

# Windows Phone 8 Jokamies metsänmittaussovellus

Samppa Tiihonen

Opinnäytetyö  
05 / 2014

Ohjelmistotekniikan koulutusohjelma  
Tekniikan ja liikenteen ala





Tekijä(t) TIIHONEN, Samppa	Julkaisun laji Opinnäytetyö	Päivämäärä 14.5.2014
	Sivumäärä 45	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty ( X )
Työn nimi Windows Phone 8 Jokamies metsänmittaussovellus		
Koulutusohjelma Ohjelmistotekniikan koulutusohjelma		
Työn ohjaaja(t) PIETIKÄINEN, Kalevi		
Toimeksiantaja(t) JAMK, "Metsävaramittaus nuorissa kunnostamattomissa metsissä"-projekti		
Tiivistelmä <p>Opinnäytetyössä jatkokehittiin Jyväskylän Ammattikorkeakoulun Metsävaramittaus-projektissa Android alustalle HTML5 tekniikoilla kehitettyä metsänmittaus mobiilisovellusta ja tehtiin siitä Windows Phone 8 toteutus. Tavoitteena oli saada sovellus toimimaan samalla tavalla Windows Phone 8 alustalla kun Android alustallakin. Tämän lisäksi sovellukselle oli tehtävä Bluetooth liitännäinen tiedonsiirtoa varten. Sovelluksesta tehtiin myös natiivi C-sharp kielellä toteutettu versio ja näiden suorituskyky verrattiin toisiinsa.</p> <p>Yhtenä osa-alueena opinnäytetyössä oli myös sovelluksen testaus sekä mittaustulosten tarkastelu. Sovelluksella tehtiin mittauksia testikoealoilla ja tuloksia verrattiin aiempiin mittauksiin. Mittaustulosten perusteella määriteltiin parannusehdotuksia sovellukseen.</p> <p>Projektin tavoitteet saavutettiin ja sovellus saatiin toimivaksi Windows Phone 8 alustalla. Opinnäytetyöhön on listattu ominaisuuksia, joita jatkossa voisi kehittää.</p>		
Avainsanat (asiasanat) Windows Phone, Bluetooth, PhoneGap, Cross-platform, HTML5, JavaScript, Metsänmittaus, C#		
Muut tiedot		



Author(s) TIIHONEN, Samppa	Type of publication Bachelor's Thesis	Date 14.5.2014
	Pages 45	Language Finnish
		Permission for web publication ( X )
Title Mobile application for Forest Resource Measurement on Windows Phone 8 platform		
Degree Programme Software Engineering		
Tutor(s) PIETIKÄINEN, Kalevi		
Assigned by JAMK, "Forest resource measurement in untreated forests"-project		
Abstract <p>The thesis focused on further development of a mobile application for Forest Resource Measurement project. The application was developed using HTML5 technique and was to be used on Windows Phone 8 platform. The target was to get the application to work on Windows Phone 8 platform in the same way as on Android platform. A Bluetooth plugin also had to be made for data transfer. The application was also written in native C sharp programming language and there is a comparison of the performance between the two versions in the thesis.</p> <p>One part of the thesis discusses the testing of the application and the analysis of the measurement results. The measurements with the application were made at test stands and the results were compared to previous measurements. Improvements to the application were made based on the measurement results.</p> <p>The goals of the project were achieved and the application was fully functional on Windows Phone 8 platform. The thesis includes a list of features that can be further developed.</p>		
Keywords Windows Phone, Bluetooth, PhoneGap, Cross-platform, HTML5, JavaScript, C#		
Miscellaneous		

## Sisältö

Sanasto .....	4
1 Opinnäytetyön lähtökohdat .....	6
2 Metsänmittaus .....	6
2.1 Metsät Suomessa .....	6
2.2 Metsätilojen pirstoutuminen .....	7
3 Metsänmittausmenetelmät .....	7
3.1 Yleistä.....	7
3.2 Lentokonekeilaus.....	8
3.3 Metsänmittauksen mobiilisovellukset .....	9
4 Windows Phone 8 .....	10
4.1 Yleistä.....	10
4.2 Puhelimet.....	11
4.3 Windows Phone 8 sovelluskehitys .....	11
4.3.1 Visual Studio .....	11
4.3.2 C#.....	12
4.3.3 HTML5 .....	12
4.3.4 PhoneGap.....	13
5 Työn toteutus .....	15
5.1 Bluetooth plugin .....	15
5.2 Windows Phone -version ero Android versioon .....	16
5.3 Mittausmenetelmät.....	17
5.3.1 Puustomittaus .....	17
5.3.2 Alikasvoslaskenta .....	17
5.3.3 Mittaustulokset .....	18

	2
5.4 Käyttöliittymä .....	22
5.4.1 Alkuperäinen käyttöliittymä.....	22
5.4.2 Graafinen käyttöliittymä .....	22
5.4.3 Responsiivisuus .....	23
5.5 C# versio .....	24
5.5.1 Yleistä .....	24
5.5.2 C# version perusta.....	25
5.5.3 Tiedonsiirto .....	25
5.5.4 Kartat ja navigointi .....	26
5.5.5 C# ja HTML5 sovelluksien erot .....	28
5.6 Taustajärjestelmä .....	29
5.6.1 Yleistä .....	29
5.6.2 Serveri ja puhelin.....	30
5.6.3 Kuvion lisääminen serverille.....	30
5.6.4 Serverin tekemät laskutoimitukset .....	32
6 Työn tulokset .....	33
7 Pohdinta .....	34
Lähteet.....	36
Liitteet .....	38
Liite 1. JokaMies-sovelluksen käyttöohje.....	38
Liite 2. Ilmakuva Liikalan koealueesta .....	44
Liite 3. Laserkeilaus pistepilvi Liikalasta .....	45

## Kuviot

KUVIO 1. Laserkeilauspisteet (Pihalava 2008).....	9
KUVIO 2. Puhelimen rekisteröinti .....	12
KUVIO 3. Visual Studio-projektin luonti .....	13
KUVIO 4. Visual Studio Solution Explorer.....	14
KUVIO 5. Puuston raivaukseen kuluva aika tuntia/ha .....	18
KUVIO 6. Mittaustulos Masser Sonar Caliper:lla.....	19
KUVIO 7. Mittaustulos JokaMies-mobiilisovelluksella .....	19
KUVIO 8. Ensimmäinen käyttöliittymä .....	22
KUVIO 9. Käyttöliittymä toisessa vaiheessa .....	23
KUVIO 10. Käyttöliittymän kolmas versio selaimessa .....	24
KUVIO 11. Sovelluksien erot. Vasemmalta oikealle: C# pääsivu, C# puustotulkinta sivu, HTML5 pääsivu ja HTML5 puustotulkinta sivu.....	25
Kuvio 12. Puhelimen ja serverin välinen toiminta .....	30
KUVIO 13. Kuviotietojen lataus tiedostoista .....	31
KUVIO 14. Kuvion lisääminen manuaalisesti.....	32
KUVIO 15. Kesquivirheen kaava .....	33

## Taulukot

TAULUKKO 1. Puiden väliset etäisyydet.....	20
TAULUKKO 2. Puiden väliset etäisyydet.....	21
TAULUKKO 3. Sovelluksen latausajat .....	28
TAULUKKO 4. Sovelluksen ruudunpäivitysnopeudet eri sivuilla.....	29

## SANASTO

### **Android**

Googlen puhelimille ja muille mobiililaitteille kehittämä maailman suosituin mobiilikäyttöjärjestelmä. Käyttöjärjestelmä perustuu Linux ytimeen ja sille kehitetään ohjelmia pääasiassa Java kielellä.

### **Glonass**

Glonass on Venäjän ilmavoimien ylläpitämä, GPS-paikannusjärjestelmää vastaava satelliittien verkosto.

### **GPS**

GPS, eli Global Positioning System on Yhdysvaltojen puolustusvoimien paikannussatelliiteista koostuva järjestelmä, joka tarjoaa paikka- ja aikatietoja. GPS vastaanotin laskee sijaintinsa satelliiteilta tulevien paikka- ja aikatietojen avulla.

### **HTML5**

HTML5 tarkoittaa HTML-merkintäkielen uusinta versiota, jota käytetään verkkosivujen tekemiseen. Termiä käytetään yleisesti myös kuvaamaan nykyaikaisia Web tekniikoita (JavaScript, CSS).

### **Metsälö**

Metsälö on taloudellinen ja hallinnollinen käsite saman omistajan hallussa olevien metsien muodostamista kokonaisuuksista. Metsälöön voi kuulua yksi tai useampi tila, joka on merkitty kiinteistörekisteriin.

### **.NET**

.NET Framework on Microsoftin kehittämä, Visual Studiossa käytettävä komponenttikirjasto. .NET tukee noin 20 ohjelmointikieltä, joista yleisimpiä ovat C# sekä VisualBasic.Net

**Responsiivisuus**

Responsiivisuudella tarkoitetaan käyttöliittymän kykyä mukautua eri näyttökokojen mukaisesti. Esimerkiksi responsiivisesta verkkosivusta ei tarvitse tehdä erikseen mobiiliversiota, koska se osaa mukautua mobiililaitteen näytölle automaattisesti.

**Windows Phone**

Microsoftin puhelimille ja muille mobiililaitteille kehittämä mobiilikäyttöjärjestelmä, jonka uusin versio on Windows Phone 8. Ohjelmia Windows Phonelle kirjoitetaan pääasiassa Microsoftin kehittämällä C# kielellä.



## 1 OPINNÄYTETYÖN LÄHTÖKOHDAT

Opinnäytetyön tarkoituksena oli tehdä mobiilisovellus Windows Phone 8 -alustalle, joka helpottaa kunnostamattomien metsien tarvitseman työmäärän arviointia ja inventointia. Sovellus oli alun perin tarkoitus toteuttaa jatkokehittämällä Metsävaraprojektissa aiemmin kehitettyä puustotietokonesovellusta, joka oli kehitetty Windows Phone 7 -alustalle. Samaan aikaan oli kuitenkin kehitteillä HTML5-pohjainen JokaMies-puustonmittaussovellus Androidille, joten toimeksiantaja ehdotti, että kyseistä sovellusta alettaisiin kehittämään myös Windows Phonelle.

Työn pääasialliseksi päämääräksi muodostui Bluetooth-liitännäisen tekeminen Windows Phonelle ja JokaMies sovelluksen kehittäminen HTML5-tekniikoilla. Työhön kuului myös sovelluksen testausta metsässä käytännön tilanteissa.

## 2 METSÄNMITTAUS

### 2.1 Metsät Suomessa

Suomessa on 75 prosenttia maan pinta-alasta metsää, mikä tekee Suomesta Euroopan metsäisimmän maan. Suomessa on metsissä 2306 miljoonaa kuutiometriä puuta ja puuston vuotuinen kasvu on yli 100 miljoonaa kuutiota vuodessa. Puuston poistuma on keskimäärin 68 miljoonaa kuutiota, joka on alle 70 % vuotuisesta kasvusta. (Metsät kasvavat vauhdikkaasti – hoitotoimien intensiteetti vaihtelee 2012.)

Hoitamatonta metsää on metsäalasta noin 5 miljoonaa hehtaaria eli noin 30 %. 1,6 miljoonaa hehtaaria näistä metsistä tarvitsisi kiireellistä hakkuuta tai hoitoa (Metsät kasvavat vauhdikkaasti – hoitotoimien intensiteetti vaihtelee 2012). Nämä hoitokohteet olisi paikannettava hoitotöiden edistämiseksi (Pesonen, Korhonen, Tuominen, Maltamo & Lukkarinen 2007).

## 2.2 Metsätilojen pirstoutuminen

Suomessa on Verohallituksen tietojen mukaan noin 440 000 yksityistä metsälöä, joista noin kolmasosa on alle kaksi ja puoli hehtaaria, toinen kolmannes alle 20 hehtaaria ja viimeinen kolmannes yli 20 hehtaaria. Keskimäärin yhden metsälön pinta-ala on 30 hehtaaria. Yksityisten henkilöiden omistuksessa olevat metsät muodostavat 60 % koko maan metsistä ja teollisuudessa käytettävä yksityismetsistä saatava puumäärä on suunnilleen saman verran. (Korpilahti 2009.)

Pääelinkeinonaan metsänhoitoa harjoittavalla täytyy olla vähintään 200 hehtaaria metsää, jotta se olisi taloudellisesti kannattavaa. Suomen yksityismetsistä näitä on ainoastaan kymmenesosa. Metsäteollisuus tarvitsee tasaista puuvirtaa, jota ne saavat suurilta metsätiloilta, jotka myyvät puuta säännöllisesti ja suuria määriä. Pienien metsäpalstojen omistajat usein odottavat parasta myyntihetkeä jopa vuosia, mikä johtaa epäsäännölliseen myyntiin.

Suomen kansallisen metsäohjelman tavoitteena on saada nostettua keskimääräisen yksityismetsälön keskikoko 30 hehtaarista 50 hehtaariin vuoteen 2015 mennessä. Ruotsissa tähän tavoitteeseen on jo päästy metsien jakamisen kieltävän lain avulla. Suomessa käytetyt houkuttimet ja pakotteet eivät ole toimineet. (Tikkanen 2010.)

## 3 METSÄNMITTAUSMENETELMÄT

### 3.1 Yleistä

Metsän inventointi on perinteisesti tehty paikan päällä metsässä. Apuna on voitu käyttää apuvälineitä kuten vuonna 1948 kehitettyä relaskooppiä, jolla saadaan arvioitua kuvion puuston pohjapinta-ala. Runkotilavuus saadaan määriteltyä relaskooppi-aulukosta, kun määritetään myös puiden pituus ja pääpuulaji. (Relaskooppi 2013.)

Nykyaikaisissa metsien inventoinnissa käytetään useampaa eri tiedon lähdettä ja tätä tekniikkaa kutsutaan monilähteiseksi metsien inventoinniksi. Siinä käytetään maastotietojen lisäksi apuna satelliittikuvia, karttoja ja korkeusmalleja. Karttatietojen avulla erotetaan metsätalousmaa ja muu maa ja korkeusmallien avulla vältetään

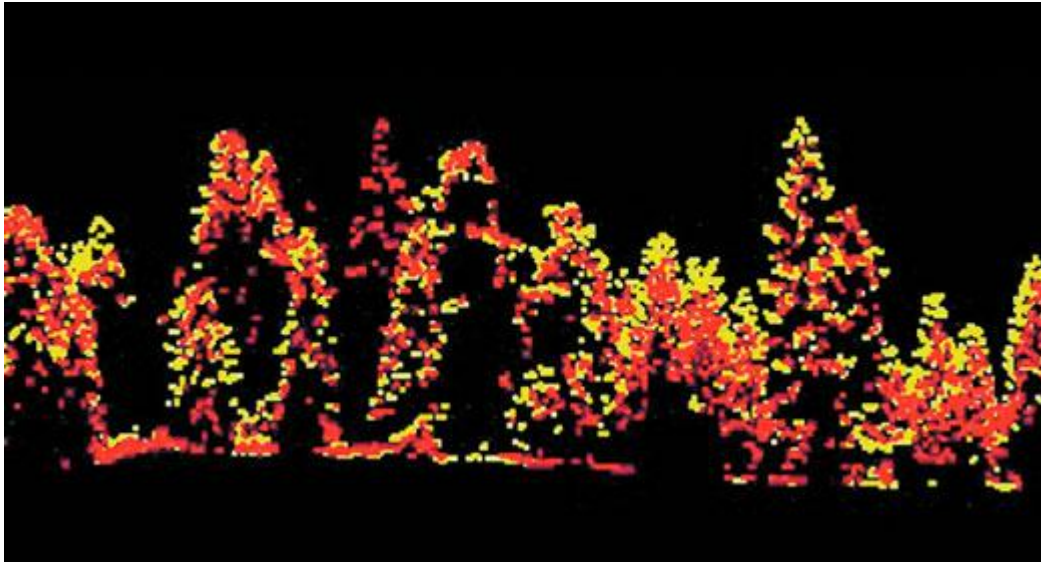
maanpinnan muotojen aiheuttamien virhetulkintojen syntymistä. (Monilähteinen VMI 2010.)

### 3.2 Lentokonekeilaus

Lentokonekeilauksessa käytetään hyväksi laserkeilausdataa ja GPS-koordinaatteja. Lentokoneessa oleva mittalaite lähettää lasersäteen, jonka kulku-aika mitataan. Kun tiedetään lasersäteen lähtökulma suhteessa mittalaitteeseen, saadaan jokaiselle mittapisteelle laskettua koordinaatti. GPS-paikannuksen avulla voidaan määrittää koneen sijainti, joten mittauspisteet voidaan sovittaa karttakoordinaatistoon. Koordinaattien lisäksi mittalaite määrittää myös jokaisen pisteen intensiteetti-arvon paluusignaalin voimakkuuden perusteella. (Joala 2006.)

Metsänmittaustarkoituksessa laserkeilausta tehtäessä tarkkuus on enimmillään 4 mittapistettä neliömetrillä. Tällä tavalla saadaan tietoa metsistä yhden puun tarkkuudella. Puiden ominaisuuksista voidaan määrittää esimerkiksi puun pituus ja latvuksen koko. Yleisempi mittaustarkkuus on kuitenkin 0,8 osumaa neliömetrille, jolloin tietoa ei saada yksittäisen puun tarkkuudella. Mittauksilla voidaan myös määrittää puuston pituuskasvua, jos mittaus tehdään muutaman vuoden välein.

Kuviossa 1 näkyy metsän viiden vuoden kasvu. Punaiset pisteet ovat ensin mitattuja ja keltaiset pisteet viisi vuotta ensimmäisen jälkeen. Kuvasta näkyy myös kaatunut puu, jonka kohdalla ei näy kuin punaisia pisteitä. Pisteiden avulla voidaan määrittää metsän vuotuinen kasvu.



**KUVIO 1. Laserkeilauspisteet (Pihlava 2008)**

Laserkeilaus on perinteiseen maastomittausmenetelmään verrattuna tarkempi. Laserkeilauksen mittavirhe voi jäädä alle 15 prosenttiin, kun se maastomittausmenetelmässä voi olla jopa 30 prosenttia. Laserkeilauksen heikkouksiin kuitenkin kuuluu se, että pienet puut voivat jäädä isojen puiden taakse piiloon. Laserkeilauksella ei myöskään saada mitattua taimikkoa tai nuorta metsää tarkasti. (Pihlava 2008.)

### 3.3 Metsänmittauksen mobiilisovellukset

Perinteisten metsänmittausmenetelmien kuten relaskoopin rinnalle on tullut erilaisia sähköisiä järjestelmiä. Näistä yhtenä esimerkkinä ovat maastotietokoneet, jotka ovat taulutietokoneen tapaisia, metsänmittaukseen tarkoitettuja laitteita.

Maastotietokoneet voivat kuitenkin olla todella kalliita, jopa monen tuhannen euron hintaisia laitteita. Maastotietokoneiden rinnalle on tullut myös älypuhelinsovelluksia, joiden hankintahinta ja käyttökustannukset ovat murto-osa maastotietokoneiden hinnasta.

Yksinkertaisimpia sovelluksia ovat esimerkiksi japanilaisen Taakuum Watakushin Applen iOS käyttöjärjestelmälle kehittämät ilmaiset iHypsometer ja iBitterlich sovellukset. Ensin mainitulla sovelluksella mitataan puiden pituutta hyödyntäen tiedetyn korkeuksista referenssimittaa. Puhelin laskee puun pituuden puhelimen kiihtyvyyssanturia käyttäen, jolla saadaan puhelimen kulma suhteessa referenssimittaan ja puuhun. Jälkimmäinen sovellus on periaatteessa elektroninen

relaskooppi. Ohjelmalla mitataan kameraa apuna käyttäen puita, joiden läpimitta sattuu puhelimen näytöllä olevan asteikon sisään. Kun puhelimen kanssa on pyörähdetty 360 astetta ja mitattu kaikki puut sekä lisätty mediaanipuun pituus, sovellus laskee puuston pohjapinta-alan sekä puuston tilavuuden. (Hemery 2012.)

VTT on kehittänyt Androidille Relasphone-nimisen sovelluksen, joka on samanlainen iBitterlich-sovelluksen kanssa, eli toimii relaskoopin tavoin. Sovellus on betavaiheessa ja ainakin vielä ilmainen. Sovelluksella voi mitata puun läpimittaa, laskea puuston runkotilavuuksia ja arvioida puuston hintaa. Käyttäjä laskee sovelluksella mittauspaikalta puulajien runkomäärät ja syöttää kuvion puiden keskimääräisen pituuden. Puiden pituuden arvioimiseen sovelluksessa on kulmamittaukseen perustuva korkeusmittaustyökalu. Tiedoista lasketaan automaattisesti puulajikohtaiset pohjapinta-alat ja puuston runkotilavuudet. Mittauspaikan GPS-koordinaatit tallennetaan myös tietoihin, mikä helpottaa tietojen täydentämistä jälkeinpäin. (Relasphone 2014.)

Kehittyneempää tekniikkaa edustaa Trestima Oy:n Trestima mobiilisovellus. Sen avulla otetaan metsästä kuvia, jotka sitten lähetetään eteenpäin Trestiman pilvipalveluun analysoitavaksi. Pilvipalvelussa kuvista analysoidaan runkojen pituudet, läpimitta ja puulajit. Sovellus antaa käyttäjälle jatkuvasti tietoa mittauksista. Käyttäjä voi päätellä tietojen avulla, milloin mittaus on tarpeeksi tarkka ja tarvittaessa pienentää mittauksen keskivirhettä tekemällä jatkomittauksia. (Trestima Oy 2013.)

## 4 WINDOWS PHONE 8

### 4.1 Yleistä

Windows Phone 8-mobiilikäyttöjärjestelmä on Microsoftin 20. kesäkuuta 2012 julkaisema Windows Phone 7 -mobiilikäyttöjärjestelmän seuraaja. Windows Phone 8 (WP8) -käyttöjärjestelmä perustuu samaan Windows NT -yttimeen kuin työpöytäpuolen Windows 8, toisin kuin Windows Phone 7 (WP7), joka pohjautui vanhaan Windows CE -yttimeen. (Windows Phone 8 2014.)

WP8:n tärkeimmät ominaisuudet verrattuna vanhaan WP7:ään ovat tuki isommille näytöille ja useampiytimisille prosessoreille, todellinen moniajo sekä tuki NFC:lle. WP8:ssa on myös uudempi Internet Explorer 10 -selain. WP8:ssa toimii myös vanhan WP7:n sovellukset, mutta WP7-puhelimiin ei voi päivittää WP8-käyttöjärjestelmää. Microsoft on luvannut WP8:lle tukea 36 kuukautta. (Mt.)

## 4.2 Puhelimet

Windows Phone -puhelimia valmistaa pääasiassa Nokia, jonka puhelintoiminnot Microsoft ilmoitti 3. lokakuuta 2013 ostavansa (Microsoft to acquire Nokia's devices & services business, license Nokia's patents and mapping services 2013). Nokian osuus Windows Phone -markkinoista marraskuussa 2013 oli 90 % lopun osan jäädessä jaettavaksi HTC:n (7 %), Samsungin (1,8 %) ja Huaweiin (1,3 %) kanssa. (Nokia now controls 90 % of the Windows Phone 8 market, with the low-end Lumia 520 grabbing over 35 % share 2013.) Windows Phone 7 -version osuus kaikista windows phoneista on noin neljännes, loppujen ollessa Windows Phone 8 -puhelimia.

Älypuhelinkäyttöjärjestelmistä Windows Phone on 3. yleisin käyttöjärjestelmä 3,6 prosentilla markkinaosuudellaan. Markkinoita dominoi Android 74,9 prosentilla osuudella Applen ollessa kakkonen 12,9 prosentilla osuudella. Windows Phonen osuus on kuitenkin kasvanut 1,6 prosenttiyksikköä vuoden aikana, mikä tarkoittaa 80 prosentilla kasvua vuodessa. (Android on 81 percent of new smartphones, Windows Phone share grows: IDC Q3 2013.)

## 4.3 Windows Phone 8 sovelluskehitys

### 4.3.1 Visual Studio

Windows Phone 8:lle kehitetään sovelluksia Visual Studion 2012:lla tai uudemmalla versiolla, ja siihen tarvitsee olla asennettuna Windows Phone 8 SDK. Ohjelmien kehityksessä käytetään XAML-ohjelmointikieltä käyttöliittymän kehittämiseen ja C#- tai Visual Basic kieltä koodin kirjoittamiseen. Grafiikkapiiristä kaiken hyödyn irtisaamiseksi on mahdollista kehittää ohjelmia myös C++-kielellä.

Ohjelmien testaamiseksi puhelimessa on puhelin ensin rekisteröitävä (ks. kuvio 2). Rekisteröinti tapahtuu erillisellä työkalulla, joka toimitetaan Windows Phone 8 SDK:n yhteydessä. Kun puhelin on rekisteröity, voi Visual Studiosta ladata kehitettävät ohjelmat puhelimeen testaamista varten. (Getting started with developing for Windows Phone 2014.)



**KUVIO 2. Puhelimen rekisteröinti**

#### 4.3.2 C#

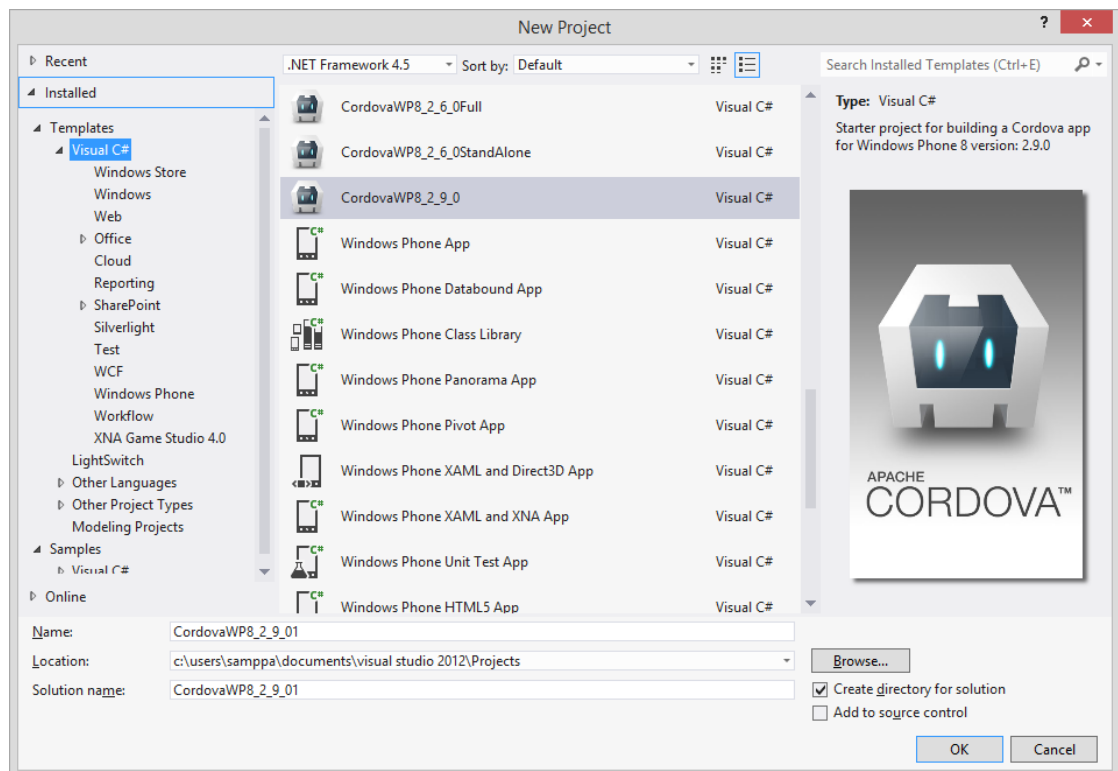
C-sharp tai lyhyemmin C# on korkean tason ohjelmointikieli, jonka Microsoft on kehittänyt käytettäväksi sovellusten kehittämiseen Visual Studiolla ja .NET Frameworkilla. C# on kehitetty siten, että se olisi mahdollisimman yksinkertainen, tehokas ja helppokäyttöinen. C# on oliopohjainen ohjelmointikieli, joka yhdistää C++-kielen tehokkuuden ja Java-kielen helppouden. (C Sharp (programming language) 2014.)

#### 4.3.3 HTML5

HTML5 on viides versio HTML-merkintäkielestä. Se on ensisijaisesti tarkoitettu internetsivujen luomiseen. HTML5 on kehitetty niin, että nettisivuilla voidaan esittää miltei mitä vaan ilman, että tarvitsee ylimääräisiä liitännäisiä. HTML5:llä voidaan esittää animaatioita, musiikkia, videoita sekä monimutkaisia selaimessa toimivia sovelluksia. (HTML5 Introduction n.d.)

#### 4.3.4 PhoneGap

Windows Phone 8:lle sovelluksia PhoneGapilla kehitettäessä pitää Visual Studioon asentaa mallipohja, "*template*", jonka pohjalta Visual Studio osaa luoda projektin, jossa on kaikki tarvittavat tiedostot PhoneGap-sovellukselle. Kun mallipohja on asennettu oikeaan kansioon, voidaan uusi PhoneGap-projekti luoda suoraan New Project-valikosta (ks. kuvio 3). (Windows Phone 8 Platform Guide n.d.)

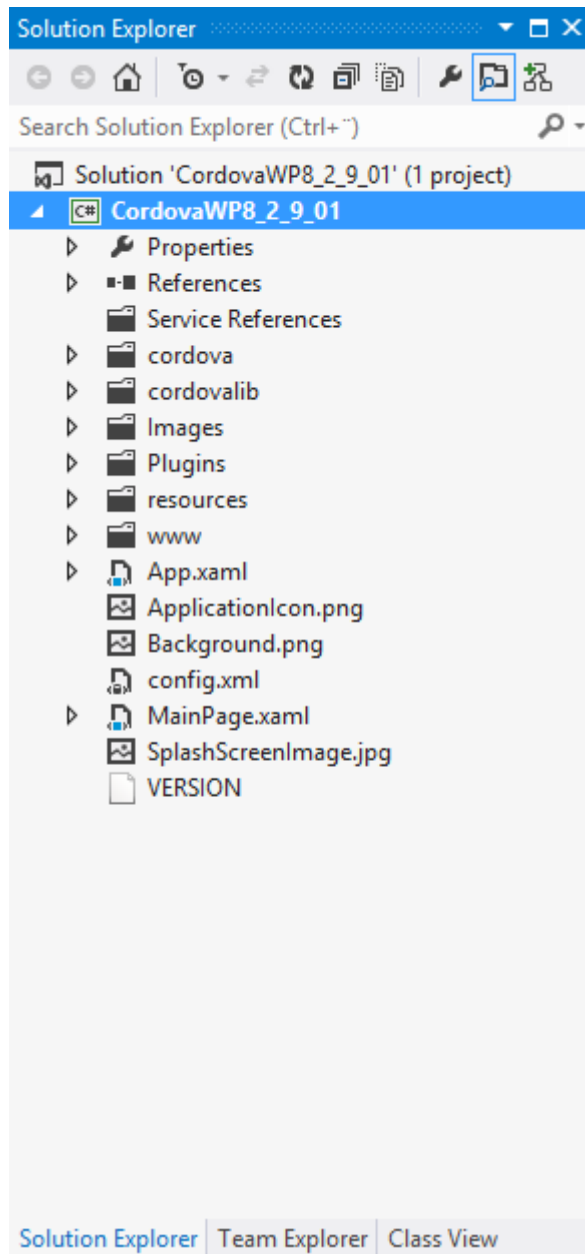


#### KUVIO 3. Visual Studio-projektin luonti

Kun uusi projekti on luotu, löytyy kaikki PhoneGap-sovelluksen tarvitsemat tiedostot projektikansiosta (ks. kuvio 4). HTML5-osio luodaan WWW-kansioon ja mahdolliset pluginit, jotka on ohjelmoitava natiivisti, tehdään Plugins-kansioon. On hyvä huomioida, että web-sovelluksessa käytettävät kuvat kannattaa sijoittaa WWW-kansion sisään eikä projektin juuresta löytyvään Images-kansioon. Kuviin viittaaminen tulee näin olemaan helpompaa.

WWW-kansiosta löytyvä cordova.js tiedosto on JavaScript-tiedosto, joka liittää web-sovelluksen ja PhoneGap-kirjaston yhteen. PhoneGapin ominaisuuksia voi käyttää, kun cordova.js on liitetty web-sovellukseen. Mahdollisten erillisten pluginien käyttämät JavaScript-tiedostot on myös lisättävä web-sovellukseen, jotta pluginin ominaisuuksia voi käyttää.





**KUVIO 4. Visual Studio Solution Explorer**

## 5 TYÖN TOTEUTUS

### 5.1 Bluetooth plugin

Yksi laajimmista osa-alueista ohjelman kääntämisessä Windows Phonelle oli Bluetooth pluginin tekeminen. Plugin on tehtävä jokaiselle alustalle erikseen alustan natiivilla ohjelmointikielellä, joka Windows Phonen tapauksessa oli C#. Pluginin kirjoittamista helpotti suuresti se, että Android versioon oli jo aiemmin tehty kyseinen plugin. Johtuen Java- ja C#-kielen yhteneväisyyksistä tämä helpotti urakkaa huomattavasti, vaikka alustojen arkkitehtuurit poikkesivatkin suuresti toisistaan.

Lähtökohtana Bluetooth pluginin teolle oli saada mahdollisimman moni ominaisuus toimimaan alustan rajoitteiden puitteissa. Windows Phonen Bluetooth API ei esimerkiksi salli Bluetoothin päälle ja pois laittamista ohjelmallisesti. Pluginista ei tullut aivan yhtä kattavaa kuin Androidin vastaavasta, mutta toimimaan saatiin kaikki ohjelman kannalta tarvittavat ominaisuudet.

PhoneGap plugin vaatii sekä JavaScript- että natiivin C#-toteutuksen. JavaScript toteutus oli tehty jo Android sovelluksen yhteydessä, joten ainoaksi toteutettavaksi osaksi jäi C# osuus. Bluetooth plugin periytyy PhoneGapin BaseClass-luokasta, jolloin metodeja voi kutsua JavaScript- rajapinnan kautta. Seuraavassa on esitetty, kuinka Bluetooth plugin JavaScript-osuus tarkistaa, onko Bluetooth päällä:

```
var exec = require('cordova/exec');

Bluetooth.prototype.isEnabled = function(onSuccess, onError)
{
    exec(onSuccess, onError, "Bluetooth", "isEnabled", []);
}
```

Vastaavan toiminnallisuuden toteutus natiivisti C#-ohjelmointikielellä:

```
namespace Cordova.Extension.Commands
{
    class Bluetooth : BaseCommand
    {
        public void isEnabled(string options)
        {
            isEnabledAsync(CurrentCommandCallbackId);
        }

        public async void isEnabledAsync(string callbackId)
        {
            try
            {
```



tiedot palautuvat skriptinä, eivätkä JSON-formaatissa. Näin voidaan kiertää selaimen "*cross-domain*"-esto.

### 5.3 Mittausmenetelmät

JokaMies-sovelluksessa on metsän tulkinta jaettu kahteen erilliseen toiminnallisuuteen. Puustotulkinnalla selvitetään yli 6 senttimetriä halkaisijaltaan olevien puiden määrä, ja se keskittyy enemmän metsän inventointiin.

Alikasvoslaskennassa lasketaan alikasvosta, alle 6 senttimetriä halkaisijaltaan olevia puita. Alikasvoslaskennalla yritetään saada selville metsän työvaikeusluokitusta.

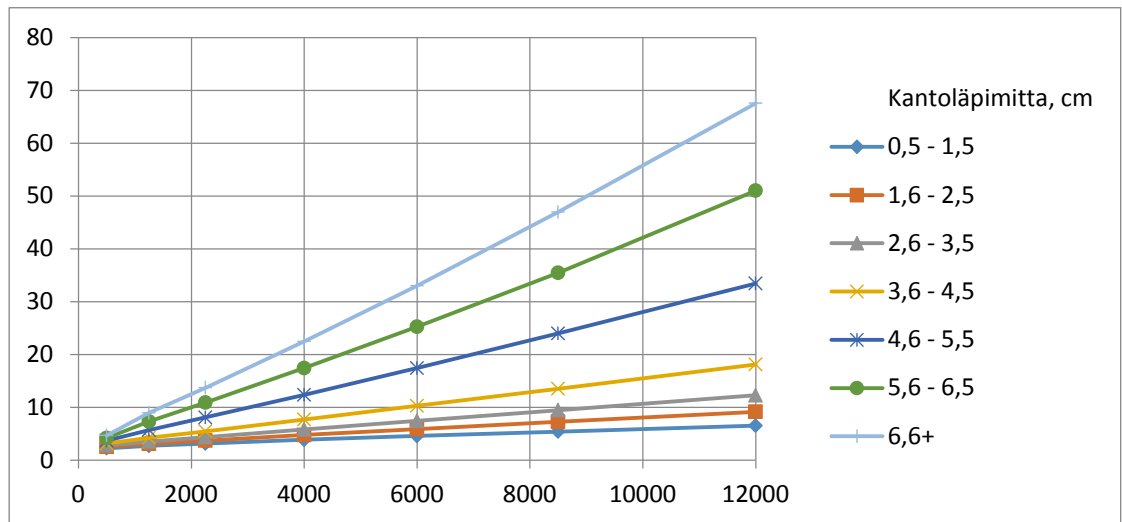
#### 5.3.1 Puustomittaus

Puustomittauksessa mitataan tietty osa kuvion yli 6 senttimetriä halkaisijaltaan olevista puista. Sovellus yleistää mittauksista saadun tiedon koko kuviolle, ja tätä kautta saadaan arvioitu puumäärä ja lajijakauma koko kuviolle. Sovelluksella ei voi mitata puiden korkeuksia, mikä tarvittaisiin tarkkaan kuutiomäärän arviointiin.

#### 5.3.2 Alikasvoslaskenta

Alikasvoslaskentaominaisuudella mittaaja määrittää, kuinka paljon kuviolla on alle 6 senttimetriä halkaisijaltaan olevia puita, risuja. Raivaustyön vaikeuden määrittelyssä risujen määrä on suurin vaikuttava tekijä. Mittaaja mittaa usealta koealalta silmämääräisesti, kuinka paljon koealalla on risuja, ja mikä niiden keskihalkaisija on. Tarpeeksi monen koealan jälkeen mittaaja saa tulokseksi kyseisen kuvion työvaikeusluokituksen, jota mittaaja voi käyttää hyväksi raivaustyön hinnan määrittämisessä.

Alikasvosken raivaukseen kuluva aika voidaan laskea sen mukaan, mikä on keskimääräisen raivattavan puun kantoläpimitta ja kuinka paljon puita on hehtaarilla. Kuviossa 5 on esitetty raivaukseen kuluva aika. Vaaka-akselilla on raivattavien puiden määrä hehtaarilla ja pystyakselilla raivaukseen kuluva aika tunteina.



KUVIO 5. Puuston raivauksen kuluva aika tuntia/ha

Raivaukseen kuluvan ajan määrän voi laskea kaavasta

$$\frac{(l^2 + 2)}{10000} + p + 2$$

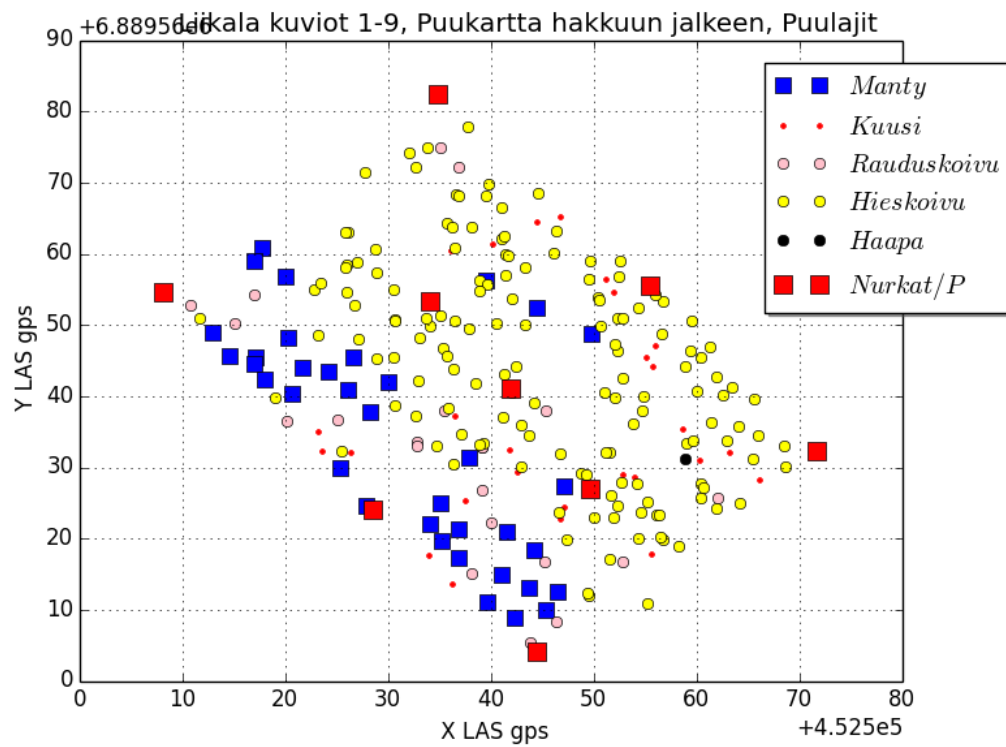
, jossa p on raivattavien puiden lukumäärä ja l on raivattavien puiden keskimääräinen läpimitta. Kaava on määritelty kuvaajan avulla ja on estimaatti. Kaavaa voi hyödyntää sovelluksen taustajärjestelmässä, joka laskee aluskasvuston raivaukseen kuluvaa aikaa.

### 5.3.3 Mittaustulokset

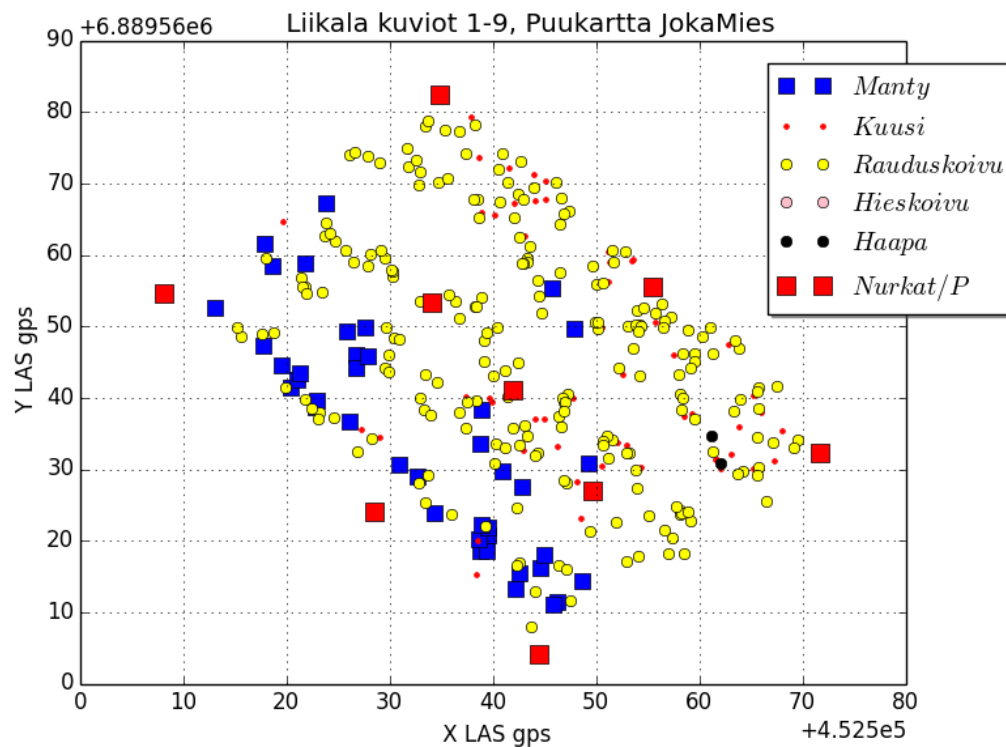
Yksi kattavimmista koemittauksista suoritettiin Toivakan kunnan Liikalan kylässä, jossa noin 0,25 hehtaarin kuviolta mitattiin kaikki puut (ks. kuvio 7). Kuvion rajat on esitetty liitteen 2 kuvassa. Kyseisen kuvion erottaa sen sisällä olevat siniset paikannus ympyrät. Kuviosta on myös laserkeilausdataa (ks. liite 3), mutta keilausdataa ei tässä tössä ole käsitelty. Mittaustuloksia verrattiin Masser Sonar Caliperilla aiemmin mitattuihin tuloksiin (ks. kuvio 6). Mittaustuloksista käy ilmi, että JokaMies mittauksissa on tapahtunut puiden osalta selvä siirtymä koilliseen. Siirtymä on vaikuttanut kaikkiin mittauksiin, joten sen vaikutus tuloksiin on pienehkö.

Puiden välimatkat ovat myös supistuneet aikaisempiin mittauksiin nähden. Pienten välimatkojen mittaaminen GPS:n avulla on hankalaa. Sitä voisi kuitenkin parantaa hyvällä suunnittelulla. Mitattavien puiden valitseminen suoralta linjalta toimisi paremmin kuin aina lähimmän puun mittaaminen. Näin vältettäisiin mittauksien

meneminen samaan kasaan, koska etäisyydet kasvavat, ja näin ollen lopputulos vastaisi myös enemmän todellisuutta.



KUVIO 6. Mittaustulos Masser Sonar Caliper:lla



KUVIO 7. Mittaustulos JokaMies-mobiilisovelluksella

Liikalassa mitattiin myös puiden välisiä etäisyyksiä laseretäisyysmittarilla. Puiden välisiä etäisyyksiä verrattiin JokaMiehestä saatuihin tuloksiin. Taulukossa 1 on JokaMiehen antamien koordinaattien perusteella lasketut etäisyydet, laseretäisyysmittarilla mitatut todelliset etäisyydet sekä kuinka paljon JokaMiehellä mitatut etäisyydet eroavat todellisista etäisyyksistä.

**TAULUKKO 1. Puiden väliset etäisyydet**

JokaMies	Todellinen	Ero
5,40 m	3,70 m	1,70 m
1,30 m	1,70 m	0,40 m
0,49 m	2,45 m	1,96 m
2,59 m	1,99 m	0,60 m
1,94 m	3,22 m	1,28 m
3,69 m	4,01 m	0,32 m
2,09 m	3,06 m	0,97 m
1,47 m	1,70 m	0,23 m
0,47 m	1,74 m	1,27 m
4,16 m	2,01 m	2,15 m
3,26 m	1,95 m	1,31 m
0,64 m	2,37 m	1,73 m
2,16 m	2,04 m	0,12 m
0,96 m	4,02 m	3,06 m
0,37 m	1,53 m	1,16 m
2,10 m	1,37 m	0,73 m

Taulukon 1 mittausten keskivirheeksi muodostui 1,187 metriä. Virhemarginaali on 0,12 metristä 3,06 metriin. Puhelin ilmoittaa GPS virheeksi 3 - 4 metriä, mikä on hyvin realistinen virhemarginaali. Erot eivät kuitenkaan aivan kumoa toisiaan, vaan etäisyyksien yhteen lasketuissa summissa on noin 5 metrin heitto.

Mittauksia tehtiin kahdessa eri paikassa ja toisen mittauksen (ks. taulukko 2) keskivirheeksi muodostui 1,082 metriä. Toisessa mittauksessa virheet olivat pääosin pienempiä, mutta muutama isompi virhe korottaa keskivirhettä reilusti.

**TAULUKKO 2. Puiden väliset etäisyydet**

JokaMies	Todellinen	Ero
4,32 m	2,43 m	1,89 m
0,73 m	1,27 m	0,54 m
3,06 m	2,44 m	0,62 m
0,71 m	1,01 m	0,30 m
1,86 m	2,80 m	0,94 m
1,45 m	4,03 m	2,58 m
2,05 m	1,69 m	0,36 m
2,33 m	3,19 m	0,86 m
0,39 m	2,30 m	1,91 m
3,74 m	3,65 m	0,09 m
2,21 m	1,50 m	0,71 m
3,87 m	3,40 m	0,47 m
5,33 m	2,54 m	2,79 m

Puiden etäisyyksien epätarkkuuteen vaikuttaa myös mittaustapa. Mitattaessa puuta Liikalassa puhelin oli takin hihataskussa, joten mitatut puiden sijainnit eivät ole aivan puiden todellisten keskipisteiden kohdalla. Mittaustarkkuutta voi parantaa viemällä puhelimen puuhun kiinni ja aina samalle puolelle puuta.

Mittaustarkkuuteen vaikuttavat kuitenkin monet seikat. Etäisyyksien ollessa lyhyet mitattavat puut näyttäisivät menevän yhteen nippuun. Ongelmaa voi korjata käymällä puhelimen kanssa kauempana, vaikka mitattava kohde olisikin lähellä. Häiriöitä voi aiheuttaa myös tiheä metsikkö. Paksut puiden oksat saattavat häiritä GPS-signaalia huonontaan tai vääristäen mittatarkkuutta.



## 5.4 Käyttöliittymä

### 5.4.1 Alkuperäinen käyttöliittymä

Ensimmäinen käyttöliittymä (ks. kuvio 8) oli todella yksinkertainen. Kaikki toiminnot löytyivät aloitussivulta, joten siitä syntyi todella pitkä näkymä. Aloitussivu listasi kaikki kuviot, Bluetooth-laitteet ja muut myöhemmin lisätyt ominaisuudet, kuten mittausten lisääminen manuaalisesti.

Huonona puolena tässä oli, että kaikki tiedot kuvioista ja Bluetooth-laitteista piti ladata sovellukseen kerralla. Tämä teki sovelluksen käynnistämisestä erittäin hidasta. Tosin tässä vaiheessa kehitystyötä käyttöliittymään ei ollut paljon aikaa uhrattukaan, vaan keskityttiin enemmän ohjelman teknisen puolen toimintaan.

### 5.4.2 Graafinen käyttöliittymä

Seuraavan kehitysversion (ks. kuvio 9) käyttöliittymä sisälsi aloitusnäkyvän, jossa oli jokaiselle toiminnallisuudelle oma ikoni. Tämä mahdollisti siistimmän, kokonaan ruudulle mahtuvan käyttöliittymän.

Ohjelman käynnistyminen myös nopeutui, koska tiedot voitiin ladata vasta, kun niitä tarvittiin.

Sovellukseen kehitettiin tässä vaiheessa myös logo, jota ei ensimmäisessä versiossa ollut. Sovellukseen lisättiin myös asetukset valikko, josta pystyy muuttamaan muun muassa karttapalvelimen ja metsätietopalvelimen osoitetta.



KUVIO 8. Ensimmäinen käyttöliittymä



KUVIO 9. Käyttöliittymä toisessa vaiheessa

#### 5.4.3 Responsiivisuus

Kolmannessa kehitysversiossa (ks. kuvio 10) käyttöliittymässä panostettiin responsiivisuuteen. Aiemmin sovellus oli lukittu pystyasentoon, eikä se skaalautunut vaaka-asentoon. Nyt sovellusta kuitenkin pystyy käyttämään puhelimen ollessa kummassa asennossa tahansa.

Ikoneihin lisättiin myös tekstit, jotta käyttäjä tietäisi paremmin, minkä toiminnan takana on mitään ominaisuuksia. Tässä vaiheessa sovelluksessa erotettiin myös puustomittaus ja työvaikeusluokitus.

Sovelluksen pohjana toimiva jQuery Mobile vaihdettiin Bootstrap kirjastoon, koska sovelluksesta haluttiin versio, joka toimii myös tietokoneen selaimella. Bootstrap kirjasto mahdollisti myös paremman responsiivisuuden kuin jQuery Mobile.



**KUVIO 10.** Käyttöliittymän kolmas versio selaimessa

## 5.5 C# versio

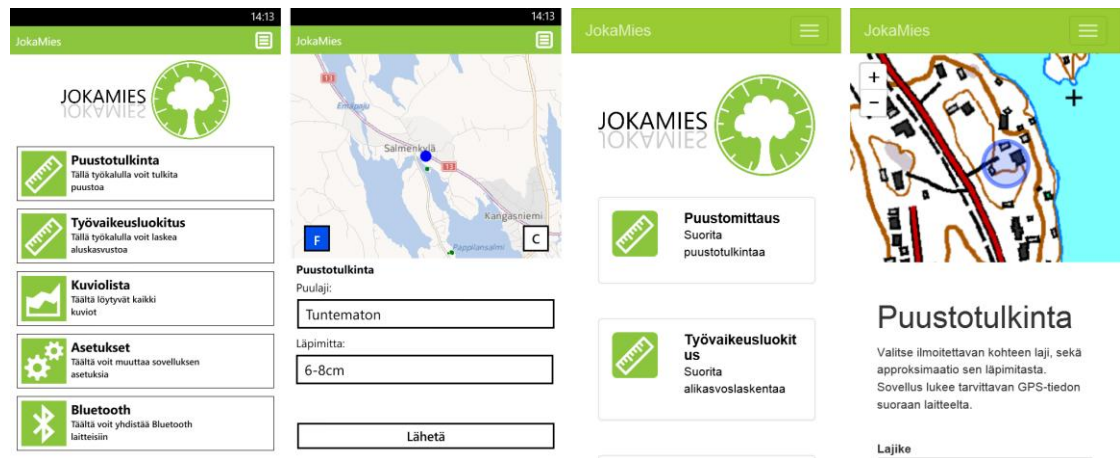
### 5.5.1 Yleistä

HTML5 version lisäksi JokaMies sovelluksesta tehtiin myös C# versio, joka on Windows Phonen natiivi ohjelmointikieli. Sovelluksen tarkoituksena oli korjata HTML version sudenkuoppia ja verrata, miten ohjelmointikieli vaikuttaa sovelluksen toimintaan.

Suurin ongelma Windows Phonella käytettäessä HTML5 versiota JokaMies sovelluksesta oli karttojen toimivuus. Kartat latautuivat hitaasti ja ne aiheuttivat myös ohjelman hidastumista, luultavasti muistivuodon takia. Tarpeeksi pitkään sovellusta käytettäessä sovellus muuttui niin hitaaksi, että se käytännössä lakkasi toimimasta.

### 5.5.2 C# version perusta

C# versiosta koitettiin tehdä käyttöliittymän osalta mahdollisimman samanlainen (ks. kuvio 11) kuin alkuperäinenkin, jotta vertailu olisi mahdollisimman helppoa.



**KUVIO 11. Sovelluksien erot. Vasemmalta oikealle: C# pääsivu, C# puustotulkinta sivu, HTML5 pääsivu ja HTML5 puustotulkinta sivu.**

Teknisessä toteutuksessa tuli kuitenkin joitain eroavaisuuksia, jotka vaikuttivat myös hieman sovelluksien vertailuun. C# versiosta ei kuitenkaan tehty aivan yhtä laajaa kuin HTML5 sovelluksesta, vaan siitä puuttui joitain ominaisuuksia. Muun muassa Bluetooth yhteyttä ei ole testattu toimivaksi ja asetuksia ei voi muuttaa.

C# versio toteutettiin tekemällä samat sivut, tai näkymät, kuin HTML5 versiossa. Näkymät koostuvat XAML ja C# tiedostoista. XAML tiedostoissa on käyttöliittymä elementit ja C# tiedostoissa on toiminnallinen toteutus.

### 5.5.3 Tiedonsiirto

C# versiossa tiedonsiirto hoidettiin RestSharp-kirjastolla, joka on .NET arkkitehtuurille kehitetty REST ja HTTP rajapinta. RestSharpilla haetaan serveriltä ohjelman tarvitsemat tiedot JSON muodossa. JSON:ia käsitellään ohjelmassa James Newton-Kingin kehittämällä Json.NET-kirjastolla. Kirjasto on JSON ohjelmistokehitys .NET arkkitehtuurille.

Tietojen haku aloitetaan määrittämällä haettavan tiedon sisältävän serverin osoite. URL osoite annetaan uudelle RestClient muuttujalle. Tämän jälkeen määritellään tarkempi tiedon osoite, joka käytännössä on alikansio serverillä sekä määritellään metodi, jota käytetään. Palautettava tieto otetaan talteen muuttujaan, joka

parsitaan. Parsiminen tapahtuu Json.NET kirjaston työkalujen avulla. Tiedot muunnetaan JSON objektiksi, jonka jälkeen tietoja voidaan käsitellä objekti tasolla.

Tietojen haku on asynkrooninen prosessi, joten tarvittavasta virheidenkäsittelystä on hyvä huolehtia. Käytännössä tämä tarkoittaa koodin lisäämistä try-catch lohkon sisään. Seuraavassa esimerkki, miten ohjelmassa on haettu serveriltä tiedot kuviosta, "stand:sta":

```
string url = "http://mehta.labranet.jamk.fi/api/jokamies/";
var client = new RestClient(url);
var request = new RestRequest("stands/", Method.GET);
string content = "";

client.ExecuteAsync(request, response =>
{
    try
    {
        content = response.Content;
        JObject result = JObject.Parse(content);
        foreach (var o in result["objects"])
        {
            // Manage objects
        }
    }
    catch (Exception e)
    {
        // Handle the error
    }
});
```

#### 5.5.4 Kartat ja navigointi

Yksi HTML5 version murheenkryyneistä oli Windows Phonella kartat. C# versiossa karttarajapintana toimi Microsoftin oma Maps API, joka käyttää Microsoftin omia karttoja. Kartat poikkeavat hieman HTML5 version käyttämisestä maanmittauslaitoksen kartoista. HTML5 versiosta poiketen C# versiossa voi "karttamoodeja" vaihtaa halutessaan tiekarttojen, maastokarttojen ja ilmakuvien välillä.

Kartta määritetään ohjelmaan lisäämällä XAML tiedostoon <maps:Map> tagi, joka automaattisesti luo kartan. Kartan ulkoasua voi muuttaa kirjoittamalla parametrit Map tagiin tai muokkaamalla karttaa Visual Studion graafisella käyttöliittymällä, joka luo koodin automaattisesti. Seuraavassa on esimerkki, kuinka kartta on luotu JokaMies sovelluksessa:

```
<maps:Map x:Name="measurementMap" CartographicMode="Road" Center="65.50, 25.00"
ZoomLevel="4.2" HorizontalAlignment="Left" Grid.Row="1" VerticalAlignment="Top"
Width="480" Height="360"/>
```

Kartta toimii sellaisenaan pelkällä XAML toteutuksella, mutta jotta kartta saadaan näyttämään käyttäjän sijainti, kuvioiden rajat ja mittaustiedot, tarvitsee kartalle tehdä C# toteutus.

Käyttäjän sijainnin määrittämiseksi tarvitsee laitteen navigointipalvelut ottaa käyttöön. Navigoinnin käyttöönottamiseksi tarvitsee laitteen GPS aktivoida. Windows Phone 8 sovelluksissa tämä tapahtuu GeoCoordinateWatcher luokan avulla. GeoCoordinateWatcher alustetaan ja annetaan sille alustettaessa navigoinnin tarkkuus. Tämän jälkeen luodaan tapahtumankäsittelijät GPS:n tilan muutokselle sekä sijainnin muutokselle. GeoCoordinateWatcherille on myös hyvä antaa kynnyсарvo, minkä alle se ei laukaise sijainnin muutoksia tarkkailevaa tapahtumankäsittelijää. Seuraavassa on yksinkertaistettu esimerkki, kuinka GeoCoordinateWatcher alustetaan, luodaan sille tapahtumankäsittelijät sekä visualisoidaan käyttäjän sijainti kartalla:

```
GeoCoordinateWatcher watcher;

private void StartLocationWatcher() {
    if (watcher == null) {
        watcher = new GeoCoordinateWatcher(GeoPositionAccuracy.High); // Set
        // GPS accuracy to high
        watcher.MovementThreshold = 0.05; // use MovementThreshold to ignore
        // noise in the signal
        watcher.StatusChanged += new
            EventHandler<GeoPositionStatusChangedEventArgs>(watcher_StatusChang
            ed);
        watcher.PositionChanged += new
            EventHandler<GeoPositionChangedEventArgs<GeoCoordinate>>(watcher_Po
            sitionChanged);
    }
    watcher.Start();
}

void watcher_StatusChanged(object sender, GeoPositionStatusChangedEventArgs e)
{
    switch (e.Status)
    {
        // Handle status changes
        case GeoPositionStatus.Disabled:
            break;
        case GeoPositionStatus.Initializing:
            break;
        case GeoPositionStatus.NoData:
            break;
        case GeoPositionStatus.Ready:
            break;
    }
}

void watcher_PositionChanged(object sender,
    GeoPositionChangedEventArgs<GeoCoordinate> e) {
```

```

    ShowMyLocationOnTheMap(App.currentPosition);
}

public void ShowMyLocationOnTheMap(GeoCoordinate position) {
    // Draw circle to the users position
}

```

### 5.5.5 C# ja HTML5 sovelluksien erot

Sovelluksien eroja määriteltiin mittaamalla sovelluksen latausnopeuksia. Testeissä mitattiin, kuinka kauan menee navigoida sivulta toiselle ja verrattiin näitä aikoja keskenään. Sovelluksesta mitattiin myös ruudunpäivitysnopeus sivuilta, joilla on kartta, koska kartat ovat raskain osa sovellusta. Ruudunpäivitysnopeus mitattiin puhelimen omalla sovelluskehitysoäkalulla.

Taulukosta 3 näkee, että C#-versio on kaikilla osa-alueilla paljon nopeampi kuin HTML5 sovellus pois lukien sovelluksen käynnistäminen. Erot selittyvät osin sillä, että C#-versio lataa tiedot palvelimelta heti sovelluksen käynnistäessä, kun taas HTML5 versio lataa seuraavalla sivulla tarvittavat tiedot sivun vaihtuessa.

**TAULUKKO 3. Sovelluksen latausajat**

Toiminto	HTML5	C#	Ero	Ero %
Sovelluksen käynnistys	3,4s	7,4s	- 4s	-118 %
Pääsivu -> Puustomittaus	5,2s	3,0s	2,2s	73 %
Pääsivu -> Työvaikeusluokitus	2,0s	1,1s	0,9s	82 %
Pääsivu -> Tulokset	3,0s	0,6s	2,4s	400 %
Tulokset -> Kuvio	5,0s	1,2s	3,8s	317 %

Sovelluksen käytettävyyteen vaikuttaa suuresti, kuinka sujuvasti sovellus toimii.

Sovelluksen sujuvuutta testattiin mittaamalla ruudunpäivitysnopeutta.

Ruudunpäivitysnopeus mitattiin kolmelta sivulta (ks. taulukko 4). Sivut olivat puustomittaus ja kaksi kuviosivua, joista toisessa oli 15 mittaus ja toisessa 318 mittaus, mikä teki jälkimmäisestä sivusta erittäin raskaan.

**TAULUKKO 4. Sovelluksen ruudunpäivitysnopeudet eri sivuilla**

Sivu	HTML5	C#	Ero	Ero %
Puustomittaus	10 fps	6 fps	- 4 fps	-40 %
Kuvio, 15 mittausta	8 fps	46 fps	38 fps	475 %
Kuvio, 318 mittausta	3 fps	8 fps	5 fps	166 %

Testien perusteella C# versio toimii pääosin sujuvammin. Ainoa testi, missä HTML5 versio päihitti C# version, oli puustomittaussivu. Tämä johtui siitä, että C# versiossa puustomittaussivulla näkyy kaikki tehdyt mittaukset kuviosta riippumatta. Mittauksia oli testiä tehdessä noin 470 kappaletta. HTML5 versio taas näyttää puustomittaussivulla ainoastaan tietyllä etäisyydellä olevat mittaukset.

Kaiken kaikkiaan natiivi C#-versio tuntuisi toimivan nopeammin ja erityisesti vakaammin kuin HTML5-versio. Erityisesti kartat toimivat paremmin, eikä sovellus hidastele missään vaiheessa. Sovelluksen kirjoittaminen oli myös helpompaa C#:lle kuin HTML5:lle, johtuen pääosin paremmista sovelluskehitystyökaluista. HTML5 sovelluksen etuna kuitenkin todettakoon sen toimivuus eri alustoilla, mikä poistaa tarpeen kehittää sovellus erikseen joka alustalle.

## 5.6 Taustajärjestelmä

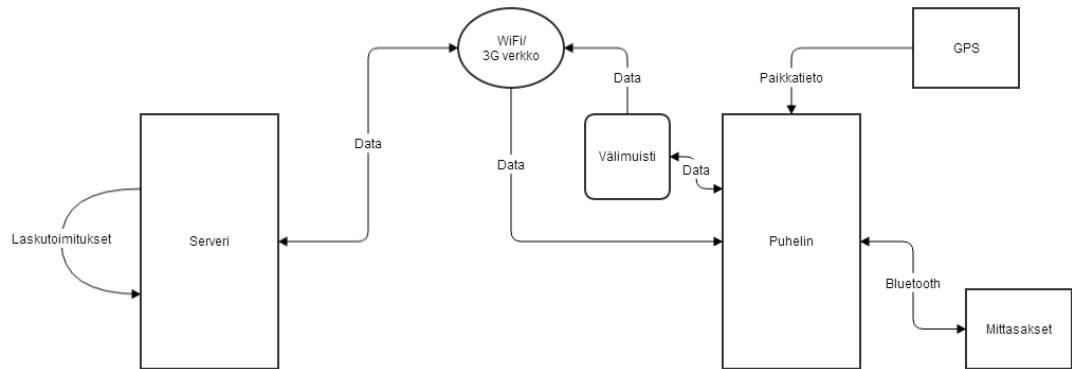
### 5.6.1 Yleistä

JokaMies-sovelluksen toiminta nojaa vahvasti taustajärjestelmänä toimivaan serveriin. Serverille tallennetaan kaikki kuvio ja mittaustiedot, josta ne haetaan puhelimelle. Serverillä tehdään myös raskaimmat laskelmat, jotta puhelin sovelluksesta ei tulisi tarpeettoman raskasta.



### 5.6.2 Serveri ja puhelin

Puhelimessa on oma välimuisti, jos serveriin ei saada yhteyttä. Välimuistissa säilyvät aiemmin ladatut kuviotiedot sekä mittaukset. Välimuistiin tallennetaan myös tehdyt mittaukset, jos niitä ei heti saada ladattua serverille. Kuviossa 12 on esitetty, kuinka sovelluksen data yhteydet toimivat.



**Kuvio 12. Puhelimen ja serverin välinen toiminta**

Mittauksen paikkatiedot ja käyttäjän sijainti tulevat puhelimeen GPS:ltä. Projektissa käytetty Nokian Lumia 620 tukee myös GLONASS-paikannusta, joka mahdollistaa paremman paikannuksen tarkkuuden. Mittasaksia käytettäessä mitatun puun tiedot tulevat Bluetooth-yhteyden kautta mittasaksilta puhelimeen.

### 5.6.3 Kuvion lisääminen serverille

Kuviot pitää lisätä serverille ennen kuin mittauksia pääsee aloittamaan. Kuviot lisätään serverille selaimessa toimivan käyttöliittymän avulla (kuvio 13). Kuviot voi lisätä tuomalla serverille kuviotiedostot tai kuviotiedostojen puuttuessa kuviorajat voi piirtää kartalle manuaalisesti selaimessa toimivan käyttöliittymän avulla.

# Latausohjeet

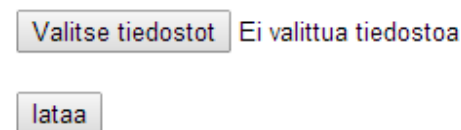
## Shapefile

Ladatessasi shx, shp ja dbf -tiedostot ovat pakollisia, jotta kuvioiden lataus on mahdollista

## Metsätietostandardi(XML)

Tällä hetkellä voi syöttää vain yhden XML-tiedoston kerrallaan

## Ladattavat tiedostot



Valitse tiedostot Ei valittua tiedostoa

lataa


### KUVIO 13. Kuviotietojen lataus tiedostoista

Jotta kuviotietojen vieminen tiedostoina onnistuu, täytyy käyttäjällä olla shx-, shp- ja dpf-päätteiset kuviotiedostot. Kuviot voi tuoda myös vaihtoehtoisesti metsätietostandardin mukaisina XML tiedostoina. Kuviotiedostot valitaan selaimen "Valitse tiedosto"-kohdasta. Kun kuviotiedostot on valittu, painetaan "lataa"-painiketta, jolloin kuvio luodaan serverille. Tämän jälkeen selain antaa ilmoituksen, onnistuiko kuvion luonti vai ei.

Manuaalisesti kuvion luonti onnistuu myös nettiselaimen kautta (ks. kuvio 14).

Kuvion rajat piirretään karttanäkymään, ja kun kuvio on valmis, annetaan kuviolle nimi ja tallennetaan se.

## Lisää stand

Shape: 

Name:

Owner:  +

Ground type:

Snow level:

Tags:   
A comma-separated list of tags.

**KUVIO 14. Kuvion lisääminen manuaalisesti****5.6.4 Serverin tekemät laskutoimitukset**

Serveri laskee mittausdatan perusteella kuusi eri tunnuslukua kuviolle. Tunnusluvut ovat kuvion pinta-ala, puupinta-ala, poistumatyyppi, poistuman tiheys, arvioitu runkoluku ja keskimääräinen puun halkaisija. Pinta-ala on ainut luku, joka ei ole arvio, vaan se lasketaan annettujen kuviorajojen mukaan. Muut luvut ovat arvioita.

Puupinta-ala lasketaan kertomalla arvioitu runkoluku ja arvioitu keskimääräinen puun halkaisija keskenään.

Poistumatyyppi määritellään sen mukaan, mitä lajia on eniten. Jos jotain lajia on enemmän kuin 70 % kaikista puista, se puulaji määrittelee, mikä on kuvion poistumatyyppi.

Poistuman tiheys ei ole ohjelmassa vielä käytössä, mutta sen pitäisi palauttaa arvo, joka kertoo, kuinka paljon puita pitäisi hehtaarilta kaataa, jotta puita olisi kasvun kannalta oikea määrä kuviolla.

Arvioitu runkoluku lasketaan mitattujen puiden välisten etäisyyksien avulla.

Mitattujen puiden etäisyydet pistetään taulukkoon ja järjestetään etäisyyksien mukaan. Taulukosta otetaan n-1 arvoa, jossa n on mitattujen puiden määrä. Arvoista otetaan keskiarvo ja keskiarvo korotetaan toiseen potenssiin. Näin saadaan pinta-ala,

jonka yksi puu vie tilaa kuviolta. Kuvion pinta-ala jaetaan puun tarvitsemalla pinta-alalla ja näin saadaan runkoluku kyseiselle kuviolle.

Keskimääräinen puun halkaisija lasketaan lisäämällä kaikkien mitattujen puiden läpimitat yhteen ja jakamalla tulos mitattujen puiden määrällä.

Serveri laskee myös kuviolle Runkolukusarjan, arvioidun keskivirheen ja lajittelee puut lajeittain. Runkolukusarja muodostetaan lajittelemalla mittaukset kahden sentin kokoisiin luokkiin. Arvioitu keskivirhe saadaan kuviossa 15 esitetystä kaavasta, jossa  $s$  on keskihajonta ja  $n$  on otoksen koko. Serveri laskee arvot jokaisen mittauksen jälkeen ja ne voi hakea serveriltä puhelimeen aina halutessaan. Arvoista voi seurata esimerkiksi keskivirheen käyttäytymistä ja saadun palautteen perusteella voi tehdä päätöksen, tarvitseeko mittausta jatkaa.

$$S = \frac{s}{\sqrt{n}}$$

**KUVIO 15. Keskivirheen kaava**

## 6 TYÖN TULOKSET

Lopputuloksena saatiin kehitettyä sovellus, joka toimii myös Windows Phone 8 alustalla. Sovellusta jouduttiin muokkaamaan Windows Phonea silmällä pitäen, jotta saatiin kaikki ominaisuudet toimimaan ilman suurempia ongelmia. Sovellus toimii täysin samalla tavalla kuin Android versio sovelluksesta, vaikkakin taustalla toimivassa koodissa jouduttiin tekemään Windows Phone kohtaisia muutoksia.

Yksi sovelluksen tavoitteista oli saada Bluetooth toimimaan Windows Phone 8 alustalla. Bluetooth saatiinkin toimimaan Windows Phonella, tosin paljon rajoitetummin kuin Androidin Bluetooth. Sovellusta kuitenkin muokattiin niin, että se käytti vain niitä Bluetoothin ominaisuuksia, mitkä toimivat myös Windows Phonella.

Sovelluksessa havaittiin myös ongelma, jota ei saatu projektin aikana korjattua. Sovelluksen kartta saa sovelluksen kaatumaan, jos sovellusta käyttää tarpeeksi pitkään. Ongelman ydin saattaa olla Microsoftin Internet Explorer selaimessa, jota Windows Phone käyttää, ja jonka päällä JokaMies sovellus toimii. Sovelluksen korjaaminen vaatisi perinpohjaisia muutoksia joko sovelluksen arkkitehtuuriin ja käytettäviin kirjastoihin tai korjauksia Internet Explorer selaimen.

Sovellusta myös testattiin käytännössä tekemällä mittauksia testi koealoilla. Mittausten tarkkuus osoittautui positiiviseksi yllätykseksi. Mittaukset olivat johdonmukaisia ja ne vastasivat hyvin muilla menetelmillä tehtyjä mittauksia.

Sovellukselle tehtiin myös käyttöohjeet (ks. liite 1) käyttöä helpottamaan. Käyttöohjeita voi jatkossa käyttää sellaisenaan ja pohjana laajemmille tuotteen jakelun yhteydessä jaettaville ohjeille.

Parannuksia voisi tehdä erityisesti mittausten perusteella laskettavissa tunnuslukujen algoritmeissa. Tunnuslukujen tarkkuutta voisi myös parantaa muuttamalla mittaustapa pinta-ala perusteiseksi. Sovelluksesta voisi myös tehdä vielä karsitumman version, joka toimisi ainoastaan kännykässä tarvitsematta taustalla toimivaa serveriä. Näin sovelluksen käyttöä voisi laajentaa alueelle, jossa mobiili internet ei toimi.

## 7 POHDINTA

Projektin alussa ei ollut aivan selvää, mitä sovellusta alan kehittämään. Lopulta tavoitteeksi tuli JokaMies sovelluksen kehittäminen Windows Phone puhelimelle, jonka jälkeen kehitystyö saatiin nopeasti käyntiin. Sovellusta oli jo melko pitkälle kehitetty Android alustalle, joten tehtäväkseni tuli saada kyseinen koodi toimimaan Windows Phone alustalla.

Sovelluskehityksessä suurin urakka oli tehdä BluetoothPlugin Windows Phonelle, mikä osoittautuikin aluksi melko hankalaksi hommaksi. Itselläni ei ennen tätä projektia ollut käytännössä mitään kokemusta Windows Phonelle koodaamisesta, mutta onneksi C# koodikieli oli melko samanlaista kuin Java, mikä helpotti työtä huomattavasti. Muiden projektissa olevien henkilöiden apu koodaamisessa oli myös ensiluokkaisen tärkeää. BluetoothPlugin saatiin kuitenkin toimimaan, mutta PhoneGap jouduttiin päivittämään 2.6 versiosta 2.9 versioon, jotta plugin saatiin toimimaan moitteitta.

HTML5-tekniikat olivat myös itselleni melko tuntematon asia ennen projektia ja niiden soveltamisesta mobiiliympäristöön itselläni ei ollut mitään kokemusta. HTML5 sovelluksen toimimaan saaminen Windows Phonella oli aluksi melko haastava

homma. Sovellus saatiin toimimaan Windows Phonella vasta kun AngularJS-ohjelmistokehys vaihdettiin Backbone-ohjelmistokehykseen.

Projekti kokonaisuudessaan oli erittäin opettava kokemus. Projektissa käytettiin työhallintajärjestelmänä Kanbanflow palvelua, mikä oli itselleni uutta palveluna ja tekniikkana. Projektin aikana opin myös koodaamista enemmän kuin koko muun koulun aikana yhteensä. Suurena apuna omassa työssä oli projektin muut henkilöt ja erityisesti Taneli Hartikainen, joka jakoi projektin aikana neuvoa koodaukseen liittyvissä ongelmissa.

## LÄHTEET

Android on 81 percent of new smartphones, Windows Phone share grows: IDC Q3. 2013. NDTV. Viitattu 14.1.2014. <http://gadgets.ndtv.com/mobiles/news/android-on-81-percent-of-new-smartphones-windows-phone-share-grows-ids-q3-445339>

C Sharp (programming language). 2014. Wikipedia - vapaa tietosanakirja. Viitattu 9.1.2014. [http://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language))

Getting started with developing for Windows Phone. 2014. Microsoft. Viitattu 14.1.2014. [http://msdn.microsoft.com/library/windowsphone/develop/ff402529\(v=vs.105\).aspx](http://msdn.microsoft.com/library/windowsphone/develop/ff402529(v=vs.105).aspx)

Hachman, A. 2013. Microsoft Stays Conservative On Web Standards In IE9-To-IE10 Shift. Viitattu 24.2.2014. <http://readwrite.com/2013/02/26/microsoft-takes-conservative-approach-to-web-standards-in-ie9-to-ie10-shift#awesm=~owNDgoyGCFHKYo>

Hemery, G. 2012. Forest mensuration on a smartphone. Viitattu 25.2.2014. <http://gabrielhemery.com/2012/04/20/forest-mensuration-on-a-smartphone/>

HTML5 Introduction. N.d. W3Schools. Viitattu 13.1.2014. [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

Joala, V. 2006. Laserkeilauksen perusteita ja mittauksen suunnittelu. Viitattu 6.3.2014. <https://docs.google.com/file/d/0B3MfAq-wXowlN2Q4MzJlYjktZTA5Ni00ZGMylTIkOWUtNTQzMdIwZTI3NDVm/edit?hl=en>

Korpilahti, E. 2009. Metsänomistuksen pirstoutuminen ja metsikkötalous. Metsätieteen aikakauskirja 3/2009. Saatavissa: <http://www.metla.fi/aikakauskirja/full/ff09/ff093181.pdf>

Metsät kasvavat vauhdikkaasti – hoitotoimien intensiteetti vaihtelee. 2012. Metsäntutkimuslaitos. Viitattu 14.1.2014. <http://www.metla.fi/tiedotteet/2012/2012-06-19-vmi.htm>

Microsoft to acquire Nokia's devices & services business, license Nokia's patents and mapping services. 2013. Microsoft. Viitattu 14.1.2014. <http://www.microsoft.com/en-us/news/press/2013/sep13/09-02announcementpr.aspx>

Monilähteinen VMI. 2010. Metsäntutkimuslaitos. Viitattu 25.2.2014. <http://www.metla.fi/ohjelma/vmi/vmi-moni.htm>

Nokia now controls 90% of the Windows Phone 8 market, with the low-end Lumia 520 grabbing over 35% share. 2013. Thenextweb. Viitattu 14.1.2014. <http://thenextweb.com/microsoft/2013/11/27/nokia-now-controls-90-windows-phone-8-market-low-end-lumia-520-accounting-35-share/#!r9WEt>

Pesonen, A., Korhonen, K., Tuominen, S., Maltamo, M. & Lukkarinen, E. 2007. Taimikonhoitotarpeen arviointi valtakunnan metsien inventoinnin metsävarakartan pohjalta. Metsätieteen aikakauskirja 2/2007. Saatavissa: <http://www.metla.fi/aikakauskirja/full/ff07/ff072077.pdf>

Pihlava, M. 2008. Laserkeilaus mittaa metsät yhden puun tarkkuudella. Viitattu 6.3.2014.

<http://www.tekniikkatalous.fi/metsa/laserkeilaus+mittaa+metsat+yhden+puun+tarkkuudella/a77320>

Relaskooppi. 2013. Wikipedia - vapaa tietosanakirja. Viitattu 15.1.2014.

<http://fi.wikipedia.org/wiki/Relaskooppi>

Relasphone. 2014. VTT Technical Research Centre of Finland. Viitattu 6.3.2014.

<https://play.google.com/store/apps/details?id=fi.vtt.socialforest>

Tikkanen, E. 2010. Metsätilat pirstoutuvat. Viitattu 6.3.2014.

[http://yle.fi/uutiset/metsatilat\\_pirstoutuvat/5659592](http://yle.fi/uutiset/metsatilat_pirstoutuvat/5659592)

Trestima Oy. 2013. Viitattu 25.2.2014. <https://www.trestima.com/>

Windows Phone 8. 2014. Wikipedia - vapaa tietosanakirja. Viitattu 9.1.2014.

[http://en.wikipedia.org/wiki/Windows\\_Phone\\_8](http://en.wikipedia.org/wiki/Windows_Phone_8)

Windows Phone 8 Platform Guide. N.d. PhoneGap Documentation (2.9.0). Viitattu

9.1.2014. [http://docs.phonegap.com/en/2.9.0/guide\\_getting-started\\_windows-phone-8\\_index.md.html#Windows%20Phone%20Platform%20Guide](http://docs.phonegap.com/en/2.9.0/guide_getting-started_windows-phone-8_index.md.html#Windows%20Phone%20Platform%20Guide)



## LIITTEET

### Liite 1. JokaMies-sovelluksen käyttöohje

#### KÄYTTÖOHJE

JokaMies metsänmittaus mobiilisovellus on kehitetty metsän inventoinnin ja työvaikeusluokituksen määrittämisen helpottamiseksi. Sovellus toimii Android ja Windows Phone 8 puhelimilla.


#### Käyttöönotto

Sovellus tulee jakoon Googlen Play kauppaan ja Microsoftin Marketplace kauppaan. Sovellus asentuu puhelimeen automaattisesti ja vaatii oikeuksiksi Androidilla sekä Windows Phonella paikkatiedot, karttapalvelut, verkkovierailut, Bluetooth oikeudet ja oikeuden käyttää web-selainta. Sovelluksen asentamisen jälkeen sovellus on heti käyttövalmis.

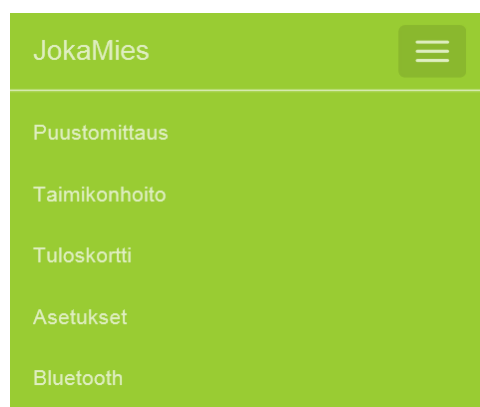
#### Navigointi sovelluksessa

Sovelluksen aloitusruudussa on ohjelman jokaiselle toiminnolle oma painikkeensa. Sovelluksessa liikutaan näiden painikkeiden avulla. Edelliselle sivulle pääsee painamalla Android tai Windows Phone puhelimen alareunassa olevaa "takaisin" näppäintä. Aloitusruutuun pääsee aina painamalla vasemmassa yläreunassa olevaa JokaMies tekstiä.

Sovelluksessa voi liikkua myös oikeasta yläkulmasta painamalla

avautuvan navigointivalikon avulla. Valikon saa auki painamalla  - nappia, jolloin avautuu valikko, jossa on kaikki samat kohteet kuin aloitusruudussa. Valikon voi

avata millä sivulla tahansa ja käyttäjä pääsee haluamalleen sivulle menemättä aloitusruudun kautta.



## Käyttö

Sovelluksessa on kaksi eri mittaustapaa. Puustomittaus on metsän inventointiin tarkoitettu työkalu ja taimikonhoito on tarkoitettu alikasvoslaskentaan, jolla määritetään kuvion raivauksen tarve. Muita toimintoja ovat Tulokortti, Asetukset sekä Bluetooth.

### Puustomittaus



Puustomittaus toiminnon pääominaisuudet ovat kartta, lajike valikko, läpimitta valikko ja lähetä nappi. Kartalta näkee käyttäjän sijainnin ja GPS virheen. GPS virhe näytetään ympyränä, joka on sitä isompi, mitä suurempi virhe on. Todellinen sijainti on jollain kohtaa ympyrän sisällä. Lajike valikosta valitaan puun laji ja läpimitta valikosta puun läpimittaluokka.

Mittaus tapahtuu mittaamalla metsässä olevan puun läpimitta halutulla mittalaitteella. Puun läpimitta voidaan myös arvioida silmämääräisesti, jos se tarkkuudeltaan riittää. Kun puun laji ja läpimitta on selvitetty, asetetaan arvot kyseisiin valikoihin, laitetaan puhelin mahdollisimman lähelle puun runkoa ja painetaan "Lähetä" nappia. Puhelin määrittää puun sijainnin puhelimen oman GPS:n avulla, joten on tärkeää, että puhelin on mahdollisimman lähellä puuta. Sovellus ei hyväksy mittauksia, jos GPS virhe on liian suuri. Virhe marginaalia voi säätää asetuksista (kts. Asetukset). Puita mitataan satunnaisessa järjestyksessä, kuitenkin niin että samaa puuta ei mitata kuin kerran.

Puun voi mitata myös Bluetooth yhteydellä varustetulla mittasaksilla (esim. Masser BT Caliper). Kun mittasakset on yhdistetty puhelimeen (kts. Bluetooth), voi mittauksen tehdä sovelluksen ollessa missä tilassa tahansa. Mittaus aloitetaan valitsemalla mittasaksista mitattavan puun laji. Sen jälkeen puhelin viedään mahdollisimman lähelle puun runkoa, kun puun läpimitta mitataan. Puun mittauksen jälkeen mittasakset lähettää tiedot puhelimeen ja puhelin määrittää sijainnin puhelimen oman GPS:n avulla.

### Taimikonhoito



Taimikonhoito toiminto on kuvion risukon raivaamiseen menevän ajan määrittämistä helpottava työkalu. Taimikonhoito ikkunasta löytyy kartta, alan säde valikko, risukon määrä kenttä ja risukon läpimitta

valikko. Kartta näyttää käyttäjän sijainnin ja kuviot ennalta määrätyn etäisyyden sisältä (oletus 200 metriä). Kuvioille sovellus tekee automaattisesti mikrokoealoja. Mittaus tapahtuu menemällä yhdelle tällaiselle mikrokoealalle, laskemalla kaikki alle 6 cm läpimitaltaan olevat risut ja määrittelemällä risuille keskimääräinen läpimittaluokka. Tämän jälkeen sovellukseen syötetään koealan säde, risujen määrä ja risujen keskimääräinen läpimittaluokka. Kun tiedot on lisätty sovellukseen, painetaan “Lähetä”-nappia.

Työvaikeusluokituksen laatua voi parantaa tekemällä mahdollisimman monta mittausta. Mittauksia tehdään kunnes saavutetaan haluttu keskivirhe. Keskivirheen näkee sovelluksen Tulokset-toiminnon kautta (kts. Tulokset).

### Tuloskortti



Tuloskortti kohdassa voidaan tarkastella tuloksia kuvioittain. Puhelimeen ladataan tiedot kaikista kuvioista ja sovellus listaa kaikki kuviot. Kuvioista valitaan kuvio, jonka tuloksia halutaan tarkastella.

Kuvio sivulla näytetään kartalla kuvion sijainti, kuvion nimi, maastoluokka ja kausiluokka. Kahta viimeisintä voi tarvittaessa myös muuttaa.

Kuviolle sovelluksen tekemät tunnusluvut saadaan näkyville painamalla “Hae Tulokset”-nappia. Sovellus hakee tiedot taustajärjestelmästä ja näyttää ne sovelluksessa. Tulokset jaetaan kahteen osaan, puustomittaukseen ja ennakkoraivauksen työvaikeusluokituksen. Puustomittaus kohdassa näytetään tiedot, jotka liittyvät puuston inventointiin ja ennakkoraivauksen työvaikeusluokituksessa tiedot, jotka liittyvät työvaikeusluokituksen.

### Asetukset



Asetukset kohdasta voidaan määrittää sovelluksen asetuksia.

**Metsätietojärjestelmän osoite** on sen serverin url-osoite, jonka kanssa sovellus keskustelee.

**Karttapalvelu** kohdassa määritellään karttapalvelun (Tile Map Service, TMS) osoite, jota sovellus käyttää. Karttapalveluksi kelpaa kaikki tile map service palvelut.

**GPS-virhemaksimi** on arvo, jonka alle menevät mittaukset sovellus hyväksyy. (oletus 30m)

**Kuviohaun kantama** on arvo, joka määrittää matkan, kuinka kaukana omasta sijainnista olevat kuviot näytetään Työvaikeuslukitus toiminnon kartalla.

**Puumerkinnän koko** on ympyrän koko metreinä, mikä näyttää puun sijainnin kartalla. (oletus 9m)

**Risukkomerkinnän koko** on ympyrän koko metreinä, mikä näyttää risukon sijainnin kartalla. (oletus 9m)

**Esimerkkikoealan koko** on oletuskoko mikrokoealalle, joka on esivalittuna Työvaikeusluokitus toiminnossa. (oletus 1.78m)

**Esimerkkikoealan väli** on matka, minkä välein sovellus määrittää mikrokoealoja Työvaikeusluokitus toiminnossa. (oletus 20m)

**Yhteystyyppi** kohdasta valitaan Bluetoothin turvallisuusluokka. Luokkaa voi laskea lisätäkseen yhteensopivuutta vanhempien Bluetooth laitteiden kanssa. *Ei toimi Windows Phone sovelluksessa.*

## Bluetooth



Bluetooth valikosta voi yhdistää puhelimen sovellusta tukevien Bluetooth laitteiden kanssa. Sovellus on testattu toimivaksi Masser BT Caliper mittasaksien kanssa, mutta myös muut mittasakset, jotka käyttävät NMEA tiedonsiirtoprotokollaa toimivat sovelluksen kanssa.

Laitteita etsitään painamalla sivulla olevaa "Etsi"-nappia. Sovellus etsii kaikki paritetut laitteet ja listaa ne napin alle. Listasta painamalla laitteen nimeä sovellus yhdistää laitteeseen. Kun yhdistäminen tapahtuu onnistuneesti, laitteen nimen alle tulee teksti "yhdistetty". Laitteen ollessa yhdistettynä voidaan mittauksia alkaa tehdä välittömästi, riippumatta siitä millä sivulla sovelluksessa ollaan (kts. Puustonmittaus). Yhteys laitteeseen katkaistaan painamalla uudestaan yhdistetyn laitteen nimeä. Yhteys katkeaa automaattisesti, jos mittalaite viedään liian kauaksi puhelimesta.

## Taustajärjestelmä

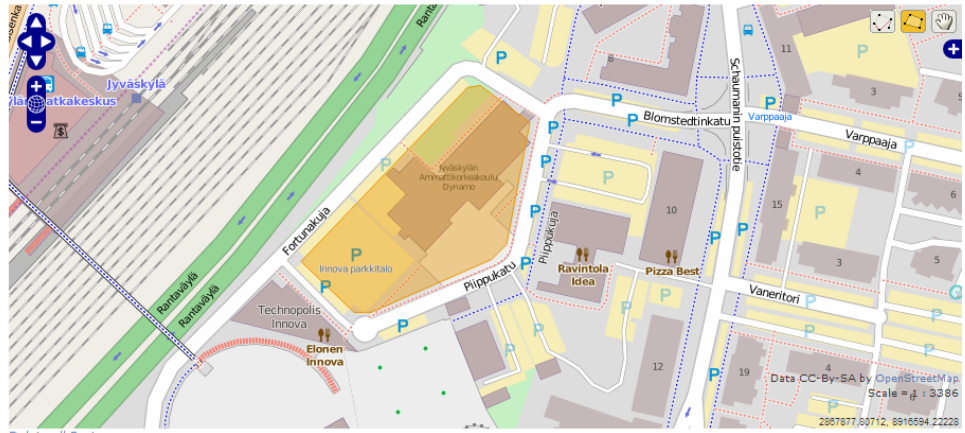
Sovellus tarvitsee toimiakseen taustajärjestelmän. Taustajärjestelmä tekee kaikkein vaativimmat laskutoimitukset ja taustajärjestelmään lisätään kuvioiden rajat, mistä mobiilisovellus hakee ne puhelimeen.

## Kuviorajojen lisääminen

Kuviotiedot lisätään järjestelmään menemällä selaimella taustajärjestelmän toimivan JokaMies serverin osoitteeseen (esim. [mehta.labranet.jamk.fi/admin/](http://mehta.labranet.jamk.fi/admin/)). Sivulla kirjaudutaan sisään, jolloin päästään käsiksi kuviotietoihin. Tiedoista löytyy kuvioiden rajat, mittaustiedot sekä muuta tietoa järjestelmästä. Rajat lisätään menemällä "Stands"-kohtaan ja klikkaamalla sivun oikeassa yläkulmassa olevaa "lisää stand"-nappia. Nappia painamalla pääsee sivulle, jossa kartalle voidaan piirtää kuvion rajat ja määrittää kuvion nimi sekä muita tietoja. Kun rajat on piirretty ja tiedot määritetty, painetaan oikeassa alakulmassa olevaa "Tallenna ja poistu"-nappia, jolloin tiedot tallentuvat järjestelmään.

### Lisää stand

Shape:



Delete all Features

Name:

Owner:  +

Ground type:

Snow level:

Tags:   
A comma-separated list of tags.

Kuvio rajat voidaan myös lisätä tiedostoina menemällä kuvion lataus sivulle (esim. [mehta.labranet.jamk.fi/upload\\_stand/](http://mehta.labranet.jamk.fi/upload_stand/)). Sivulla on ohje, miten kuviotiedot ladataan serverille ja mitä tiedostoja siihen tarvitsee. Järjestelmä tekee tiedostojen avulla kuviorajat ja määrittää kuviolle nimen. Kuvion tietoja voi jälkepäin muokata admin sivustolta.

## Latausohjeet

### Shapefile

Ladatessasi shx, shp ja dbf -tiedostot ovat pakollisia, jotta kuvioiden lataus on mahdollista

### Metsätietostandardi(XML)

Tällä hetkellä voi syöttää vain yhden XML-tiedoston kerrallaan

### Ladattavat tiedostot

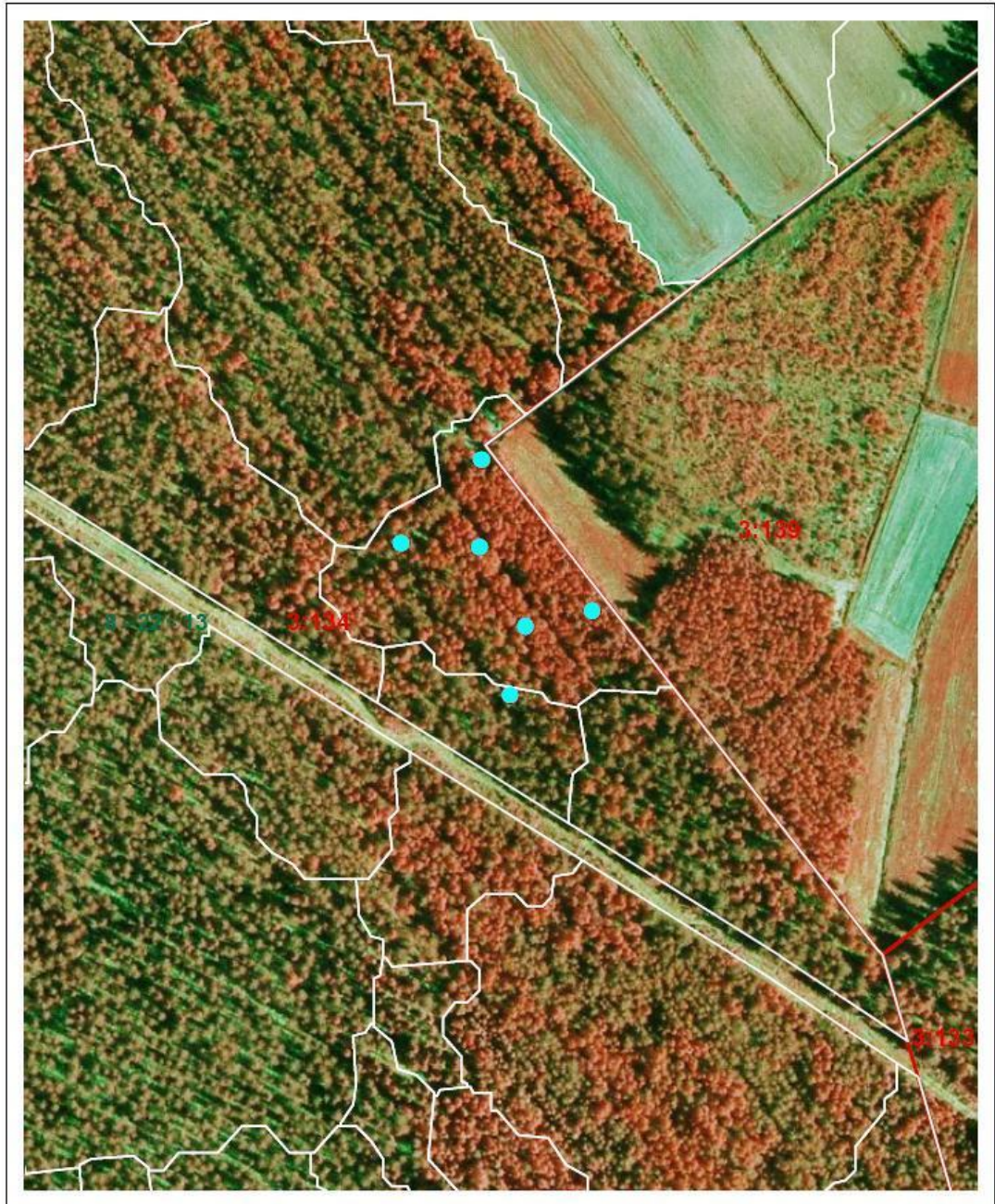
Ei valittua tiedostoa

### Mittaustulosten tarkastelu

Mittauksia voidaan tarkastella menemällä selaimella JokaMies serverin admin sivulle.

Sivulta voi tarkastella mittauksia "Tree measurements"-kohdasta. Mittaukset listataan yksitellen ja mittausta klikkaamalla voidaan tarkastella mittauksen tietoja kuten puun lajia ja läpimittaa, mittauksen ajankohtaa sekä mittauksen sijaintia.

## Liite 2. Ilmakuva Liikalan koealueesta

**Toivakka**

### Liite 3. Laserkeilaus pistepilvi Liikalasta

