrumentin nimen ja sen tiedot ruudulle.



KUVA 21. Kirjaston toiminta.

5.3 Tietokanta

Valmistunut tietokanta rakentui kolmesta taulusta, vaikka suunnitelmana oli rakentaa kahden taulun tietokanta. Ensimmäinen taulu sisältää vain laboratorion puolen instrumentit. Toiseen ja kolmanteen tauluun rakentuvat leikkauspuolen instrumenttien tiedot. Näistä ensimmäisestä taulusta löytyy instrumenttien tiedot ja toisesta taulusta löytyy instrumenttien pääluokat.



KUVA 22. Tietokantamalli.

6 OPINNÄYTETYÖN TULOS JA JATKOKEHITYS

6.1 Opinnäytetyön tulos

Opinnäytetyön tarkoituksena oli saada sosiaali- ja terveysalan opiskelijoille sovellus, jonka avulla he voisivat tunnistaa leikkaus- ja laboratorioinstrumentteja. Tuloksena muodostui hyvä ensimmäinen versio tulevaisuuden instrumenttien tunnistus ohjelmasta. Ohjelmassa on kolme toimivaa kokonaisuutta, jotka toimivat hyvin yhteen. Käyttöliittymä on pyritty suunnittelemaan ja toteuttamaan yksinkertaisena ja toimivana.

Kuvatunnistustoiminto toteutettiin TensorFlow-koneoppimisalustalla. Tämän sovelluksen toiminto on toteutettu kahdella eri tavalla. Leikkaus- ja laboratoripuolen toteutus ovat kummatkin erilaiset. Leikkauspuolen toteutuksessa, otetaan kaksi kertaa kuva instrumentista. Ensimmäisessä kuvassa, otetaan koko instrumentin kuva, tätä kuvaa verrataan pääluokkien malliin, jonka avulla saadaan selville tämän instrumentin pääluokka. Toisessa kuvassa otetaan instrumentin päästä tai kärjestä kuva, jota verrataan kaikkiin niihin instrumentteihin, jotka kuuluvat äsken saatuun pääluokan malliin. Laboratoripuolen instrumentit on taasen helpompi tunnistaa joten, tässä on päädytty vain yhden kuvan ottamiseen ja yhteen malliin, jossa on kaikki instrumentit.

Koodintunnistustoiminnon toteutuksessa käytettiin kahta valmista luokkakirjastoa. Toinen näistä luokkakirjastoista tuottaa itsessään tämän koodin tunnistuksen, jonka avulla tunnistetaan erilaiset 2D-koodit. Toinen luokkakirjasto on helpottamassa toimintojen käyttöoikeuksien pyytämistä. Tässä toiminnossa se on apuna kameran käyttöluvan pyytämiseen.

Kolmas ja viimeinen toiminto on kirjasto, jossa pystytään tutustumaan kaikkiin instrumentteihin alakohtaisesti. Kuten kuvatunnistus toiminnossa, kirjasto on toteutettu kahdella eri tavalla, niin että leikkaus- ja laboratoripuolen toteutukset ovat erilaiset. Puolen valitsemisen jälkeen leikkauspuolella joudutaan ensimmäiseksi valitsemaan pääluokka, jonka jälkeen nähdään tämän pääluokan alla olevat instrumentit. Laboratoripuolella nähdään heti puolen valitsemisen jälkeen kaikki instrumentit. Syynä tähän on tietokantataulujen erilainen toteutus.

Opinnäytetyön tuloksena syntyi myös kolme ohjekirjaa asiakkaalle heidän pyynnöstään. Ohjekirjojen tarkoituksena on auttaa asiakasta jatkamaan sovelluksen kehittämistä, opinnäytetyön jälkeen. Ohjekirjat luotiin PowerPoint-ohjelmistoa käyttäen.

Ensimmäisessä ”Tietokannat” ohjekirjassa opetetaan:

- Tietokannan avaaminen
- Taulujen luominen
- Tietojen muokkaaminen ja poistaminen
- Uusien tietojen lisääminen.

Toisessa ”Teachable Machine ja mallin päivitys sovellukseen” ohjekirjassa opetetaan:

- Teachable Machine toimintaa
- Uuden mallipohjan luonti
- Vanhan mallin tuominen
- Tietojen lisääminen, poistaminen tai muokkaaminen
- Mallipohjan tallentaminen
- Mallin opetus.

Kolmannessa ”Asennuspaketin teko” ohjekirjassa opetetaan:

- Miten Android Studiossa luodaan asennuskuvake sovelluksesta.

Opinnäytetyön tuloksena syntyi sovelluksen kuvake, joka on alla olevassa kuvassa (KUVA 23). Tätä voi myös käyttää sovelluksen logona.



KUVA 23. Sovelluksen kuvake.

6.2 Jatkokehitys

Sovellukset eivät ole ikinä valmiita ja niihin löytyy aina lisättävää tai kehitettävää. Tämän sovelluksen kanssa olisi vaihtoehtoisesti kaksi eri jatkokehitysversiona.

Jatkettaisiin vain tämän Android sovelluksen kehittämistä. Paranneltaisiin sovelluksen toimintaa ja puhdistettaisiin koodia turhista päällekkäisyyksistä. Koko sovellus on rakenneltu monista sovelluspaloista, jotka on kasattu yhdeksi sovellukseksi. Voi olla, että löytyy joku parempi sovellusosa, joka hoitaa tehtävänsä paremmin ja se vaihdetaan tämänhetkisen palan paikalle.

Toisena vaihtoehtona olisi, että lähdettäisiin rakentamaan uutta PWA-sovellusta. Tämä toisi lisäarvoa sovellukselle, kun sitä voi käyttää useammalla eri laitteella. TF on kuitenkin huomattu toimivaksi ja helppokäyttöiseksi koneoppimissovellukseksi. Se tarjoaa myös JavaScript kirjaston, jonka takia ei tarvitsisi vaihtaa koneoppimissovellusta.

6.2.1 Käyttöliittymä

Käyttöliittymä on yksinkertainen, mutta jos tulevaisuudessa tullaan lisäämään tai muokkaamaan muita kohtia, joudutaan muokkaamaan myös käyttöliittymää. Kun sovellusta päästään testaamaan

ja sitä kautta saadaan asiakkailta käyttökokemuksia, voidaan niiden avulla muokata sovellusta asiakasystävällisempään suuntaan. Vaikka insinöörin mielestä käyttöliittymä on helppo ja yksinkertainen, käyttäjän mielestä se voi olla hankala käyttää.

6.2.2 Ominaisuudet

Asiakas on keksinyt jo nyt sovelluksen kehitysvaiheessa uusia ominaisuuksia. Sovelluksen alusta saakka siihen on mietitty peliominaisuutta, mutta tätä ei nyt ole vielä sisällytetty sovellukseen. Toinen isompi ominaisuus olisi instrumenttien 3D-mallien tai 360°-kuvan lisääminen, jotta tutkittavaan instrumenttiin voi tutustua paremmin. Tätä voisi myös käyttää itse pelissä, jossa voisi tämän kuvan tai mallin avulla esittää instrumenttia ja käyttäjä voisi yrittää tunnistaa esinettä.

Lisättävänä ominaisuutena voisi olla sisäänkirjautuminen, jonka voisi liittää jo valmiisiin Savonian oppilaiden käyttäjätunnuksiin. Tällä saataisiin ohjelma vain sisäiseen käyttöön, eikä sovellus leviäisi ilmaisena sovelluksena puhelimesta toiseen. Tässä on kuitenkin tärkeintä ottaa huomioon tietoturva. Esimerkiksi miten salataan liikkuva data, jottei käyttäjätunnukset ja salasanat leviä vieraisiin käsiin.

6.2.3 Kuvatunnistus

Kuvatunnistusta voidaan varmasti parantaa monellakin osa-alueella. Ensimmäinen ja suurin alue on mallien rakentaminen. Tätä varten kannattaisi tutustua myös muihin keinoihin kuin Teachable Machine:en. Toinen on kuvien lisääminen tai kokonaan uusien kuvien ottaminen instrumenteista, joita mallinuksissa käytetään. Malleista tulee vain aina parempia mitä enemmän kuvia mallissa on. Tässä olisi myös hyvä lisätä vielä eri taustoja, jotta tämäkin tulisi mallinnuksessa paremmin huomioitua.

6.2.4 Kirjasto

Kirjastossa voi olla parannettavaa itse käyttöliittymässä, kuin toiminnassakin. Lisättävänä ominaisuutena on ainakin hakukoneen teko, jonka avulla käyttäjä pystyy hakemaan helpommin tietyn tai tietyt instrumentit, joita hän haluaa tarkastella.

6.2.5 Muut kehityskohdat

Sovelluksella ei ole tällä hetkellä kuin työnimi "Instrument Identification", joten asiakas voi suunnitella uuden nimen niin halutessaan. Sovelluksella on kuvake, jota voi käyttää logona. Sitä voisi parantaa ja saada siihen visuaalisempaa ilmettä. Tässä olisi hyvä käyttää Savonia-ammattikorkeakoulun suunnittelupuolen opiskelijoita. Heiltä voisi saada myös apua sovelluksen ulkonäön kanssa. Olisi myös hyvä teetättää uusi opinnäytetyö käyttämällä Savonian sosiaali- ja terveystieteiden opiskelijoita. Opinnäytetyössä kannattaa tehdä tutkimus, miten sovellus toimii oppimisen näkökulmasta ja miten sitä voisi myös parantaa tästä samasta näkökulmasta nähtynä.

7 POHDINTA

Opinnäytetyössä innovoitiin uudenaikainen sovellus, sosiaali- ja terveysalan leikkaus- ja laboratorio- koulutusten opetuksen tueksi. Opinnäytetyö eteni nopeasti toimeksiannon jälkeen. Sovelluksen suunnitteleminen ja luominen kehittyi yhtäaikaaisesti. Aiheena opinnäytetyö oli erittäin mielenkiintoinen, vaikkakin melko haastava. Koneoppiminen oli aiheena minulle melko uusi, joten asian perehtymiseen meni paljon aikaa. Opintojen aikana koneoppimista käsiteltiin teoriapohjaisesti Helsingin yliopiston tekoälykurssin ”Elements of AI” luennolla.

Sovelluksen haastavin osuus oli opettaa instrumentit koneelle. Koneopetuksen toimiminen vaati, että yhdestä instrumentista tulisi olla paljon kuvia. Kuvia kannattaisi olla mahdollisimman monesta eri kuvakulmasta sekä lisäksi erilaisilla taustoilla. Näin ollen kone ymmärtäisi ne paremmin ja virheiden määrä laskisi.

Yksi nykyajan opettajien haasteista on opiskelijoiden motivointi opiskelemaan, sekä heidän jakama aika jokaiselle opiskelijalle. Tämän sovelluksen tarkoitus on juuri helpottaa näissä haasteissa. Sovellus tarjoaa konenäön avulla helpotusta instrumenttien opiskeluun henkilökohtaisesti. Uusi teknologia on myös motivaatiokeino nykyisessä digitalisoituvassa yhteiskunnassa. Tämän sovelluksen avulla tuodaan helpotusta opettajien tämänhetkisiin haasteisiin.

Tämän opinnäytetyö prosessin aikana huomasin, että sovellus olisi ehkä ollut parasta luoda PWA-sovelluksena, jolloin sovellusta olisi voinut käyttää suuremmassa laite määrässä. Jos sovellus olisi luotu PWA-sovelluksena, sitä olisi voitu myös käyttää Microsoftin Hololenseissä ja Applen iPhoneissa, sekä tietokoneilla. Se jäänee jatkokehitysideaksi. Opinnäytetyön aikana sovellusta ei ehditty testata, joten se rajattiin pois tästä opinnäytetyöstä. Asiakas voi tehdä testaukset itse, jolloin he saavat käyttökokemuksista hyvää tietoa. Niiden perusteella he voivat korjata mahdolliset virheet ja kehittää sovellusta, jos tarve vaatii.

Itse kehityin opinnäytetyön aikana ymmärtämään kuinka TensorFlow toimii ja miten tätä käytetään erilaisissa sovelluksissa. Kaikista opinnäytetyön aikana opituista asioista on varmasta hyötyä tulevaisuudessa, ja erityisesti työmarkkinoilla.

LÄHDELUETTELO

- Alabaster, J. (16. 04 2013). Haettu 25. 04 2020 osoitteesta PCWorld:
<https://www.pcworld.com/article/2034723/android-founder-we-aimed-to-make-a-camera-os.html>
- Android Developers. (23. 09 2008). *Android Developers Blog*. Haettu 03. 05 2020 osoitteesta <https://android-developers.googleblog.com/2008/09/announcing-android-10-sdk-release-1.html>
- Android Developers. (22. 01 2020). *Android Developers: Android Studio features*. Haettu 05. 05 2020 osoitteesta <https://developer.android.com/studio/features>
- Android Open Source Project. (06. 01 2020). *Android source*. Haettu 03. 05 2020 osoitteesta <https://source.android.com/setup/start/brands>
- Burd, B. (2016). *ava Programming for Android Developers for Dummies*. New Jersey: John Wiley & Sons, Inc.
- Coelho, L. P.; & Richert, W. (2015). *Building Machine Learning Systems with Python*. Packt Publishing, Limited: Packt Publishing Ltd.
- Eason, J. (08. 12 2014). *Android Developers Blog*. Haettu 05. 05 2020 osoitteesta <https://android-developers.googleblog.com/2014/12/android-studio-10.html>
- Fawcett, J.; Quin, L.; & Ayers, D. (2012). *Beginning XML*. Indianapolis: John Wiley & Sons, Inc.
- Foot, K. D. (26. 03 2019). *A Brief History of Machine Learning*. Haettu 30. 03 2020 osoitteesta Dataversity:
<https://www.dataversity.net/a-brief-history-of-machine-learning/>
- FutureEdu. (2020). *Lyhyesti suomeksi*. (Savonia) Haettu 29. 3 2020 osoitteesta
<https://futureedu.savonia.fi/lyhyesti-suomeksi/>
- Guido, S.; & Müller, A. (2016). *Introduction to Machine Learning with Python*. USA: O'Reilly Media, Inc.
- Karim, R. (2018). *TensorFlow: Powerful Predictive Analytics with TensorFlow*. Birmingham: Packt Publishing, Limited.
- Litovski, I.; & Maynard, R. (2010). *A Mobile Developer's Guide to SQLite*. United Kingdom: John Wiley & Sons Inc.
- Material Design. (2020). *Material Design: Pixel density*. Haettu 02. 06 2020 osoitteesta
<https://material.io/design/layout/pixel-density.html>
- Moroney, L. (30. 03 2018). Noudettu osoitteesta TensorFlow Blog: <https://blog.tensorflow.org/2018/03/using-tensorflow-lite-on-android.html>
- Mueller, J. P.; & Massaron, L. (2016). *Machine Learning for Dummies*. USA: John Wiley & Sons, Inc.
- Ojasalon, K.; Moilainen, T.; & Ritalahti, T. (2014). *Kehittämistyön menetelmät: uudenlaista osaamista liiketoimintaan*. Sanoma Pro Oy.
- Open Handset Alliance. (11 2007). *Industry Leaders Announce Open Platform for Mobile Devices*. Haettu 25. 04 2020 osoitteesta [openhandsetalliance.com: http://www.openhandsetalliance.com/press_110507.html](http://www.openhandsetalliance.com/press_110507.html)
- Savonia. (2020). *Logot*. Haettu 26. 04 2020 osoitteesta Savonia-ammattikorkeakoulu:
<https://portal.savonia.fi/amk/fi/tutustu-savoniaan/organisaatio-ja-johtaminen/savonian-viestinta/logot>
- Savonia. (2020). *Savonia-ammattikorkeakoulu opetussuunnitelmat*. Noudettu osoitteesta Bioanalyytikon tutkinto-ohjelma, opintojen rakenne: <https://portal.savonia.fi/amk/node/209?yks=KS&krtid=1343&tab=3>
- Savonia. (2020). *Savonia-ammattikorkeakoulu, amk- ja yamk-tutkinnot, kevaan yhteishaku*. Haettu 31. 05 2020 osoitteesta Sairaanhoidaja (AMK), päivätoteutus: <https://portal.savonia.fi/amk/fi/hakijalle/amk-ja-yamk-tutkinnot/kevaan-yhteishaku/sairaanhoidaja-amk-paivatoteutus-kuopio>
- Savonia. (2020). *Savonia-ammattikorkeakoulu, amk- ja yamk-tutkinnot, kevaan yhteishaku*. Haettu 31. 05 2020 osoitteesta Bioanalyttikko (AMK), päivätoteutus: <https://portal.savonia.fi/amk/fi/hakijalle/amk-ja-yamk-tutkinnot/kevaan-yhteishaku/bioanalyttikko-amk-paivatoteutus>

- Savonia. (2020). *Savonia-ammattikorkeakoulu, opetussuunnitelmat*. Haettu 31. 05 2020 osoitteesta <https://portal.savonia.fi/amk/node/209?yks=KS&krtid=1280&tab=3>
Sairaanhoitajan tutkinto-ohjelma, opintojen rakenne:
- Savonia. (2020). *Savonia-ammattikorkeakoulu*. Haettu 24. 02 2020 osoitteesta <https://portal.savonia.fi/amk/>
- Savonia. (2020). *Savonia-ammattikorkeakoulu*. Haettu 30. 03 2020 osoitteesta <https://portal.savonia.fi/amk/fi/tutustu-savoniaan/opiskelu-koulutusaloilla/sosiaali-ja-terveysala>
- Shankar, R. (09. 04 2020). *Hackr.io Blog: What is Java?* Haettu 03. 05 2020 osoitteesta <https://hackr.io/blog/what-is-java>
- SQLite. (2020). Haettu 25. 04 2020 osoitteesta What Is SQLite?: <https://www.sqlite.org/>
- SQLite. (2020). Haettu 25. 04 2020 osoitteesta About SQLite: <https://www.sqlite.org/about.html>
- SQLite Browser. (2020). *SQLite Browser: About*. Haettu 05. 05 2020 osoitteesta <https://sqlitebrowser.org/about/>
- StatCounter. (04 2020). *statcounter*. Haettu 08. 05 2020 osoitteesta <https://gs.statcounter.com/android-version-market-share/mobile-tablet/finland>
- TensorFlow. (2020). *TensorFlow*. Haettu 03. 05 2020 osoitteesta https://www.tensorflow.org/extras/tensorflow_brand_guidelines.pdf
- W3C. (10. 11 2016). *W3C Information and knowledge domain*. Haettu 11. 05 2020 osoitteesta <https://www.w3.org/XML/>
- Webster, B. (17. 04 2018). *Designing (and Learning From) a Teachable Machine*. Haettu 26. 04 2020 osoitteesta Google Desing: <https://design.google/library/designing-and-learning-teachable-machine/>
- Wikipedia. (30. 04 2017). Haettu 03. 05 2020 osoitteesta https://fi.wikipedia.org/wiki/Tiedosto:Javan_logo.svg
- Wikipedia Commons. (27. 02 2020). *Wikipedia Commons*. Haettu 05. 05 2020 osoitteesta https://commons.wikimedia.org/wiki/File:Android_Studio_icon.svg