

Henri Haavisto

**ANDROID-SOVELLUS AUTON POLTTOAINETALouden TARK-
KAILUUN**

ANDROID-SOVELLUS AUTON POLTTOAINETALouden TARK- KAILUUN

Henri Haavisto
Opinnäytetyö
Kevät 2017
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

Tekijä(t): Henri Haavisto
Opinnäytetyön nimi: Android-sovellus auton polttoainetalouden tarkkailuun
Työn ohjaaja(t): Kari Laitinen
Työn valmistumislukukausi ja -vuosi: Kevät 2017
Sivumäärä: 41 + 1 liite

Tässä opinnäytetyössä suunniteltiin ja toteutettiin Android-käyttöjärjestelmälle auton polttoaineen kulutusta mittaava sovellus. Tilaajana toimi Oulun ammattikorkeakoulu. Sovellus näyttää käyttäjän sijainnin kartalla ja tämän liikkumisnopeuden, jonka mukaan sovellus laskee arvioidun polttoaineen kulutuksen.

Android-sovelluksen toiminnollisuus kehitettiin Java-kielellä ja ulkoasu toteutettiin XML-kielellä. Opinnäytetyössä kuvaillaan, kuinka sovellus on toteutettu ja minkälainen sovellus on tuotettu.

Sovellus on laajennettavissa edelleen, mutta sitä ei enää tällä hetkellä kehitetä eteenpäin.

Asiasanat: mobiiliohjelmointi, mobiilisovellus, Android

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software development

Author(t): Henri Haavisto

Title of thesis: Android application for monitoring fuel efficiency of car

Supervisor(s): Kari Laitinen

Term and year when the thesis was submitted: Spring 2017

Pages: 41 + 1 appendix

In this thesis the topic was to plan and develop an Android application which calculates the consumption of gasoline when driving a car. Oulu University of Applied Sciences acts as a subscriber. The application shows user's location on map and the user's movement speed, and calculates the estimated consumption of gasoline.

The application was written in Java and its layout in XML. The thesis describes how the application is designed and implemented.

It is possible to further develop this application with multiple ways, but right now the app is not in development anymore.

Keywords: mobileprogramming, mobilessoftware, Android

ALKULAUSE

Haluan kiittää kaikkia niitä henkilöitä, jotka auttoivat minua saattamaan päätökseen tämän opinnäytetyön. Erityiskiitokset perheelleni, jotka tukivat minua silloinkin, kun minusta tuntui, että usko loppuu tämän tehtävän tekemiseen. Opinnäytetyö tuntui aina viivästyvän, vaikka loppua kohden työtä on tehty vain kiihtyvällä tahdilla.

Muhoksella 4.4.2017

Henri Haavisto

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	7
1 JOHDANTO	10
2 MOBIILILAITTEET	11
2.1 Matkapuhelin	11
2.2 Älypuhelin	11
2.3 Tabletti	12
3 KÄYTTÖJÄRJESTELMÄ	13
3.1 Android	13
3.1.1 Androidin kehitys	14
3.1.2 Android Studio	14
3.2 Microsoft Windows	15
3.3 Linux	15
4 OPINNÄYTETYÖN TOTEUTUS	17
4.1 Opinnäytetyön eteneminen	17
4.2 Sovelluksen toteutus	18
4.3 Polttoainetalouden laskenta	22
4.4 Ohjelman viimeistely ja testaus	24
5 PROTOTYYPPISOVELLUKSEN OMINAISUUDET	25
6 PROTOTYYPIN OHJELMISTORAKENNE	29
7 JATKOKEHITYSMAHDOLLISUUDET	36
8 YHTEENVETO	38
LÄHTEET	39
LIITE	
Liite 1 Ohjelmakoodi	

SANASTO

Aktiviteetti	Yksittäinen, yleensä vuorovaikutteinen toiminto ohjelmassa käyttäjän kanssa
Android	Käyttöjärjestelmä, yleinen mobiililaitteissa
Android Studio	Android-sovelluksien ohjelmointiympäristö
Android Wear	Android-pohjainen älykello
Apple	Yhdysvaltalainen suuryritys
Debian	Linux-pohjainen käyttöjärjestelmä
Eclipse	Ohjelmointiympäristö
Gentoo	Linux-pohjainen käyttöjärjestelmä
Google	Yhdysvaltalainen yhtiö, myös hakukone
Google Play	Googlen sovelluskauppa
GPS	Global Positioning System, maailmanlaajuinen paikallistamisjärjestelmä
HTC	Taiwanilainen suuryritys
Huawei	Kiinalainen suuryritys
Intent	Android-ohjelmoinnissa käytettävä abstrakti kuvaus suoritettavasta operaatiosta
iOS	Applen kehittämä käyttöjärjestelmä mobiililaitteille
IoT	Internet of Things, esineiden internet
iPad	Applen kehittämä tabletti
iPhone	Applen kehittämä älypuhelin

Java	Ohjelmointikieli, Android-laitteiden sovelluskehitys ta- pahtuu usein Java-kielellä
Käyttäjä	Henkilö, joka käyttää laitetta, konetta tai ohjelmaa
Käyttöjärjestelmä	Keskeinen tietokoneen ohjelmisto, joka hallinnoi ko- neen resursseja, mahdollistaa muiden ohjelmien toi- minnan
Linux	Unix-klooni, Linus Torvaldsin kehittämä käyttöjärjestel- mäydin, polveutuu epäsuorasti Unixista
Microsoft	Yhdysvaltalainen suuryritys
Mobiili	Liikkuva, paikasta toiseen siirrettävä
Mobiiliapplikaatio	Sovellus, joka on suunniteltu mobiilipäätelaitteisiin
Mobiilipäätelaite	Matkapuhelin, älypuhelin, tabletti
OS X	Applen kehittämä käyttöjärjestelmä pöytäkoneelle
Ohjelmointiympäristö	Ohjelma, jolla kehitetään ohjelmia, esimerkiksi Android Studio ja Eclipse
Palvelin	Tietoliikenteessä käytettävä palvelinohjelmisto
Red Hat	Linux-pohjainen käyttöjärjestelmä
Samsung	Korealainen suuryritys
Samsung Kies	Samsungin kehittämä ohjelma tiedonsiirtoa varten
String	Ohjelmoinnissa käytettävä merkkijonomuuttuja
Sony	Japanilainen suuryritys
Tabletti	Kämmentietokone, taulutietokone, mobiililaite
Tux	Linuxin maskotti, ei kuitenkaan virallinen logo

Ubuntu	Linux-pohjainen käyttöjärjestelmä
Unix	Laitteistoriippumaton käyttöjärjestelmä
Windows	Microsoftin kehittämä käyttöjärjestelmä
Windows Phone	Microsoftin kehittämä älypuhelin
Xbox One	Microsoftin kehittämä pelikonsoli
XML	Extensible Markup Language, rakenteellinen kuvauskieli
.apk	Tiedostomuoto, jonka Android-ohjelmointiympäristö luo mobiililaitteella ajettavaksi ohjelmaksi

1 JOHDANTO

Työni aiheena on Android-mobiilisovellus jolla voi seurata auton polttoaineta-
loutta. Aiheen sain koulutusohjelmavastaavalta Riitta Ronnulta.

Aiheen valinnan tein nopeasti, koska mobiilisovelluksen tekeminen on miele-
kästä ja aihe oli kiinnostava. Lisäksi valmistuva sovellus voisi olla hyödyllinen
minulle itsellenikin. En ole toistaiseksi kuullut vastaavasta ohjelmasta, mutta
iOS:lle kehitetään ohjelmia noin 1000 kpl päivässä, joten sille ilmestyy mobiiliso-
velluksia toiseksi eniten maailmassa Android -alustan olevan ensimmäisenä (1;
2; 3). Hyvin todennäköisesti vastaava ohjelma on jo kehitetty tai on kehitteillä,
mutta insinöörin tehtävä on kuitenkin luoda sellaista jota ei vielä ole olemassa.

Mahdollisesti vastaavanlainen ohjelma on julkaistu jo viimeistään siinä vai-
heessa, kun tämä ohjelma on saatu päätökseen. Olettamus perustuu pelkäs-
tään siihen nopeuteen, millä vauhdilla Google Playn ohjelmien määrä kasvaa
(2; 3). Koska Androidille mobiilisovelluksen julkaiseminen on huomattavasti hal-
vempaa kuin iOS-laitteille, niin hyvin todennäköisesti ennemmin tai myöhemmin
vastaavanlainen ohjelma kehitetään.

Tällä hetkellä eletään aikaa, jolloin kuulee puhuttavan esineiden internetistä,
englanniksi IoT eli Internet of Things. Tässä ajatuksena on, että laitteet ja esi-
neet, joissa on jonkin verran ”älyä”, liitetään myös verkkoon. Uskon, että tähän
alueeseen kuuluu myös autot. IoT on tällä hetkellä vielä melko vähäisesti hyö-
dynnetty tekniikan alue, mutta yleinen uskomus on, että se tekee lähivuosina lo-
pullisen läpimurron (4).

Autoissa on tällä hetkellä ajotietokoneessaan polttoaineen kulutus, mutta monia
 muita ominaisuuksia kuluttajat varmasti kaipaavat. Tämä sovellus on tehty sel-
laisena aikana, jolloin IoT ei ole vielä lyönyt läpi, eikä autoihin ole vielä integ-
roitu reaaliajassa toimivaa karttaa. Näin ollen tämä sovellus on tällä hetkellä tar-
peellinen, mutta saattaa jäädä ajan saatossa tarpeettomaksi.

2 MOBIILILAITTEET

Mobiililaitte-sanalla tarkoitetaan tässä tekstissä tästä eteenpäin kaikissa asiayhteyksissä kannettavaa tai liikuteltavaa älylaitetta, jossa on käyttöjärjestelmä ja joka pystyy suorittamaan ohjelmia. Tällaisia laitteita ovat matkapuhelimet, älypuhelimet, tabletit ja kannettavat tietokoneet. Eri yritysten kehittämiä laitteita kutsutaan puhekielessä niiden nimillä, kuten iPad, mutta teknisesti määritellen iPad, iPhone ja muut kannettavat tai siirrettävät älylaitteet kuuluvat mobiililaitte-kategoriaan.

2.1 Matkapuhelin

Matkapuhelin on matkaviestintään tarkoitettu telepäätelaitte, jota voidaan käyttää liikkeellä oltaessa tai ennalta määräämättömissä paikoissa matkaviestinverkon toiminta-alueella ja jolla voidaan puhua puheluita (5). Matkapuhelin on vanhempi kuin älypuhelin tai tabletti, joten siinä on paljon vähemmän toimintoja ja ominaisuuksia kuin kahdessa muussa laitteessa. Kaikissa matkapuhelimissa ei ole mahdollisuutta yhdistyä internettiin, joten perinteinen matkapuhelin on jäämässä väistämättä pois käytöstä.

2.2 Älypuhelin

Älypuhelin on matkapuhelimesta kehittyneempi versio. Älypuhelimelle on tyypillistä internetyhteys, monipuolinen mukautettava sovellusvalikoima sekä sellainen käyttöjärjestelmä, johon kolmansien osapuolten on mahdollista tuottaa uusia sovelluksia. Älypuhelimessa on usein myös graafinen käyttöliittymä ja kosketusnäyttö (6). Älypuhelimessa käyttöjärjestelmä on usein sen valmistajan oma käyttöjärjestelmä, esimerkiksi Applen laitteissa on iOS ja Microsoftilla Windows. On myös muita yrityksiä, jotka valmistavat älypuhelimia, mutta käyttävät Android -käyttöjärjestelmää. Tällaisia älypuhelin valmistajia ovat esimerkiksi Samsung, Sony, Motorola, Huawei ja HTC.

2.3 Tabletti

Tabletti on mobiilikäyttöön tarkoitettu pienikokoinen ja litteä kannettava tietokone, jossa on sormin käytettävä kosketusnäyttö (7). Tabletissa yhdistyvät älypuhelimien ja tietokoneen ominaisuudet, mutta siitä on jätetty myös ominaisuuksia pois, kuten hiiriohjauksen ja tekstiviestit. Tabletissa on myös samat käyttöjärjestelmät saatavilla kuin älypuhelimissa, myös samalla logiikalla eli valmistajasta riippuen. Tabletin yleisin koko vaihtelee seitsemästä tuumasta 10,1 tuumaan, joten se tarjoaa reilusti suuremman näytön kuin älypuhelimissa. On myös olemassa hybridilaitteita, joissa tabletti yhdistetään telakkaan, joka yhdistää näppäimistön ja hiiren tablettiin, jolloin kokonaisuudeksi saadaan pöytäkoneen veroinen laite.

3 KÄYTTÖJÄRJESTELMÄ

Käyttöjärjestelmä on keskeinen tietokoneen ohjelmisto, joka mahdollistaa muiden ohjelmien toiminnan. Se hallinnoi tietokoneen resursseja ja luo laitteiston yksityiskohdista riippumattoman operointialustan ja järjestelmäkutsut (8).

Käyttöjärjestelmiä on useita erilaisia pöytäkoneille ja kannettaville laitteille. Pöytäkoneille yleisiä käyttöjärjestelmiä ovat Windows, Linux ja OS X. Kannettaville laitteille yleisiä käyttöjärjestelmiä ovat Android, iOS ja Windows Phone (8).

3.1 Android

Android on tällä hetkellä älypuhelinien yleisin käyttöjärjestelmä. Sen kehityksen voidaan katsoa todella alkaneen elokuussa 2005, jolloin Google osti Android Inc -nimisen yrityksen, joka oli perustettu vuonna 2003 kehittämään entistä älykkäämpiä mobiililaitteita (1, s. 11).

Ensimmäinen Androidia käyttävä laite oli T-Mobile G1 -puhelin (kuva 1), joka julkaistiin lokakuussa 2008. Seuraavan vuoden aikana Android-puhelimia ilmestyi yli 10 kpl lisää. Vuonna 2010 Android puhelimia ilmestyi yli 44 mallia useilta eri valmistajilta ja seuraavina vuosina yhä kiihtyvällä tahdilla. Android saavutti markkinajohtajuuden nopeasti julkaisemisensa jälkeen. (1, s. 13.)



KUVA 1. T-Mobile G1 (9)

3.1.1 Androidin kehitys

Android kehittyy jatkuvasti ja kirjoitushetkellä viimeisin vakaa versio on 6.0, joka on julkaistu 17. elokuuta 2015. Tämän version nimi on Marshmallow (10). Kaikki versiot ovat nimetty jonkin makean mukaan (kuva 2).

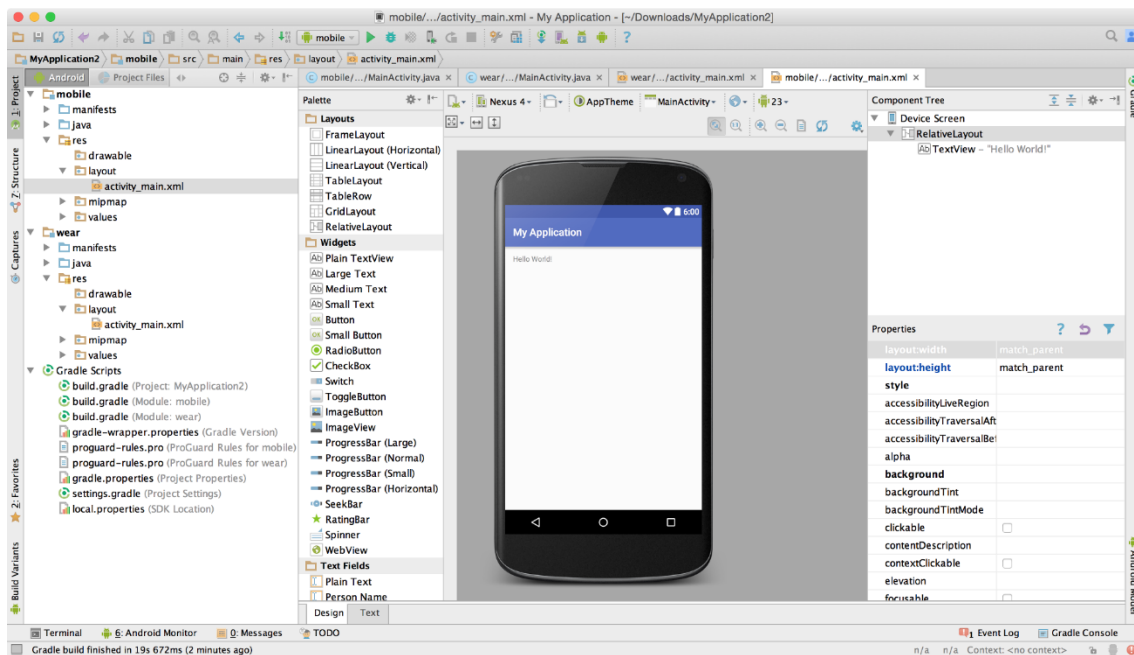


KUVA 2. Android –versiot (11)

Nykyään Androidia käyttäviä laitteita on jo jokapäiväisessä elämässä: puhelimissa, tableteissa, televisioissa, aktiivisuusrannekkeissa, jopa autoissa (11).

3.1.2 Android Studio

Android Studio on Googlen lanseeraama kehitysympäristö Android-laitteiden sovelluskehitykseen (12). Ensimmäinen versio ilmestyi toukokuussa 2013 versionumerolla 0.1, mutta vakaa versio ilmestyi vasta joulukuussa 2014, versionumerolla 1.0. Android Studiolla ohjelmoidaan Java- ja XML-kielillä, mutta myös C++ on mahdollinen kieli. Android sovelluksia voidaan ohjelmoida myös muilla kielillä, mutta silloin kaikkia Androidin ominaisuuksia ei saada käyttöön (12). Android Studiolla voidaan kirjoittaa ohjelmia puhelimiin, tabletteihin, Android Weariin, televisioihin ja autoihin joissa on Android käyttöjärjestelmänä.



KUVA 3. Esimerkinäkymä Android Studiosta

Aikaisemmin Androidin sovelluksia ohjelmoitiin pääasiassa Eclipse-kehitysympäristöllä, mutta Android Studio on syrjäyttänyt sen pääasiallisena kehitysympäristönä (12).

3.2 Microsoft Windows

Mahdollisesti yleisin käyttöjärjestelmä on Microsoftin Windows, jota käytetään ainakin kouluissa ja useissa yrityksissä. Useimmiten se on valmiiksi asennettuna tietokoneissa. Sen markkinaosuus oli 88,12 % henkilökohtaisten tietokoneiden käyttöjärjestelmistä vuonna 2014 (13). Tällä hetkellä uusien Windows-käyttöjärjestelmien on nimeltään Windows 10, joka on käytössä Windows-puhelimissa, tableteissa, Xbox One:ssa, kannettavissa tietokoneissa ja pöytäkooneissa.

3.3 Linux

Linux on alkujaan polveutunut Unixista, joskin melko epäsuorasti. Se on pikemminkin Unix-kloonin, koska Linux on Unix-pohjainen käyttöjärjestelmä. Erilaisia Unix-pohjaisia käyttöjärjestelmiä alkoi levitä 1970- ja 1980-luvulla (14). Myös

Android käyttää Linux-ydintä. Linux on ehkä maailman käytetyin palvelinkäyttöjärjestelmä, sillä se on hyvin vakaa. Linux on tunnettu esimerkki yhteistyöstä vapaan ja avoimen lähdekoodin ohjelmistojen kehityksessä. Yleensä kuka tahansa saa käyttää Linuxia vapaasti, levittää tai muokata sitä. Tunnettuja Linux pohjaisia käyttöjärjestelmiä ovat Ubuntu, Debian, Gentoo ja Red Hat. Ubuntu on ladattavissa ilmaiseksi osoitteesta <http://www.ubuntu-fi.org/>.



KUVA 4. Linux -maskotti "Tux"

4 OPINNÄYTETYÖN TOTEUTUS

4.1 Opinnäytetyön eteneminen

Opinnäytetyö toteutettiin siten, että tilaajana toimii tässä tapauksessa OAMK, koska aihe saatiin koulun opinto-ohjaajalta, joka kyseli mahdollisista opinnäytetyön aiheista koulun IT-tukihenkilöltä. Opinnäytetyössä ei siis ole yritys toimeksiantajana. Näin ollen ideaa lukuun ottamatta kaikki muu opinnäytetyössä oleva on allekirjoittaneen ideoimaa.

Opinnäytetyö toteutettiin oman aikataulun mukaan, mutta kuitenkin siten, että edistymistä osoitettiin viikon tai parin viikon välein ohjaavalle opettajalleni Kari Laitiselle. Tavoitteena oli saada opinnäytetyö tehtyä joulukuuhun 2016 mennessä. Pienempiä välietappeja ei katsottu tarpeelliseksi asettaa, koska tilaajana toimii koulu eikä yritys, joten aikataulujen ei tarvitse olla niin pitäviä kuin normaalisti.

Opinnäytetyö viivästyi aikataulustaan, sillä vielä tammikuun 2017 lopulla ohjelmassa ilmeni sellaisia virheitä, jotka olivat välttämättömiä korjata. Alkuperäinen tavoite saada ohjelma valmiiksi joulukuussa oli melko optimistinen, sillä kuten ohjaava opettajani totesi, tätä ohjelmaa voi halutessaan laajentaa melkein loputtomasti. Näin ollen ohjelmasta karsittiin sellaiset ominaisuudet, jotka eivät olleet välttämättömiä, mutta ideat niistä esitellään jäljempänä tässä opinnäytetyön selostuksessa.

Tätä opinnäytetyötä työstettiin epäsäännöllisin väliajoin, mikä mahdollisesti edesauttoi sitä, että aikataulussa ei pysytty. Siihen kuitenkin pyrittiin, että ainakin kaksi tai kolme kertaa viikossa tätä opinnäytetyötä työstettiin eteenpäin. Ohjelma oli pääosin valmis tammikuun lopussa, mutta kirjallisen osuuden määrä oli työläämpi kuin osasin odottaa.

Opinnäytetyö aloitettiin syyskuun puolessa välissä, jolloin pidettiin tapaaminen ohjaavan opinnäytetyön opettajan Kari Laitisen kanssa ja keskusteltiin aihealueesta. Vielä saman päivän aikana saatiin yhteisymmärrys siitä, minkälainen tuotos on odotettavissa. Vaikka aivan kaikki ei ollut vielä loppuun asti mietitty, niin

pääpiirteet olivat kuitenkin selvät. Aiheena tulisi olemaan mobiililaitteelle, tässä tapauksessa Android-laitteelle oleva ohjelma, joka mittaa polttoaineen kulutuksen.

4.2 Sovelluksen toteutus

Aloitin aivan alusta sovelluksen toteuttamisen. Koska olin vaihtanut hiljattain tietokoneeni uudempaan, niin kaikki ohjelmisto piti asentaa koneelle. Ensimmäiseksi asensin Android Studion osoitteesta <https://developer.android.com/studio/install.html>, jossa latsin tiedoston ja seurasin annettuja ohjeita. Seuraavaksi asensin koneelle Java 7:n osoitteesta <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Erikoista oli se, että viimeisin versio eli Java 8 ei tuntunut toimivan. Java 7 tarvitaan Android Software Development Kitin asentamiseksi, joka on välttämätön ohjelman kirjoittamiseen ja ajamiseen. Lisäksi asensin Android NDK:n, koska sen avulla ohjelmaa voidaan kirjoittaa myös C- ja C++-kielellä. Toistaiseksi ohjelmassa ei ole kuitenkaan käytetty muuta kuin Javaa ja XML:ää.

Android Studion mukana tulee emulaattori (kuva 5), jolla voidaan ajaa ohjelmaa, mikäli ohjelmoija ei halua käyttää puhelinta ohjelman ajamiseen.



KUVA 5. Android Studion Emulator

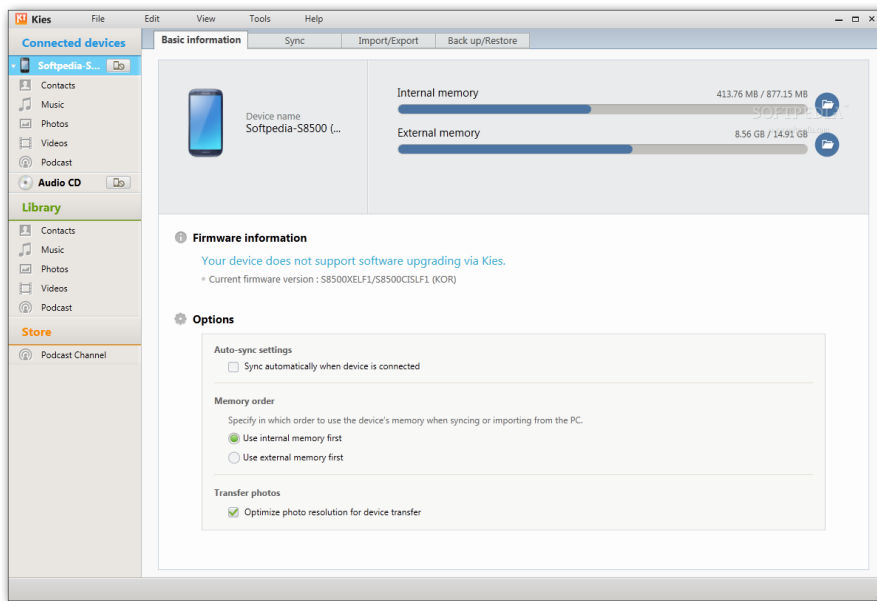
Tämä on kuitenkin hyvin aikaa vievää, sillä emulaattori on niin raskas, että sen käynnistymiseen menee useita minuutteja aikaa. Normaalisti parin minuutin odottelu ei ole ylipääsemätön asia, mutta ohjelmoitaessa se alkaa pian muodostua raskitukseksi. Toki emulaattorin voi käynnistää itsenäisesti, ennen kuin käynnistää kehitysympäristön, mutta emulaattorilla työskentely on kuitenkin huomattavasti hitaampaa kuin aidolla puhelimella. Emulaattoria käytettäessä täytyy ensin luoda kloonipuhelin, jossa ohjelmoija päättää mistä puhelinmallista kloni tehdään. Ohjelmoija voi halutessaan myös muuttaa kloonipuhelimen muistin kokoa, näytön kokoa tai muuta sellaista. Android Studio luo käytännössä puhelimen käyttöjärjestelmineen ja ohjelmistoineen päivineen tietokoneen kovalevylle. Tiedoston koko on helposti gigatavun luokkaa.

On myös olemassa vaihtoehtoisia emulaattoreita, jos haluaa nopeuttaa ohjelman ajamista. Eri projektissa olen käyttänyt Genymotion -nimistä emulaattoria (kuva 6). Se on varovaisen arvion mukaan 3–4 kertaa Android Studion emulaattoria nopeampi. Asentaminen ei ole erityisen hankalaa ja ohjeita kyllä löytyy, kuinka Genymotionin saa liitettyä kehitysympäristöön. Huonona puolena Genymotionissa on se, että se ei aina skaalaa oikeassa mittasuhteessa ajettavan ohjelman ulkoasua. Kun ohjelman tekee valmiiksi eli rakentaa siitä .apk-tiedoston, vasta ajettavalla laitteella saattaa paljastua, että näytöllä näkyvät kuvakkeet eivät ole siellä, minne ohjelmoija on ne suunnitellut.



KUVA 6. Genymotion emulaattorin aloituskuva

Kaikista paras vaihtoehto Android-ohjelman ajamiseen on Android-alustainen laite, yleensä älypuhelin. Asensin Samsung Kies 3:n koneelleni, koska tiedonsiirto on jouhevaa kyseisen ohjelman kanssa ja siitä saadaan tarvittavat laiteohjelmat USB-tiedonsiirtoon (kuva 7). Puhelimessa pitää avata sovelluskehittäjän asetukset painamalla seitsemän kertaa koontiversion kohtaa (kuva 8). Siinä täytyy sallia asennus muualta kuin Play-kaupasta ja USB-virheenkorjaus päälle. Sen jälkeen puhelimella pystytään ajamaan ohjelmaa. Käytettävä puhelin tässä projektissa on Samsung S5 Neo (kuva 9).



KUVA 7. Esimerkkikuva Samsung Kiesistä



KUVA 8. Sovelluskehittäjän asetukset päälle



KUVA 9. Samsung S5 Neo

Alku lähti hieman tahmeasti liikkeelle, kun edellä kuvatut toimet kestivät hyvän aikaa, ennen kuin kaiken sain tehtyä ja asennettua onnistuneesti. Tässä täytyi etsiä netistä ohjeita, kun kaikki ei sujunut heti ensi yrittämällä. Eritoten Javan asennuksessa ja Android SDK:n kanssa oli hieman ongelmia, kun ensin mainitun uusin versio ei toiminut ja sen myötä jäljempänä mainittu ei siten toiminut Android Studiossa. Kun lopuksi kun löytyi oikeat versiot, SDK:n kansiopolku piti vielä manuaalisesti asettaa Android Studiossa kohdalleen.

Tein omaan ohjelmakoodiin vielä sellaisen varmistuksen, että avasin kehitysympäristössä AndroidManifest-tiedoston ja kirjoitin sinne seuraavan koodin:

```
android:debuggable="true"
```

Tällä tavalla ohjelman ajaminen on mahdollista puhelimella. Ajoin valmiin esimerkkiohjelman varmistuakseni asiasta (kuva 10).



KUVA 10. Esimerkkiohjelma "Hello world!"

4.3 Polttoainetalouden laskenta

Seuraavaksi ryhdyin etsimään tietoa polttoaineen kulutuksesta autolla ajettaessa. Mistään en löytänyt kaavaa, jolla laskettaisiin polttoaineen kulutus. Ainoastaan seuraavat faktat olivat selvät:

- Auto kuluttaa paljon käynnistyessään.
- Auto kuluttaa enemmän kaupunkiajossa kuin maantieajossa.
- Auto kuluttaa kiihdyttäessä enemmän.
- Auto kuluttaa polttoainetta myös tyhjäkäynnillä.

Hain tietoa useista eri lähteistä, mutta ainoaksi käytännössä hyödylliseksi tiedoksi jäi laskelma, jonka mukaan auto kuluttaa polttoainetta lähes litran jokaista 18:aa kilometriä kohti (15). Laskelmassa keskikokoinen henkilöauto kuluttaa kuusi litraa polttoainetta 100 kilometrillä. Muutamalla laskutoimituksella tästä

saadaan tulokseksi 0,0017 litraa minuuttia kohti eli tunnissa 6,12 litraa. Tällainen kulutus autossa on keskitasoa, eli auto ei ole pieniruokainen eikä kuluta suuresti. Koska muuta kaavaa ei ollut saatavilla, ohjelmaa ei voida luonnehtia tarkaksi, ennemmin suuntaa-antavaksi. Käynnistyksen aikana auto kuluttaa polttoainetta hieman vajaa puoli litraa (16), joten ohjelma lisää kulutukseen puoli litraa jokaisella käynnistyskerralla. Tunnin ajolla saadaan kulutukseksi $6,12 + 0,5 = 6,62$ litraa.

Koska kiihdytyksestä ei ollut myöskään saatavilla kaavaa, jolla lasketaan kulutuksen nousu, niin kiihdytyksestä aiheutuva kulutuksen nousu täytyisi ohjelmoin itse laskea. Tämä on kuitenkin sellainen ominaisuus, jota en katsonut ensisijaisesti tarpeelliseksi, se jää suosiolla odottamaan jatkokehitystä. Toki kiihdytys vaikuttaa suoraan kulutukseen, mutta ohjelman kokonaisuuden ja ajankäytön kannalta tämä täytyi jättää myöhemmäksi. Auto kuluttaa enemmän polttoainetta kovassa vauhdissa kuin hitaalla ajaessa, mutta kulutus näkyy selkeästi vasta sitten, kun ajetaan nopeampaa kuin 100 km/h. Koska Suomen maanteillä on kesäisin pääasiassa 100 km/h nopeusrajoitus, ohjelmaan ei ole katsottu tarpeelliseksi laskea lisäystä kulutukseen, jos vauhti ylittää kyseisen tuntinopeuden.

Ohjelmakoodin kannalta on mahdotonta tehdä hyvin tarkkaa kulutuslaskelmaa, koska polttoaineen kulutukseen vaikuttavat monet seikat. Näitä ovat kylmäkäynnistys, renkaiden kunto ja ilmanpaine, käytössä oleva vaihde, aerodynamiikka, auton moottorin koko, auton ikä ja malli, ilmastointi jos on ajon aikana käynnissä, Webasto, jarruttelu ja kiihdytys sekä ajonopeus (16; 17). Lisäksi myös osien kuluneisuus vaikuttaa polttoaineen kulutukseen (18).

Tämän vuoksi ohjelma laskee vain aikalaskurilla polttoaineen kulutuksen eikä huomioi suurinta osaa edellä mainituista seikoista, koska niitä on mahdoton toteuttaa puhelinsovelluksella. Ohjelma laskee polttoaineen kulutuksen sekunnissa 6,12 litraa tunnissa jaettuna 3600. Tästä saadaan tulokseksi 0,0017, joka on litramäärä, joka kuluu sekunnin ajon aikana. Lopputulokseen lisätään puoli litraa, koska auto on käynnistetty ainakin kerran.

Laskuri käynnistyy vasta sitten, kun ohjelman käyttäjä päättää käynnistää soveluksen aktiviteetin, jolla on oma fragmentti (19). Fragmentin voidaan kuvailla

olevan aktiviteetin itsenäinen osa-alue, jolla on oma elinkaarensa ja joka voi ottaa vastaan käyttäjältä syötteitä. Tämä on harkittu ratkaisu, koska ohjelmaa oletetaan käytettävän siten, että käyttäjä käynnistää ohjelman sen jälkeen, kun on käynnistänyt ajoneuvonsa. Sitten kun käyttäjä on valmis ajamaan ajoneuvoaan, hänen oletetaan käynnistävän aktiviteetin, jolloin käynnistyy myös laskuri.

Kun käyttäjä lopuksi poistuu aktiviteetista, ohjelma lähettää viestin ruudulle arvioidusta polttoaineen kulutuksesta ja sen hinnasta. Laskuri nollautuu joka kerta, kun käyttäjä poistuu aktiviteetista painamalla joko paluu-painiketta tai laitteensa menu-painiketta.

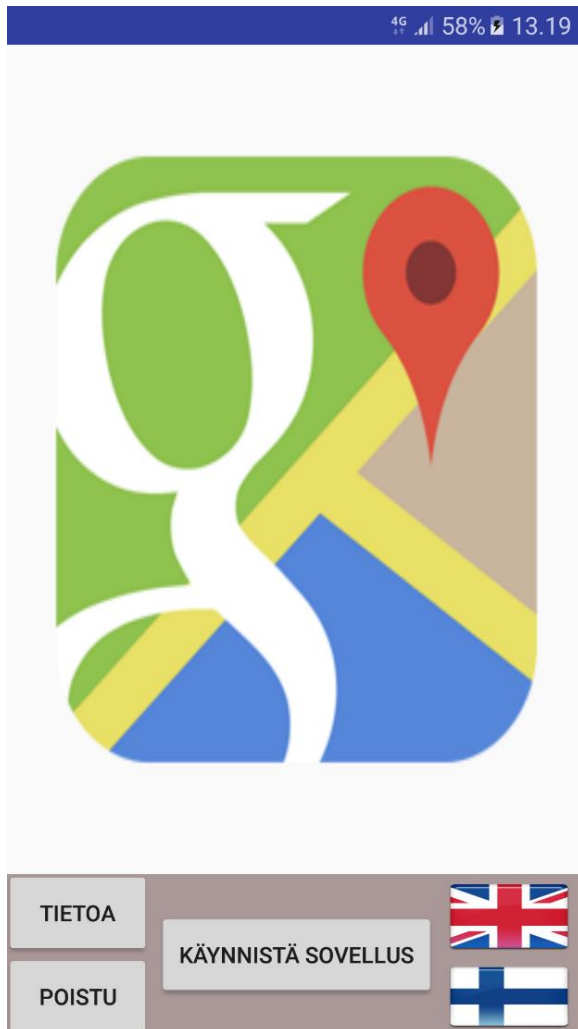
4.4 Ohjelman viimeistely ja testaus

Ohjelma viimeisteltiin siihen kuntoon, jossa se näytettiin ohjaavalle opettajalle. Tässä opinnäytetyössä olevat kuvat ovat samasta ohjelmaversiosta. Ohjelma on tällä hetkellä toimiva versio, mutta sitä voidaan edelleen kehittää luvussa Jatkokehitysmahdollisuudet mainitulla tavalla. Tällä hetkellä ohjelma ei ole jatkokehityksessä. Ohjelman toimintaidea rakentuu sijainnin paikantamiseen ja sijainnin nopeuden muutokseen sekä niiden välisiin laskutoimituksiin. Muut ominaisuudet laajentavat ohjelmaa. Siksi erillistä ja kattavampaa testausosiota ei ole tässä opinnäytetyössä, koska käytännössä ohjelmaa testattiin heti sen jälkeen, kun siihen kirjoitettiin uusi ominaisuus. Keskeisimmän ominaisuuden täytyi toimia alusta alkaen. Tarkemmin testauksesta kerrotaan luvussa Yhteenveto.

5 PROTOTYYPPISOVELLUKSEN OMINAISUUDET

Jotta sovellusta voidaan käyttää, käyttäjän on annettava sovellukselle lupa käyttää verkkoyhteyttä ja GPS-sijaintia. Nämä ovat ohjelman keskeisimmät toiminnot, joita ilman ohjelma on hyödytön käyttäjälle.

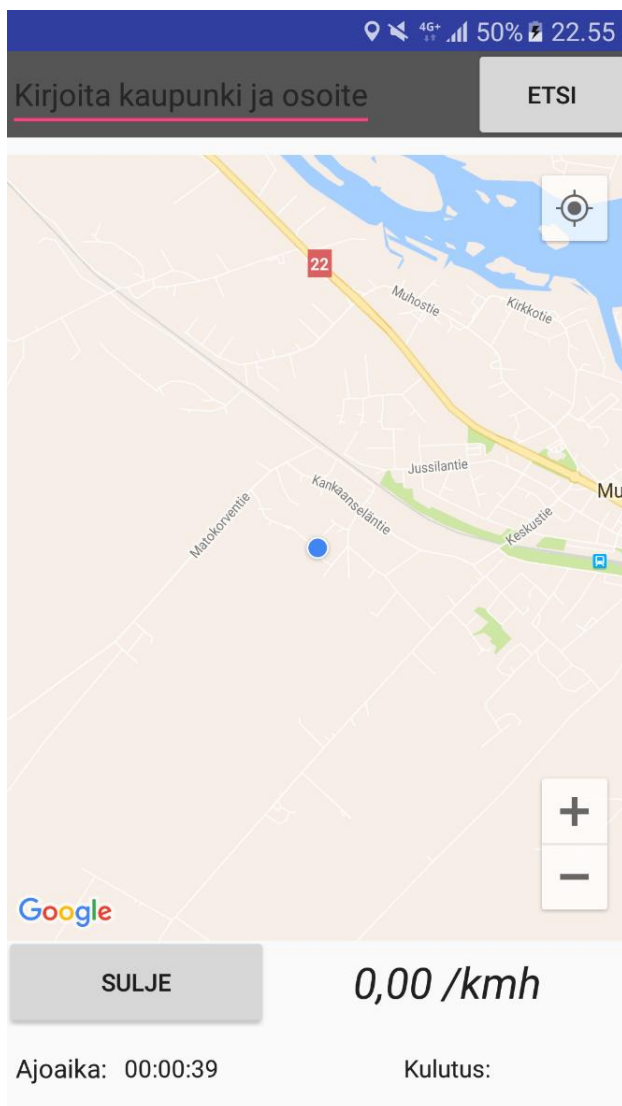
Sovellus tukee tällä hetkellä kahta kieltä, suomea ja englantia. Aiemmassa versiossa kieli oli valittavissa päävalikosta painikkeella (kuva 11), mutta tällaisen käytännön ylläpitäminen on turhan työlästä. Siksi oletuskieli on englanti ja jos käytettävässä laitteessa kieleksi on asetettu suomi, niin myös sovellus on suomenkielinen.



KUVA 11. Alkuperäinen näkymä sovelluksen pääsivulta

Sovellus käyttää Googlen kehittämää Google Maps -karttapalvelua, joka näyttää melko tarkasti koko maailman kartan (20).

Ohjelma hakee käyttäjän maantieteellisen sijainnin käyttämällä GPS-koordinaatteja ja liikuttaa karttakuvan käyttäjän sijainnin kohdalle. Käyttäjä voi halutessaan lähentää ja loitontaa karttaa. Painikkeet näkyvät ohjelmassa painikkeina + ja -. Sovellus päivittyy sekunnin väliajoin, ja piirtää käyttäjän sijainnin suhteessa maantieteelliseen sijaintiin (kuva 12).



KUVA 12. Ohjelma näyttää käyttäjän maantieteellisen sijainnin

Sovellus laskee käyttäjän etenemisvauhdin käyttämällä Google Location Services API -rajapintaa, joka käyttää vastaavanlaista metodia kuin Location-luokan `getSpeed()`-metodi. Android Developers sivuston mukaan Google Location Services API on tehokkaampi kuin framework-rajapinta, jonka lisäksi ensin mainitussa on muun muassa virranhallinta tehokkaampaa (21). Jotta Google Location Services API -rajapintaa voidaan käyttää, niin on luotava `GoogleApiClient`-instanssi (kuva 13) ja siinä on oltava mukana pakolliset metodit (kuva 14).

```
// Create an instance of GoogleApiClient.
if (mGoogleApiClient == null) {
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
}
```

KUVA 13. Esimerkkikoodi `GoogleApiClient`in instanssin luomisesta

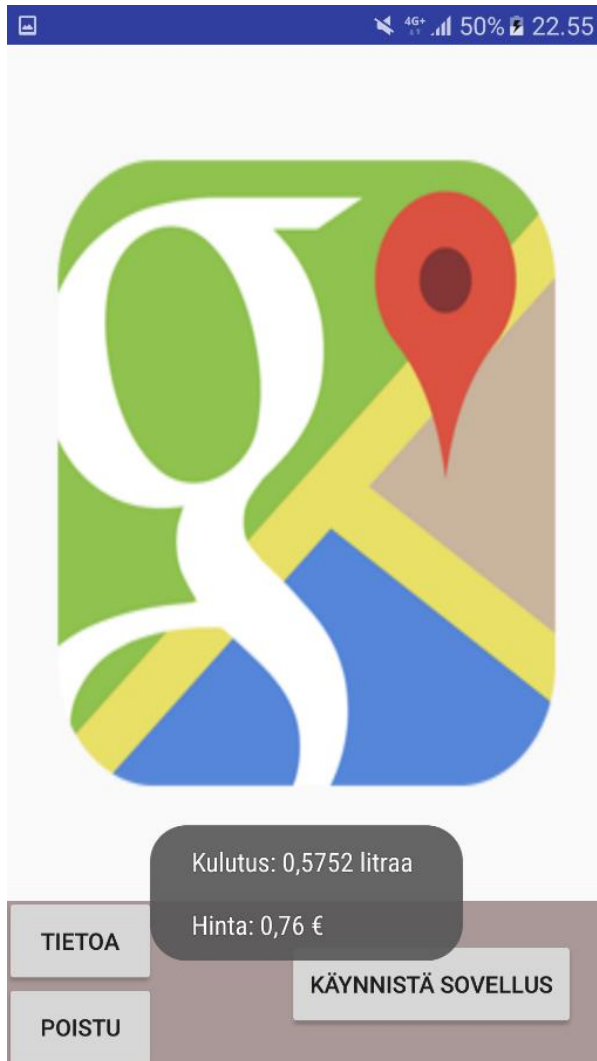
```
protected void onStart() {
    mGoogleApiClient.connect();
    super.onStart();
}

protected void onStop() {
    mGoogleApiClient.disconnect();
    super.onStop();
}
```

KUVA 14. `GoogleApiClient`in pakolliset metodit

Karttanäkymästä poistuttaessa ohjelma lähettää laitteen näytölle lyhyen teksti-ilmoituksen, jota kutsutaan nimellä `Toast`. Ilmoitus tulee näkyviin heti kun käyttäjä sulkee karttanäkymän. Ilmoituksen tarkoitus on informoida käyttäjää suunnan-antavasta polttoaineen kulutuksesta ja siitä, kuinka paljon kulkeminen on maksanut. Ohjelmassa taustalla pyörii aikalaskuri, jonka perusteella lasketaan auton polttoaineen arvioitu kulutus. Sovellus laskee aikaa maksimissaan kaksi tuntia, jolloin laskuri tulee täyteen. Tällöin laskuri pysähtyy ja ohjelma lähettää

näytölle ilmoituksen kuluneesta polttoaineen määrästä ja arvioidusta hinnasta (kuva 15).



KUVA 15. Arvioidun kulutuksen näyttäminen

6 PROTOTYYPIN OHJELMISTORAKENNE

Android -laitteiden sovelluskehityksessä kaikki sovelluksen tarvitsemat luvat, kuten nettiyhteyden muodostaminen, luetellaan AndroidManifest-tiedostossa (kuva 16). AndroidManifest-tiedostoa voidaan kutsua myös julistukseksi tai manifestiksi.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.henry.myapplication">
    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the 'MyLocation' functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

KUVA 16. AndroidManifest -tiedostossa luetellut luvat ohjelman käyttöön

Android-laite tarvitsee tämän tiedoston lukeakseen siitä keskeisimmän informaation, jonka käyttöjärjestelmä tarvitsee, jotta ohjelmaa voi ajaa (22). Tässä julistuksessa luetellaan myös ohjelman kaikki luokat. Ohjelma ei toimi, jos jokin luokka on jäänyt luettelosta pois. Julistuksessa käy ilmi aktiviteetin luokka ja sen toiminto, intentti. Jos esimerkiksi sovelluksessa painetaan "Etsi" -painiketta, niin manifestissa on määritelty, mistä löytyy edellä mainitun "Etsi" -painikkeen toiminto. Ohjelma etsii mistä luokasta se suorittaa kyseisen ohjelmakoodin.

Android laitteille on kielikohtainen tuki. Google suosittelee sovelluskehityksessä käyttämään string-resourcea, jotta ohjelmaan saa helposti lisättyä muita kieliä. String-resource on ohjelman hakemistopuussa res -kansiossa sijaitseva joukko-values- kansioita ja niiden alla olevia strings.xml-tiedostoja, joihin on tarkoitus kirjoittaa kielikohtaiset sanat toisiaan vastaaville paikoille (kuva 17).

English strings (default locale), /values/strings.xml:

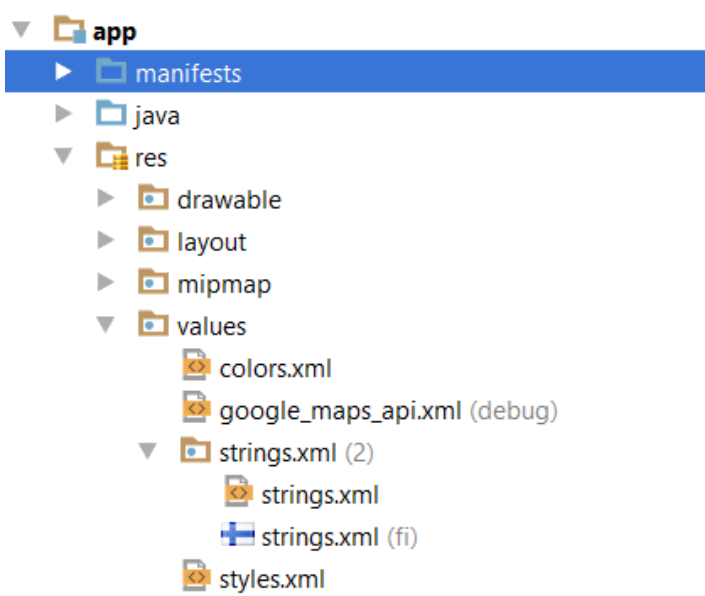
```
<resources>
  <string name="hello_world">Hello World!</string>
</resources>
```

Spanish strings (es locale), /values-es/strings.xml:

```
<resources>
  <string name="hello_world">¡Hola Mundo!</string>
</resources>
```

KUVA 17. strings.xml -tiedostot englanniksi ja espanjaksi

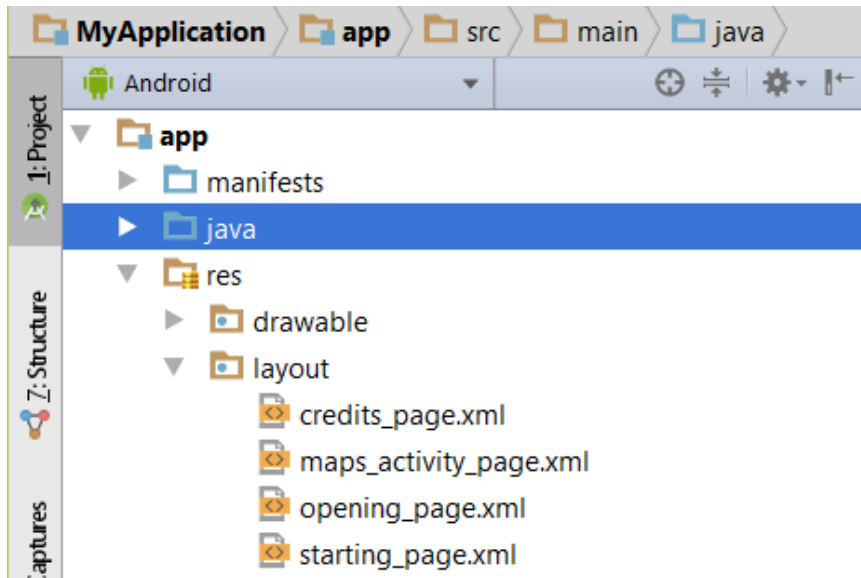
Laite osaa hakea oikean kielen sovellukselle, jos ohjelmaan on lisätty suosituk-
sen mukaisesti maakohtainen values-kansio (kuva 18). Se täytyy nimetä siten,
että tekstin "values" jälkeen tulee ISO-639-1-standardin mukainen maakoodi.
Edellisessä kuvassa molemmat xml-tiedostot ovat res-kansion alla omissa kan-
sioissaan, mutta samalla strings.xml -nimellä. Myös yksittäisen stringin nimi on
sama, mutta sisältö on eri. Tämä ohjelma tukee suomen ja englannin kieltä, jo-
ten hakemistoon on luotu values-fi-niminen kansio, josta ohjelma hakee sisäl-
lön, mikäli laitteessa on käytössä suomen kieli.



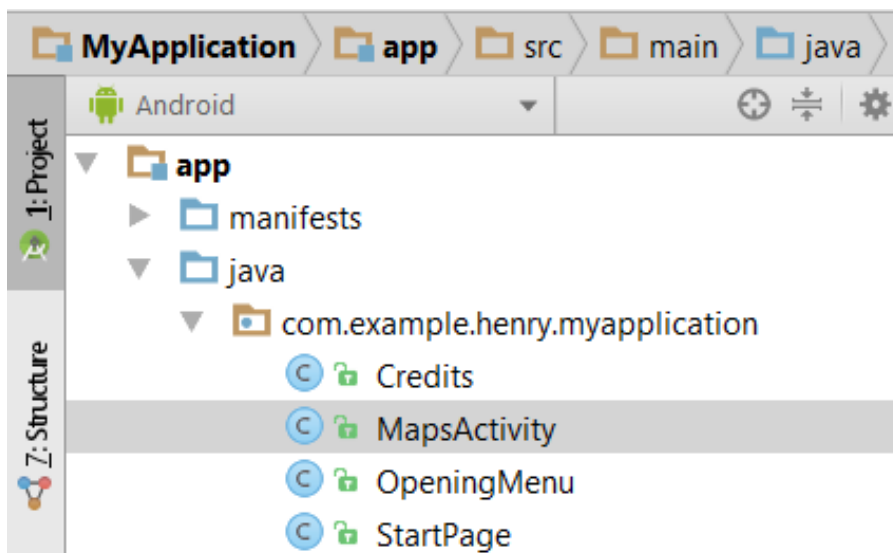
KUVA 18. Android Studion kansiorakenne, kohdassa values

Prototyypin ohjelmistorakenne koostuu neljästä XML-tiedostosta (kuva 19) ja neljästä Java-tiedostosta (kuva 20). XML -tiedostot ovat näkymien tiedostoja ja Java-tiedostot ovat luokkien tiedostot.

Kuten olio-ohjelmoinnissa yleensä, myös Androidille ohjelmoitaessa luokat kirjoitetaan omiksi tiedostoiksi (1). Luokat sijaitsevat omissa kansiossaan ja ne tunnustetaan .java -päätteestään.



KUVA 19. Android Studion kansiorakenne, kohdassa layout eli näkymä



KUVA 20. Android Studion kansiorakenne, kohdassa luokat

Prototyypiohjelmassa on neljä eri luokkaa, Credits, MapsActivity, OpeningMenu ja StartPage. Luokat on nimetty sen mukaan, mitä niiden nimestä päätellen niiden oletetaan tekevän.

OpeningMenu-luokka on aloitusaktiiviteetti, josta ohjelma käynnistyy, ja joka ennen sulkeutumistaan aloittaa uuden säikeen josta käynnistyy ohjelman valikkonäkymä. AndroidManifestissa aktiiviteetin kategoria-osio määrittelee sen, mistä ohjelman käynnistys aloitetaan (kuva 21). OpeningMenussa kategoria on LAUNCHER, kun muissa aktiiviteeteissa se on DEFAULT.

```
<activity
    android:name=".OpeningMenu"
    android:label="Map"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

KUVA 21. AndroidManifestissa oleva aloitusaktiiviteetti.

StartPage-luokasta päästään kaikkiin ohjelman sivuihin tai näkymiin kuvan 22 esittämällä tavalla.

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.starting_page);

    final Button buttonOne = (Button) findViewById(R.id.buttonStartCalculate);
    final Button buttonTwo = (Button) findViewById(R.id.buttonCredits);
    final Button buttonFour = (Button) findViewById(R.id.buttonExit);

    buttonOne.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(StartPage.this, MapsActivity.class);
            startActivity(intent);
        }
    });
};
```

KUVA 22. Osa ohjelmakoodia StartPage-luokasta.

MapsActivity on koko ohjelman tärkein luokka. Siinä luodaan yhteys Googlen karttapalveluun, luodaan laskuri, taustalla lasketaan aikaa, päivitetään sijainti

sekunnin välein ja näytetään sijainti omassa karttafragmentissa. Lopuksi tämä luokka huolehtii vielä kulutuksen näyttämisestä tästä näkymästä poistuttaessa. MapsActivityn koko ohjelmakoodi on luettavissa liitteenä.

Yleinen linjaus Android-sovelluskehityksessä on, että toiminnallisuudet kirjoitetaan Java-kielellä ja ulkoasu on kuvattu XML:llä. Sovelluksen jokainen näkymä, esimerkiksi sivu tai ikkuna, on siis XML:llä kirjoitettu, eli jokaista sivua kohden on yksi XML-tiedosto. Kuvassa 23 on aloitusruudun ohjelmakoodi.

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:background="@drawable/launhcer_image"
        android:id="@+id/imageView1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="80dp"/>

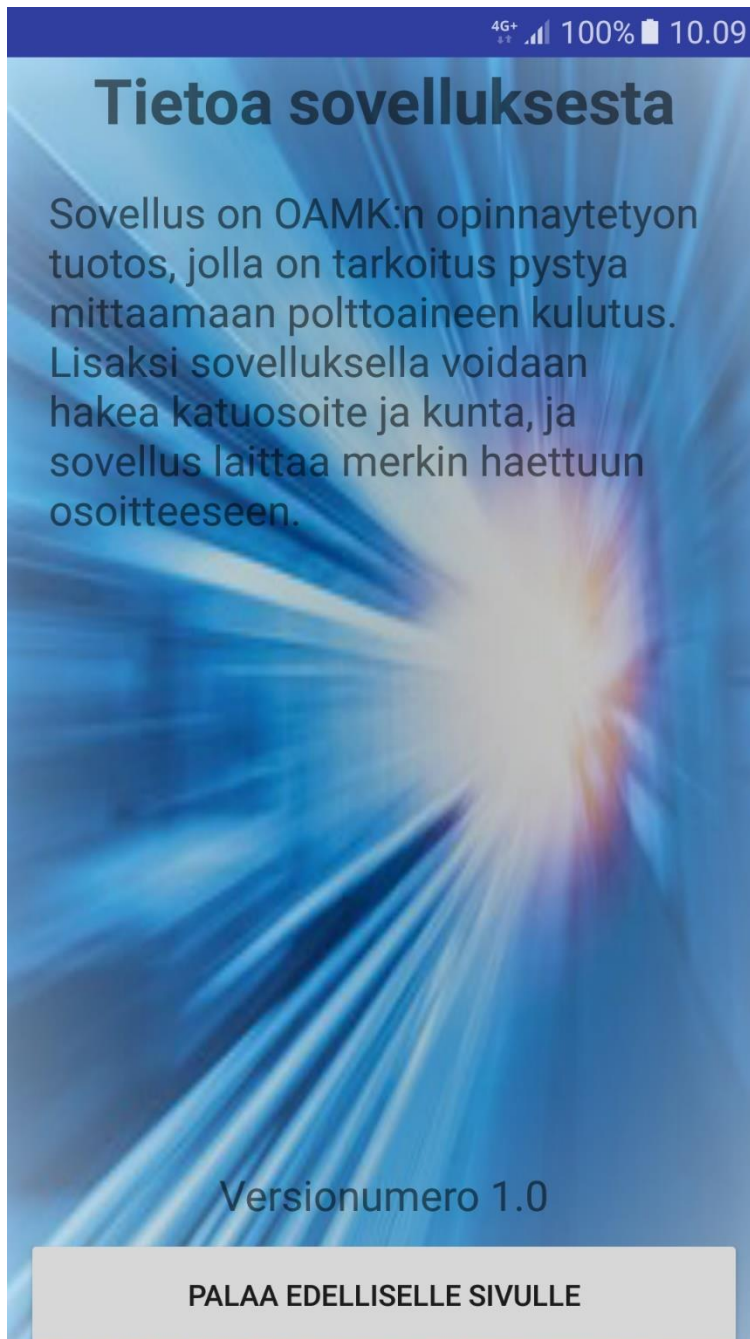
    <TextView
        android:text="@string/openingtext"
        android:textSize="20dp"
        android:layout_marginLeft="40dp"
        android:layout_marginRight="40dp"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/imageView1"
        android:layout_marginTop="43dp"
        android:id="@+id/opTV"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true" />

    <TextView
        android:text="@string/openingtext2"
        android:textSize="20dp"
        android:textStyle="bold"
        android:layout_marginLeft="40dp"
        android:layout_marginRight="40dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:id="@+id/opTV2"
        android:layout_below="@+id/opTV"
        android:layout_alignEnd="@+id/opTV"
        android:layout_alignParentStart="true" />

</RelativeLayout>
```

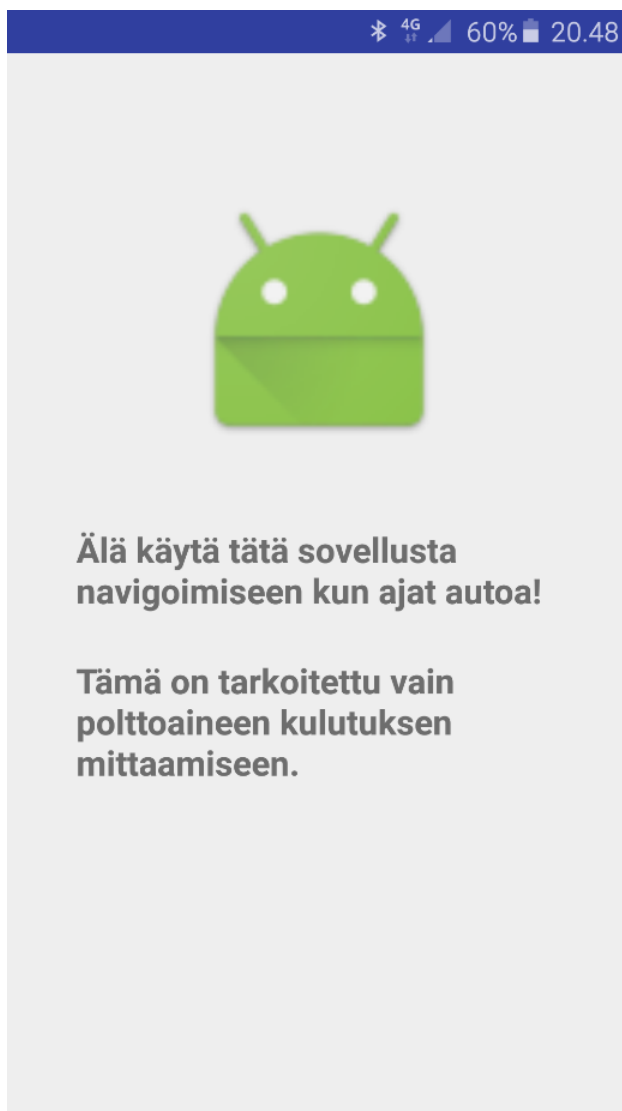
KUVA 23. Aloitusruudun XML-koodi

Credits-luokka avaa ohjelmassa sivun, jossa kerrotaan tietoa sovelluksesta. Aloitussivulla mainitaan kuinka ohjelmaa tulisi käyttää, Tietoa-sivulla kerrotaan versionumero ja tarkoitus, johon ohjelma on kehitetty (kuva 24).



KUVA 24. Tietoa-sivu.

Ohjelmassa esiintyy aloitusnäyttö, joka näkyy ohjelman suorituksen aikana vain kerran. Muut näkymät ovat mahdollista avata ja sulkea niin monta kertaa kuin tarvitaan. Aloitusnäyttö on useissa sovelluksissa ikkuna, jota näytetään ohjelman latautumisen aikana. Hyvin usein aloitusnäytössä näkyy yrityksen logo tai muuta informaatiota, esimerkiksi kyseisen ohjelman käyttötarkoitus. Tämän ohjelman aloitusnäyttö (kuva 25) antaa käyttäjälle tietoa ohjelman tarkoituksesta.



KUVA 25. Ruutukaappaus sovelluksen aloitusnäytöstä

7 JATKOKEHITYSMÄHDOLLISUUDET

Ohjelmaan on mahdollista kehitellä useita erilaisia ominaisuuksia, esimerkiksi ajoneuvon valinta, jolloin polttoaineen kulutuksen laskennassa käytetään erilaisia arvoja. Yksi ratkaisu on mahdollisesti sellainen, että käyttäjä syöttää sovellukselle ajoneuvon kulutuksen 100 kilometrillä, josta lasketaan sekuntikulutus. Tällä kertoimella ohjelma sitten laskee polttoaineen kulutuksen aiemmin selitetyllä menetelmällä. Eri ajoneuvot kuluttavat polttoainetta eri tahtiin, esimerkiksi auto, kuorma-auto, rekka, mopo, moottoripyörä, moottorikelkka, asuntoauto ja auto varustettuna peräkärryllä tai asuntovaunulla.

Toinen mahdollinen lisäominaisuus voisi olla sellainen, että ohjelma arvioi kulutuksen pisteestä A pisteeseen B. Ohjelma katsoisi senhetkisen sijainnin ja mitaisi etäisyyden siihen pisteeseen tai toisin sanoen osoitteeseen, jonka käyttäjä on antanut. Tällöin pitäisi mahdollisesti kehittää toinen laskutapa, koska ohjelma on kirjoitettu laskemaan kulutus sekunneissa, tai ohjelman tulisi ottaa selvää arvioidusta ajoajasta, jolloin laskutoimitus voitaisiin laskea suoraan siitä.

Yksi mahdollinen lisäominaisuus ohjelmalle on sellainen, että ohjelma piirtää kartalle kuljetun reitin. Tällä ominaisuudella on se etu, että voi tarkastella sitä, missä käyttäjä on kulkenut. Mikäli käyttäjä eksyy kulkiessaan, niin voi ainakin palata samaa reittiä takaisin lähtöpisteeseensä. Kaupunkialueella tai taajamassa eksyminen ei ole kovin todennäköistä, mutta esimerkiksi moottorikelkalla tai motocross-pyörällä ajelulla metsässä se on todennäköisempää.

Monessa ohjelmassa tiedot tallennetaan joko laitteen omaan muistiin tai esimerkiksi Googlen omaan tietokantaan. Tällöin käyttäjä joutuu yleensä luomaan Google -tunnukset ohjelman käyttöä varten. Yleensä tallennetun datan historiasta näytetään jonkinlainen kaavio, esimerkiksi kuinka monta kilometriä käyttäjä on liikkunut vaikka kuukauden aikana. Tällainen ominaisuus sopisi tähän ohjelmaan todella hyvin, jos rinnalla näkyy esimerkiksi polttoaineen kulutus ja hinta kyseisellä matkalla. Sovelluksen käyttäjä näkee, kuinka paljon autolla kulkeminen kuluttaa polttoainetta ja rahaa.

Toisaalta polttoaineen hinta vaihtelee melko usein. Saadaksesen ajankohtaisen hinnan käyttäjän pitäisi joko syöttää sovellukseen senhetkinen litrahinta tai ohjelman pitäisi hakea jostain osoitteesta tai tietokannasta polttoaineen hinta ennen jokaista ohjelman käynnistämistä.

Ohjelmaan voisi lisätä sellaisen ominaisuuden, että se jatkaisi toimintaansa taustalla. Kun Android-laitteessa painaa valikkopainiketta, ohjelma jää välimuistiin tai toisin sanoen taustalle. Tässä Autosovellus-ohjelmassa ei ole tällaista ominaisuutta, vaan laskuri pysähtyy heti, kun karttanäkymästä poistutaan.

Ohjelman ulkoasua voi muokata ja parantaa, esimerkiksi värit ja tekstitykset, fontit, kuvat, ja mahdollisesti lisätä äänet. Painikkeet näyttävät aivan peruspainikkeilta, eli niissä ei ole mitään omaa tai persoonallista. Toisaalta kauneus on katsojan silmässä, ja ehkä tähän ohjelmaan sopivat paremmin perusvärit ja kuvat kuin pastellinsävyt ja kaunokirjoitusfontti.

Ohjelmaan ulkoasun muuttamisessa voisi olla paikallaan lisätä valikkopainike näkymän ylälaitaan, josta painettaessa avautuu esimerkiksi "Tietoa ohjelmasta"-sivu tai "Anna palautetta"- tai "Arvostele sovellus" -painike, joka lähettää käyttäjän antaman arvostelun sovelluksen luojaan omaan sähköpostiin.

Viimeisenä etappina kehityskaassa on laittaa ohjelma ladattavaksi Google Play -kauppaan, jos haluaa nähdä, miten ohjelma arvostellaan. Vaikka ohjelmaan lisäisi ominaisuuden, jolla käyttäjät voisivat antaa palautetta, niin se ei mahdollisesti ole välttämätön palautteen saamiseksi, koska Google Play -kaupassa jokaisen käyttäjän on mahdollista arvioida lataamansa sovellus ja arvostelu näkyy julkisesti (23).

8 YHTEENVETO

Opinnäytetyön tekeminen on ollut suurempi projekti, kuin sen oli odottanut olevan. Hyvällä suunnittelulla olisi voinut välttää joitain ongelmia. Tämä näkyi ohjelman kirjoittamisessa, kun ei ollut miettinyt tarpeeksi jotain asian toteutusta. Kun myöhemmin oivaltaa jonkin asian, ohjelman korjaaminen tai parantaminen vie enemmän aikaa, kuin jos asian olisi miettinyt kunnolla ja esimerkiksi hahmotellut asian paperille.

Tiedon haku on jatkuvasti ollut läsnä ohjelman tekemisessä. Androidille on kyllä olemassa hyvä ja kattava tukisivusto, jolla on tullut käytyä todella useasti. Sen lisäksi myös Stackoverflow-sivusto on ollut hyvä paikka hakea ratkaisua ongelmaan, koska siellä ohjelmoijat sekä kysyvät neuvoja että antavat niitä.

Ohjelman luonteesta johtuen sitä on jouduttu testaamaan siten, että asunnosta on poistuttava ja lähdettävä liikkeelle. Enimmäkseen sitä on testattu kävellen, mutta olen joutunut testaamaan myös autolla liikkeessäni, koska sitä varten koko ohjelma on suunniteltu. Alkuun en havainnut ohjelmassa olevan sellaista virhettä, että se ei päivity jatkuvasti. Ohjelma käyttää GPS:ää eli satelliittipaikannusta määrittäläkseen käyttäjän sijainnin. Ohjelma kyllä haki sijainnin joka kerta kun se käynnistyi, mutta sijainnin täytyy päivittyä jatkuvasti. Koska ohjelman käyttötarkoituksen mukaisesti käyttäjän sijainnin täytyy muuttua aika ajoin, myös paikannuksen täytyy tapahtua riittävän nopein väliajoin. Ohjelma päivittää sijainnin sekunnin väliajoin, joten ohjelma kuluttaa melko paljon virtaa.

Ohjelmaa kirjoittaessa tuli monta kertaa tilanne, jolloin havaitsi jonkin seikan, miten olisi voinut toimia paremmin tai järkevämmiin. Yleensä siinä vaiheessa ohjelman toimintatavan muuttaminen oli jo työläämpää ja aikaa vievää, mutta se täytyi tehdä, jotta itsekin kykeni olemaan tyytyväinen sovellukseen. Ohjelma on suurimmaksi osaksi käyttökelpoinen, mutta ulkoasun parantamisella ja muuttaman toiminnallisuuden lisäämisellä siitä on mahdollista saada vielä parempi.

LÄHTEET

1. Harju, Jukka 2013. Android-ohjelmoinnin perusteet. Helsinki: Books on Demand GmbH.
2. Android. 2017. Wikipedia. Saatavilla: <https://fi.wikipedia.org/wiki/Android>. Hakupäivä 17.11.2016.
3. Näin paljon Android-laitteita maailmasta löytyy. Saatavilla: <http://www.uusi-suomi.fi/teknologia/120458-google-paljasti-nain-paljon-android-laitteita-maailmasta-loytyy>. Hakupäivä 21.11.16.
4. Ashton, Kevin. 22.9.2009. That 'Internet of Things' Thing. RFID Journal. Saatavilla: <http://www.rfidjournal.com/articles/view?4986>. Hakupäivä 18.2.2017.
5. Matkapuhelin. 2017. Wikipedia. Saatavilla: <https://fi.wikipedia.org/wiki/Matkapuhelin>. Hakupäivä 11.2.2017.
6. Älypuhelin. 2017. Wikipedia. Saatavilla: <https://fi.wikipedia.org/wiki/%C3%84lypuhelin>. Hakupäivä 19.2.2017.
7. Taulutietokone. 2017. Wikipedia. Saatavilla: <https://fi.wikipedia.org/wiki/Taulutietokone>. Hakupäivä 19.2.2017.
8. Käyttöjärjestelmä. 2017. Wikipedia. Saatavilla: <https://fi.wikipedia.org/wiki/K%C3%A4ytt%C3%B6j%C3%A4rjestelm%C3%A4>. Hakupäivä 17.11.16.
9. Turi, Jon. 22.6.2014. T-Mobile G1. Saatavilla: <https://www.engadget.com/2014/06/22/gadget-rewind-2008-t-mobile-g1-htc-dream/>. Hakupäivä 31.3.17.
10. Android Marshmallow. 2017. AppsGeyser. Saatavilla: <http://www.appsgeyser.com/blog/android-m-arrived-marshmallows/>. Hakupäivä 21.11.15.

11. Android (operating system). 2017. Wikipedia. Saatavilla: https://en.wikipedia.org/wiki/Android_%28operating_system%29. Hakupäivä 14.11.2015.
12. Android Studio. Android Developers. Saatavilla: <https://developer.android.com/sdk/index.html>. Hakupäivä 6.11.15
13. Microsoft Windows. 2017. Wikipedia. Saatavilla: https://fi.wikipedia.org/wiki/Microsoft_Windows. Hakupäivä 20.2.2017.
14. Unix. Unix-perusteet. Saatavilla: http://users.metropolia.fi/~kuivi/unix/perus_unix/unix_perusteet.pdf. Hakupäivä 17.2.2017.
15. Polttoaineen kulutus. 2017. Laskurit.fi. Saatavilla: <http://www.laskurit.fi/polttoaineen-kulutus/>. Hakupäivä 31.3.2017.
16. Selvät sävelet polttoaineen kulutukseen. Neste. Saatavilla: <https://www.neste.fi/artikkeli/selvat-savelet-polttoaineen-kulutukseen>. Hakupäivä 5.2.2017.
17. Vuollet, Anna-Mari. 2011. Säästä rahaa ja moottoria. Iltalehti. Saatavilla: http://www.iltalehti.fi/autot/2011121114881183_au.shtml. Hakupäivä 3.2.2017.
18. Suuri polttoaineenkulutus. RVS. 2009. Saatavilla: <http://www.rvs.fi/diagnostic/page34.html>. Hakupäivä 5.2.2017.
19. Fragments. Android Developers. Saatavilla: <https://developer.android.com/guide/components/fragments.html>. Hakupäivä 7.2.2017.
20. Google Maps. 2017. Wikipedia. Saatavilla: https://fi.wikipedia.org/wiki/Google_Maps. Hakupäivä 20.2.2017.
21. Google Location Services. Android Developers. Saatavilla: <https://developer.android.com/guide/topics/location/index.html>. Hakupäivä 20.2.2017.
22. Android Manifest. Android Developers. Saatavilla: <https://developer.android.com/guide/topics/manifest/manifest-intro.html>. Hakupäivä 6.2.2017.

23. Google Play. Saatavilla: <https://play.google.com/store/apps>. Hakupäivä
20.2.2017.

AndroidManifest

```

<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.henry.myapplication">

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">

        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />
        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="com.example.henry.myapplication.MAPSACTIVITY" />
                <category android:name="android.intent.action.DEFAULT" />
            </intent-filter>
        </activity>

        <activity android:name=".Credits"
            android:label="Map"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="com.example.henry.myapplication.CREDITS"/>
                <category android:name="android.intent.category.DEFAULT"/>
            </intent-filter>
        </activity>

        <activity
            android:name=".StartPage"
            android:label="@string/title_activity_maps"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="com.example.henry.myapplication.STARTPAGE" />
                <category android:name="android.intent.action.DEFAULT" />
            </intent-filter>
        </activity>

        <activity
            android:name=".OpeningMenu"
            android:label="@string/title_activity_maps"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity><!-- ATTENTION: This was auto-generated to add Google Play services
to your project for
App Indexing. See https://g.co/AppIndexing/AndroidStudio for more information. -->
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

    </application>
</manifest>

```

starting_page

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:id="@+id/RL01">

```

```

<RelativeLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:background="#aa9898"
    android:id="@+id/LL02"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true">

    <Button
        android:text="@string/credits"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:id="@+id/buttonCredits"
        android:layout_gravity="center_vertical"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

    <Button
        android:text="@string/exit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/buttonExit"
        android:layout_weight="1"
        android:onClick="closeProgram"
        android:layout_gravity="center_vertical"
        android:layout_alignParentBottom="true"
        android:layout_alignParentStart="true" />

    <Button
        android:text="@string/start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:id="@+id/buttonStartCalculate"
        android:layout_gravity="center_vertical"
        android:layout_centerVertical="true"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="20dp" />

</RelativeLayout>
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/mapsicon"
    android:id="@+id/imageView"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    android:layout_above="@+id/LL02"
    android:layout_alignParentEnd="true"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:layout_marginBottom="50dp"
    android:layout_marginTop="50dp" />

</RelativeLayout>

```

opening_page

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:background="@drawable/launhcer_image"
        android:id="@+id/imageView1"

```

```

        android:layout_centerHorizontal="true"
        android:layout_marginTop="80dp"/>

<TextView
    android:text="@string/openingtext"
    android:textSize="20dp"
    android:layout_marginLeft="40dp"
    android:layout_marginRight="40dp"
    android:textStyle="bold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/imageView1"
    android:layout_marginTop="43dp"
    android:id="@+id/opTV"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true" />

<TextView
    android:text="@string/openingtext2"
    android:textSize="20dp"
    android:textStyle="bold"
    android:layout_marginLeft="40dp"
    android:layout_marginRight="40dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="25dp"
    android:id="@+id/opTV2"
    android:layout_below="@+id/opTV"
    android:layout_alignEnd="@+id/opTV"
    android:layout_alignParentStart="true" />

</RelativeLayout>

```

credits_page

```

<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/picture1"
    android:id="@+id/llocredits">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/info_header"
        android:id="@+id/textView"
        android:textAlignment="center"
        android:textStyle="bold"
        android:textSize="30dp"
        android:gravity="center_horizontal"
        android:layout_weight="0"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/credits_txt"
        android:id="@+id/textView2"
        android:layout_weight="0"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp"
        android:textSize="20dp"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        android:gravity="center|bottom"
        android:orientation="vertical" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

```

```

        android:text=""
        android:id="@+id/textView3"
        android:layout_weight="0.3"
        android:textSize="20dp"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/version"
    android:id="@+id/textView4"
    android:layout_weight="0.01"
    android:textSize="20dp"/>

<Button
    android:text="@string/back"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:id="@+id/creditsButton"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_gravity="center_horizontal|center"/>

</LinearLayout>

</LinearLayout>

```

maps_activity_page

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:id="@+id/RLO">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:id="@+id/RLO_buttons"
        android:background="#555">

        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/location_textview"
            android:layout_centerVertical="true"
            android:layout_alignParentStart="true"
            android:hint="@string/cityandaddress"/>

        <Button
            android:text="@string/search"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/searchButton"
            android:layout_centerVertical="true"
            android:layout_alignParentEnd="true"
            android:onClick="onSearch" />
    </RelativeLayout>

    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        tools:context="com.example.henry.myapplication.MapsActivity"
        android:layout_below="@+id/LLO"
        android:layout_above="@+id/LLO2"
        map:cameraZoom="18"
        map:uiZoomControls="true"
        android:layout_width="match_parent"
        android:layout_height="350dp" />

</LinearLayout>

```

```
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="10dp"
        android:layout_below="@+id/RLO_buttons"
        android:layout_centerHorizontal="true"
        android:background="#0000"
        android:id="@+id/IL0">
</LinearLayout>

<RelativeLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:background="#0000"
    android:id="@+id/IL02"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true">

    <Button
        android:text="@string/close"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:id="@+id/closeButton" />

    <TextView
        android:text="@string/speed"
        android:layout_alignParentRight="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tvSpeed"
        android:layout_alignParentEnd="true"
        android:layout_alignParentTop="true"
        android:layout_toEndOf="@+id/closeButton"
        android:textAppearance="@style/TextAppearance.AppCompat"
        android:textAlignment="center"
        android:gravity="center_vertical"
        android:textSize="25dp"
        android:textStyle="italic"
        android:layout_alignBottom="@+id/closeButton"
        android:layout_alignStart="@+id/tvConsumption" />

    <TextView
        android:text="@string/consumeinfo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tvConsumption"
        android:layout_alignParentEnd="true"
        android:layout_below="@+id/tvSpeed"
        android:textAppearance="@style/TextAppearance.AppCompat"
        android:textAlignment="center"
        android:gravity="center_vertical"
        android:layout_alignParentBottom="true"
        android:layout_toEndOf="@+id/closeButton" />

    <TextView
        android:text=""
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tvNotification3"
        android:textAppearance="@style/TextAppearance.AppCompat"
        android:textAlignment="center"
        android:gravity="center_vertical"
        android:layout_alignParentBottom="true"
        android:layout_below="@+id/closeButton"
        android:layout_toEndOf="@+id/tvNotification2"
        android:layout_marginLeft="10dp" />

    <TextView
        android:text="@string/drivetime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:textAlignment="center"
        android:layout_marginLeft="5dp"
        android:textAppearance="@style/TextAppearance.AppCompat"
        android:textSize="15dp"
```

```

        android:gravity="center_vertical"
        android:id="@+id/tvNotification2"
        android:layout_below="@+id/closeButton"
        android:layout_alignParentBottom="true" />
    </RelativeLayout>
</RelativeLayout>

```

OpeningMenu

```

package com.example.henry.myapplication;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;

public class OpeningMenu extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.opening_page);

        Thread thread = new Thread()
        {
            public void run()
            {
                try
                {
                    sleep(2000);
                }
                catch (Exception e)
                {
                    Toast.makeText(OpeningMenu.this, "Failed!",
Toast.LENGTH_SHORT).show();
                }
                finally
                {
                    Intent startApp = new Intent(OpeningMenu.this, StartPage.class);
                    startActivity(startApp);
                }
            }
        };
        thread.start();
    }

    @Override
    protected void onPause() {
        super.onPause();
        finish();
    }
}

```

StartPage

```

package com.example.henry.myapplication;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class StartPage extends Activity {

```

```

@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.starting_page);

    final Button buttonOne = (Button) findViewById(R.id.buttonStartCalculate);
    final Button buttonTwo = (Button) findViewById(R.id.buttonCredits);
    final Button buttonFour = (Button) findViewById(R.id.buttonExit);

    buttonOne.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(StartPage.this, MapsActivity.class);
            startActivity(intent);
        }
    });
    buttonTwo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(StartPage.this, Credits.class);
            startActivity(intent);
        }
    });
    buttonFour.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v)
        {
            finishAffinity();
            System.exit(0);
            // finish();
        }
    });
}
}

```

MapsActivity

```

package com.example.henry.myapplication;

import android.*;
import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.os.CountDownTimer;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.FusedLocationProviderApi;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

import java.io.IOException;
import java.util.List;
import java.util.concurrent.TimeUnit;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback, LocationListener,

```



```

        GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener
    {
        private GoogleMap mMap;
        private FusedLocationProviderApi locationProviderApi = LocationServices.FusedLoca-
tionApi;
        private GoogleApiClient googleApiClient;
        private LocationRequest locationRequest;
        public static final int MILLISECONDS_PER_SECOND = 1000;

        public CountdownTimer myTimer;

        public static TextView tvSpeed;
        public static TextView tvNotification2;
        public static TextView tvNotification3;
        public static TextView tvConsumption;

        double oldLocationX = 0.0;
        double oldLocationY = 0.0;

        final long totalSeconds = 7200;
        final long intervalSeconds = 1;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.maps_activity_page);

            tvConsumption = (TextView) findViewById(R.id.tvConsumption);
            tvNotification2 = (TextView) findViewById(R.id.tvNotification2);
            tvNotification3 = (TextView) findViewById(R.id.tvNotification3);
            tvSpeed = (TextView) findViewById(R.id.tvSpeed);
            tvSpeed.setText("");

            // Obtain the SupportMapFragment and get notified when the map is ready to be
            used.

            final SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragment-
Manager()
                .findFragmentById(R.id.map);
            mapFragment.getMapAsync(this);

            googleApiClient = new GoogleApiClient.Builder(this)
                .addApi(LocationServices.API)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .build();

            locationRequest = new LocationRequest();

            locationRequest.setFastestInterval(MILLISECONDS_PER_SECOND);
            locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

            if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LO-
CATION)
                != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission
            (this, Manifest.permission.ACCESS_COARSE_LOCATION) != Pack-
ageManager.PERMISSION_GRANTED) {
                // TODO: Consider calling
                //     ActivityCompat#requestPermissions
                // here to request the missing permissions, and then overriding
                //     public void onRequestPermissionsResult(int requestCode, String[] per-
missions,
                //                                     int[] grantResults)
                // to handle the case where the user grants the permission. See the documen-
tation
                // for ActivityCompat#requestPermissions for more details.
                return;
            }

            Button closeButton = (Button) findViewById(R.id.closeButton);

```

```

closeButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        closeProgram();
    }
});
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    myTimer = new CountdownTimer(totalSeconds * 1000, intervalSeconds * 1000) {

        long timerresult = 0;
        long secondBeenUsed = 0;
        int seconds = 0;

        int min = 0;
        int hours = 0;

        public void onTick(long millisUntilFinished) {

            timerresult = (totalSeconds * 1000 - millisUntilFinished) / 1000;

            min = (int) TimeUnit.SECONDS.toMinutes(timerresult);
            seconds = (int) TimeUnit.SECONDS.toSeconds(timerresult) % 60;
            hours = (int) TimeUnit.SECONDS.toHours(timerresult) / 3600 ;

            String time = String.format("%02d:%02d:%02d", hours, min, seconds);
            tvNotification3.setText(time);

            secondBeenUsed = timerresult;
        }

        public boolean showConsume()
        {
            long consume = secondBeenUsed;
            double amount;
            amount = (0.5 + (consume * 0.00167));

            double d = amount;
            String formattedData = String.format("%.04f", d);

            String cInfo = getString(R.string.consumeinfo);
            String literInfo = getString(R.string.literinfo);

            double prize;
            prize = (amount * 1.33);
            String formattedDataPrize = String.format("%.02f", prize);

            Toast.makeText(MapsActivity.this, cInfo + " " + formattedData + " " +
literInfo + "\n\n" + "Hinta: " + formattedDataPrize + " €", Toast.LENGTH_SHORT).show();

            //below works OK!
            //Toast.makeText(MapsActivity.this, cInfo + " " + (1 + (consume *
0.00167)) + " " + literInfo, Toast.LENGTH_SHORT).show();

            return false;
        }

        public void onFinish()
        {
            showConsume();
            cancel();
        }
    };

    myTimer.start();

    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LO-
CATION)

```

```

        != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(
sion
            (this, Manifest.permission.ACCESS_COARSE_LOCATION) != Pack-
ageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //     ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //     public void onRequestPermissionsResult(int requestCode, String[] per-
missions,
        //                                     int[] grantResults)
        // to handle the case where the user grants the permission. See the documen-
tation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }

    mMap.setMyLocationEnabled(true);
}

public void onSearch(View view) {
    EditText location_textview = (EditText) findViewById(R.id.location_textview);
    String location = location_textview.getText().toString();

    List<Address> addressList = null;

    if (!location.equals("")) {
        Geocoder geocoder = new Geocoder(this);
        try {
            addressList = geocoder.getFromLocationName(location, 1);
        } catch (IOException e) {
            e.printStackTrace();
        }

        Address address = addressList.get(0);
        LatLng latLng = new LatLng(address.getLatitude(), address.getLongitude());
        mMap.addMarker(new MarkerOptions().position(latLng).title(location));
        mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
    }

    else {
        Toast.makeText(this, R.string.validlocation, Toast.LENGTH_LONG).show();
    }
}

public void closeProgram()
{
    myTimer.onFinish();
    myTimer.cancel();
    finish();
}

@Override
public void onConnected(@Nullable Bundle bundle) {
    requestLocationUpdates();
}

private void requestLocationUpdates() {
    if (ActivityCompat.checkSelfPermission(this, android.Manifest.permission.AC-
CESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(
sion
            (this, android.Manifest.permission.ACCESS_COARSE_LOCATION) != Pack-
ageManager.PERMISSION_GRANTED) {

        return;
    }
    LocationServices.FusedLocationApi.requestLocationUpdates(googleApiClient, loca-
tionRequest, this);
}

@Override
public void onConnectionSuspended(int i) {
}

```

```

@Override
protected void onStart() {
    super.onStart();
    googleApiClient.connect();
}

@Override
protected void onStop() {
    super.onStop();
    googleApiClient.disconnect();
}

@Override
protected void onResume() {
    super.onResume();
    if (googleApiClient.isConnected()) {
        requestLocationUpdates();
        myTimer.start();
    }
}

@Override
protected void onPause() {
    super.onPause();
    LocationServices.FusedLocationApi.removeLocationUpdates(googleApiClient, this);
    myTimer.onFinish();
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
}

@Override
public void onLocationChanged(Location location) {

    // car consumes 1 liter per 18km
    // 1 liter per minute, so every second consumes 16,67 ml of gasoline / 0,001667
liter

    // first, let's calculate speed
    //float gasolinePerSecondInMilliliters = 16.67f;
    float speed = (float) location.getSpeed();
    float speedInKmh = (float) (speed * 3600 / 1000);
    String formattedString = String.format("%.2f", speedInKmh);

    String speedString = String.format("%.1f", speed);
    tvSpeed.setText(formattedString + " /kmh");

    // now we get our X and Y coordinates
    double locationX = location.getLatitude();
    double locationY = location.getLongitude();
    LatLng newerLatLng = new LatLng(locationX, locationY);

    mMap.animateCamera(CameraUpdateFactory.newLatLng(newerLatLng));

    oldLocationX = locationX;
    oldLocationY = locationY;
}
}

```

Credits

```

package com.example.henry.myapplication;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Credits extends Activity {
    @Override

```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.credits_page);

    Button creditsButton = (Button) findViewById(R.id.creditsButton);
    creditsButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });
}
```