

Lappeenrannan teknillinen yliopisto
Tuotantotalouden tiedekunta
Tietotekniikan koulutusohjelma

Kandidaatintyö

Joonas Pohjala

**SUUNNISTUSSOVELLUKSEN SUUNNITTELU WINDOWS
PHONELLE**

Työn tarkastaja: TkT Ari Happonen

Työn ohjaajat: TkT Ari Happonen
DI Antti Knutas

Päiväys: 11.12.2013

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto

Tuotantotalouden tiedekunta

Tietotekniikan koulutusohjelma

Joonas Pohjala

Suunnistussovelluksen suunnittelu Windows Phonelle

Kandidaatintyö

2013

37 sivua, 5 kuvaa, 2 taulukkoa

Työn tarkastaja: TkT Ari Happonen

Hakusanat: suunnistus, paikannus, mobiilisovellus

Keywords: orienteering, navigation, mobile application

Tässä työssä luotiin suunnistussovelluksen demo Windows Phone 8:lle. Työn tavoite oli tehdä sovellus, jonka kanssa suunnistaminen tuntuisi mahdollisimman samalta, kuin oikeakin suunnistus. Työssä kiinnitettiin huomiota lähinnä sovelluksen tekniseen toteutukseen sekä paikannuksen riittävään tarkkuuteen. Työssä vertaillaan eri mobiilialustoja, selainten tietokantaratkaisuja, erilaisia selaimen ja serverin tietokantojen synkronointimenetelmiä sekä olemassaolevia suunnistussovelluksia. Työssä käytettiin indeksitietokantaa, jQuery mobilea ja Cordovaa sekä JSON:ia tietokantojen synkronointiin. Työssä tehty demo on lupaava esitys siitä, miten suunnistustaitojaan voisi harjoittaa tekniikkaa apuna käyttäen. Lopuksi työssä käydään vielä läpi sitä, mitä tällaiseen palveluun voisi toteuttaa työssä toteutetun demon lisäksi.

ABSTRACT

Lappeenranta University of Technology
School of Industrial Engineering and Management
Degree Program in Computer Science

Joonas Pohjala

Planning an orienteering application for Windows Phone

Bachelor's Thesis

37 pages, 5 figures, 2 tables

Examiners: D.Sc. (Tech.) Ari Happonen

Keywords: orienteering, navigation, mobile application

In this thesis there were created an orienteering application for Windows Phone 8. The goal of the thesis was to create an application, which would allow user to orienteer as similarly to normal orienteering, as possible. In the thesis the concentration was on technical implementation and on accurate enough navigation. In the thesis there are compared different mobile platforms, databases in browsers, different browser and server database synchronizing methods and already made orienteering applications. In the demo there were used indexed database, jQuery mobile, Cordova and JSON for synchronizing browser and server databases. The demo was a promising presentation, that orienteers could improve their orienteering abilities with a little help of technology. At the end there are discussions of what could be implemented on an application like this.

ALKUSANAT

Tämä kandidaatintyö aloitettiin alkukesästä 2013 ja vasta nyt joulukuussa voi sanoa työn olevan valmis. Sovelluksen osalta työ jatkuu, toivottavasti myös opintojeni jälkeen. Aikatauluun vaikutti suuresti se, että sovellus on melko laaja-alainen ja sen tekemiseen meni yllättävän paljon aikaa. Työhön halusin myös paljon sellaista, mitä työ ei olisi välttämättä vaatinut. Tätä työtä tehdessä oppi ainakin sen, että ensin kannattaa ihmetellä ja vasta sitten tehdä, muuten voi tulla tehtyä valtavasti aivan turhaa työtä ennen kuin huomaa ihmetellä. Työnteossa suurena apuna ovat olleet rakas kihlattuni sekä suunnistuksesta valtavasti tietotaitoa omaava isoveljeni, jotka molemmat ovat jaksaneet kuunnella ideoitani ja suunnitelmiani ja ovat jakaneet ajatuksiaan minulle. Kotiväki on ollut suurena tukena kyselemällä työn etenemistä ja työn ohjaajat ovat antaneet suurta tukea työn valmistumiselle. Varsinkin Ari on jaksanut vastaila maileihini myöhäisin illoin ja jakanut ideoita, jotka eivät olisi tulleet itselleni mieleenkään.

SISÄLLYSLUETTELO

1	JOHDANTO.....	3
1.1	TAVOITELTU TOTEUTUS	3
1.2	TAVOITTEET JA RAJAUKSET	4
1.3	TYÖN RAKENNE	5
2	TEORIAA JA EMPIIRIAA KÄYTÄNNÖN TOTEUTUKSEN TUKENA.....	6
2.1	SUUNNISTUS	6
2.2	PAIKANNUS.....	7
2.2.1	<i>Paikannuksen perusteita</i>	8
2.3	MOBIILIALUSTAT	10
2.4	JQUERY MOBILE JA CORDOVA.....	11
2.5	SELAINTIETOKANNAT	12
2.6	TIETOKANTOJEN SYNKRONOINTI.....	15
2.7	SAMANKALTAISET SOVELLUKSET	16
3	SOVELLUKSEN TOTEUTUS.....	19
3.1	PAIKANNUS.....	19
3.2	MOBIILIALUSTA	21
3.3	JQUERY MOBILE JA CORDOVA.....	22
3.4	SELAINTIETOKANNAT	23
3.5	SYNKRONOINTI	23
4	POHDINTA JA TULEVAISUUS	24
4.1	SAMANKALTAISET SOVELLUKSET	24
4.2	SOVELLUKSEEN TEHTÄVÄT MUUTOKSET JA LISÄYKSET	25
4.3	SELAINSOVELLUS.....	27
	YHTEENVETO.....	29
	LÄHTEET	30

SYMBOLI- JA LYHENNELUETTELO

A-GPS	Assisted GPS
AJAX	Asynchronous JavaScript And XML
API	Application programming interface
CSS	Cascading Style Sheets
DGPS	Differential GPS
EGNOS	European Geostationary Navigation Overlay Service
Glonass	Globalnaja navigatsioonaja sputnikovaja sistema, Venäjän paikannusjärjestelmä
GPS	Global Positioning System
HTML	Hypertext Markup Language
IDB	Indexed database
IOF	International Orienteering Federation
IRNSS	Indian Regional Navigational Satellite System
JPEG	Joint Photographic Experts Group
JSON	JavaScript object notation
NFC	Near field communication
OCad	Orienteering CAD
QR	Quick response
RTK	Real Time Kinematic
SQL	Structured Query Language
SSL	Suomen Suunnistusliitto
XML	Extensible Markup Language
W3C	World Wide Web Consortium
WAAS	Wide Area Augmentation System
WP7	Windows Phone 7
WP8	Windows Phone 8

1 JOHDANTO

Olen harrastanut suunnistusta pienestä pitäen ja minua kiinnostaa siinä tarvittava nopeuden ja taidon tasapaino. Jos juoksee liian lujaa, rasteista menee helposti ohi, kun taas hitaammin ja tarkemmin suunnistava voi olla nopeampi jollakin rastivälillä. Suunnistuksen harjoittelua voisi tehostaa huomasti nykyisillä mobiiliteknologioilla. Esimerkiksi paikannus on jo niin tarkkaa, että muutaman metrin tarkkuudella voidaan määrittää suunnistajan sijainti [1]. Esimerkiksi Jukolan viestissä kärkijoukkueilla on GPS-vastaanottimet, jotka lähettävät sijaintia kilpailuorganisaatiolle koko ajan. Näin suunnistuksesta on myös saatu yleisöystävällisempi laji.

Suunnistussovelluksia on olemassa muutamia, kuten GPS Orienteering ja V-Orienteering. Olemassaolevista sovelluksista kerrotaan enemmän luvussa 3.1. Tässä työssä pyritään luomaan demo sovelluksesta, jossa tarkistetaan paikannuksella, onko käyttäjä rastipisteessä. Paikannus mahdollistaa sen, ettei rastipisteessä tarvitse olla mitään kiinteää leimasinta, vaan rastipiste on aina olemassa ja käytettävissä. Muita toteutustapoja voisivat olla esimerkiksi QR-koodit tai NFC-tägit mutta ne saattaisivat hävitä rastipisteeltä tai rikkoutua käyttökelvottomiksi.

1.1 Tavoiteltu toteutus

Sovelluksen tavoitellussa toteutuksessa käyttäjällä on käytettävissään sovelluksen mobiiliversio, joka on tarkoitettu lähinnä suunnistukseen sekä tarvittaessa reittivaihtoehtojen selailuun, ja selainversio, joka on tarkoitettu sovelluksen tarjoamien tietojen, kuten eri reittivaihtoehtojen ja tulosten, selailuun. Suurella näytöllä on helpompi selaila erilaisia tietoja, joten suurin osa reittien hankinnoista tehtäisiin selainsovelluksella.

Ensimmäiseksi käyttäjän pitää rekisteröityä sovellukseen. Tämä voidaan tehdä joko mobiilisovelluksesta, tai selainsovelluksesta. Käyttäjä saa automaattisesti oikeudet selata, ostaa ja suunnistaa reittejä sekä selata reittivaihtoehtojen tietoja. Myöhemmin käyttäjälle voidaan lisätä oikeudet luoda ratoja, jolloin käyttäjä saa käyttöönsä ratojentek ominaisuudet. Niillä hän voi luoda uusia ratoja, asettaa niille rastipisteitä, lisätä reitin kartan sovellukseen sekä tietenkin julkaista reitin. Reittiä julkaistaessa reitin pitää olla valmis, eikä siihen voi enää tehdä suuria muutoksia. Kun rata on julkaistu, se on

ostettavissa palveluun luotavasta kaupasta.

Käyttäjä voi hankkia reittejä selain- ja mobiilisovelluksella palvelun kaupasta. Käyttäjän hankittua reitin, se synkronoituu hänen älypuhelimensa sovellukseen automaattisesti. Käyttäjä voi tulostaa reitin kartan selainsovelluksella tai ladata kartan puhelimeensa. Käyttäjän on mahdollista suunnistaa joko paperisella kartalla tai mobiilisovelluksessa olevalla kartalla. Kun haluttu rata on valittu mobiilisovelluksessa, kerrotaan käyttäjälle, kun lähtöpiste on saavutettu. Käyttäjä on voinut määrittää asetuksista, haluaako hän ilmoittaa sovellukselle olevansa rastilla, vai haluaako hän sovelluksen ilmaisevan rastin löytymisestä. Asetuksista voi määrittää myös, haluaako sovelluksen tallentavan käyttäjän kulkeman reitin myöhempää tarkastelua varten.

Kun kaikki rastit on kierretty ja maalipiste on löytynyt, ilmoittaa sovellus ajan sekä sijoituksen radalla. Käyttäjä voi ilmoittaa, haluaako hän nimensä julkaistavan tuloksen yhteydessä radan tiedoissa. Mikäli käyttäjä valitsee ”En salli”, julkaistaan käyttäjän suorituksesta vain aika, väliaikatiedot sekä reitti. Jokaisen reitin tuloksia voi selata sekä selain-, että mobiilisovelluksella.

1.2 Tavoitteet ja rajaukset

Tämän kandidaatintyön tarkoitus on ollut luoda WP8:lle (Windows Phone 8) suunnistussovelluksen demo, jolla voi sekä tehdä erilaisia reittejä, että suunnistaa niitä. Demo ei käsittele kaikkia käyttötapauksia, ja työssä keskitytään pelkästään laitteessa olevaan sovellukseen, vaikka laite pystyykin synkronoimaan reittejä myös verkossa olevien reittien kanssa.

Tässä työssä tärkeää on se, että sovelluksella suunnistaminen olisi mahdollisimman lähellä oikeaa suunnistamista. Tämä tarkoittaa, että paikannuksella tarkistettavien rastipisteiden pitää olla riittävän tarkkoja. Suunnistuskartta puhelimessa tuottaa monenmoisia ongelmia, kuten mittakaavan ja kartan suuntauksen. Pitäisikö kartan pyöriä kompassin lailla, jolloin jo muutaman asteen virhe saattaisi aiheuttaa käyttäjälle harmaita hiuksia, vai siten, että käyttäjä kääntää älypuhelimensa suunnatakseen kartan. Tässä työssä ei myöskään keskitytä sovelluksen kuluttamaan energiamäärään eikä akun keston, vaan pitäydytään suunnistajan vaatimien tarpeiden täyttämässä.

Suuren ongelman luo myös se, että suunnistuskartat ovat suunnistusseurojen omaisuutta, ja jo karttojen saaminen tällaiseen käyttöön vaatii erillisiä sopimuksia, koska monikaan seura ei halua antaa tuhansien eurojen panostusta vaatineita karttoja ilmaiseksi jakoon. Suunnistuskartoissa maastoa on kuvattu yksityiskohtaisemmin, kuin ilmaiseksi saatavilla olevassa karttamateriaalissa, kuten maanmittauslaitokselta saatavassa materiaalissa.

1.3 Työn rakenne

Luvussa 1 esitellään työn taustoja, tavoitteita ja rajoituksia. Luvussa 2 käsitellään työhön liittyviä asioita teoreettiselta kannalta. Luvusta saa yleiskuvan työn aiheesta ja siihen liittyvistä seikoista. Luvussa 3 käsitellään, miten edellisessä luvussa esiteltyjä asioita tehtiin tässä työssä ja miksi kyseisen tavan käyttöön päädyttiin. Luvussa 4 käydään läpi sovelluksen tulevaisuutta, eli mitä sovellukseen pitää toteuttaa ennen julkaisua, sekä mitä siihen voi toteuttaa julkaisun jälkeen. Luvussa 5 esitetään yhteenveto.

2 TEORIAA JA EMPIRIAA KÄYTÄNNÖN TOTEUTUKSEN TUKENA

Työn tekemiseen vaadittiin tietoa monelta eri alueelta, tärkeimpinä suunnistus, mobiilitekniikat ja paikannus. Tämän luvun tarkoituksena on esittää työssä esille nousseita teemoja ja avata niitä. Tässä luvussa kerrotaan, miten asiat voisi tehdä ja seuraavassa luvussa paneudutaan siihen, miten ne juuri tässä työssä toteutettiin ja miksi ne toteutettiin tehdyllä tavalla.

2.1 Suunnistus

Suunnistus on urheilulaji, jossa on tarkoitus kiertää karttaan merkityt rastipisteet vain kartan ja kompassin avulla. Suunnistusta voidaan harrastaa monella tavalla monessa eri ympäristössä. Perinteistä suunnistusta harrastetaan yleensä metsäisessä maastossa, mutta suunnistuksen alalajia, sprinttisuunnistusta, yleensä kaupunkialueella. IOF:n (International Orienteering Federation) muita suunnistuslajeja ovat hiihtosuunnistus, jossa liikutaan hiihtäen, pyöräsuunnistus, jossa liikutaan pyöräillen, sekä tarkkuussuunnistus, jossa painotus on nopeuden sijaan tarkkuudella ja maastossa voi liikkua vaikka pyörätuolilla [2]. Lisäksi lajiliiton ulkopuolisia lajeja ovat esimerkiksi autosuunnistus ja kanoottisuunnistus.

Suunnistus on saanut alkunsa armeijan urheilulajina. Maailman ensimmäiset viralliset suunnistuskilpailut olivat Norjassa vuonna 1897 [3]. Suomen ensimmäinen suunnistustapahtuma oli vuonna 1904 järjestetty hiihtosuunnistusviesti. Suomen Suunnistusliitto perustettiin vuonna 1945. SSL:n (Suomen Suunnistusliitto) mukaan vuonna 2013 suunnistusta harrastaa Suomessa noin 60 000 eri ikäistä ihmistä [4], kansainvälisesti harrastajia on moninkertaisesti enemmän. Maailman suurin suunnistustapahtuma, Suomessa järjestettävä Jukolan viesti, sekä sen yhteydessä järjestettävä, naisille suunnattu, Venlojen viesti, keräävät vuosittain liki 15 000 osallistujaa noin 20:stä maasta [5]. Kilpailupaikalle saapuu vuosittain 30 000 – 40 000 osanottajaa, huoltajaa ja katsojaa. Suunnistus on suosittu erityisesti Pohjoismaissa, mutta lajilla on paljon harrastajia myös Keski- ja Etelä-Euroopassa, Venäjällä, pohjois-Amerikassa sekä Australiassa. Suunnistusta voi harrastaa miltei missä tahansa, kuten veden alla [6] tai aavikolla [7].

Suunnistuskartat ovat yleensä mittakaavaltaan 1:10 000 tai 1:15 000, sprinttikartoissa ja opetuskartoissa on myös tarkempia mittakaavoja. Kartoissa on kuvattu maastoa yksityiskohtaisemmin, kuin normaaleissa topografisissa kartoissa ja siinä on korostettu suunnistajalle tärkeitä kohteita, kuten maastonmuotoja, kosteikkoja ja kasvuston tiheyttä [8]. Nykyään osa kartoista tehdään laserkeilausaineiston ja paikannuksen avulla, jolloin kaikille kartan pisteille tiedetään koordinaatit. Tunnetuin suunnistuskartoissa käytetty kartoitusohjelma on OCad (Orienteering CAD) [9].

2.2 Paikannus

Ihmiset ovat käyttäneet erilaisia paikannusmenetelmiä aina. Aikoinaan on paikannettu erilaisten maamerkkien ja matka-aikojen perusteella, merimiehet taas ovat käyttäneet tähtiä apunaan [10]. Kartat eivät aluksi olleet tarkkoja, koska ne olivat piirretty usein matkustajien muistikuvien mukaan. Kiinalaiset keksivät kompassin ensimmäisinä noin 2500 vuotta ennen ajanlaskun alkua.

Nykyaikainen satelliittipaikannus sai alkunsa, kun Yhdysvaltain laivasto alkoi kehittää 1950-luvun puolivälissä Transit-navigaationsatelliittijärjestelmää [11]. Myöhemmin Yhdysvaltain ilmavoimat kehitti oman Timation satelliittijärjestelmänsä. Hankkeet yhdistyivät 1970-luvun alkupuolella, ja yhdistyneistä hankkeista rakentui GPS-satelliittipaikannusjärjestelmän ensimmäinen versio.

Ensimmäinen GPS-navigointisatelliitti lähetettiin taivaalle vuonna 1978. Yhteensä Block I:den, eli ensimmäisen tyypin, satelliitteja lähetettiin 11, joista yhden saaminen kiertoradalle epäonnistui. Kaikki Block I –tyyppiset satelliitit on poistettu jo käytöstä. Ensimmäinen Block II –satelliitti lähetettiin vuonna 1989. Kaikki vuonna 2013 käytössä olevat GPS-satelliitit ovat Block II, Block IIA, Block IIF ja Block IIR –tyyppisiä [12]. Lockheed Martin suunnittelee paraikaa Block III –tyyppistä satelliittia [13], jossa on uusia ominaisuuksia verrattuna aiempiin satelliitteihin. GPS-satelliitit lähettävät Yhdysvaltojen armeijan lähettämää salattua signaalia, sekä vuodesta 2000 alkaen häiritsemätöntä siviilikäyttöön tarkoitettua salaamatonta signaalia.

Venäjällä on oma Glonass (Globalnaja navigatsionnaja sputnikovaja sistema) satelliittipaikannusjärjestelmä, jonka ensimmäinen satelliitti laukaistiin vuonna 1982 [14].

Glonass-satelliittien signaalia ei ole millään tavalla salattua. Uusimmat mobiililaitteet osaavat hyödyntää sekä GPS, että Glonass –signaaleja [15]. EU:lla on myös tekeillä oma, siviilikäyttöön tarkoitettu Galileo [16]. Projekti on kuitenkin pahasti myöhästynyt ja ylittänyt budjettinsa. Toiminnassa olevaa Galileo-satelliittipaikannusjärjestelmää voitaneen odottaa joskus seuraavalla vuosikymmenellä.

Kiinalaiset ovat kehittäneet omaa Compass (viralliselta nimeltään BeiDou) satelliittipaikannusjärjestelmäänsä [17], jonka olisi tarkoitus toimia globaalisti vuoden 2020 tienoilla. Myös Intialla on rakenteilla oma IRNSS-satelliittipaikannusjärjestelmä (Indian Regional Navigational Satellite System) [18], jonka on tarkoitus parantaa paikannuksen tarkkuutta Intian alueella.

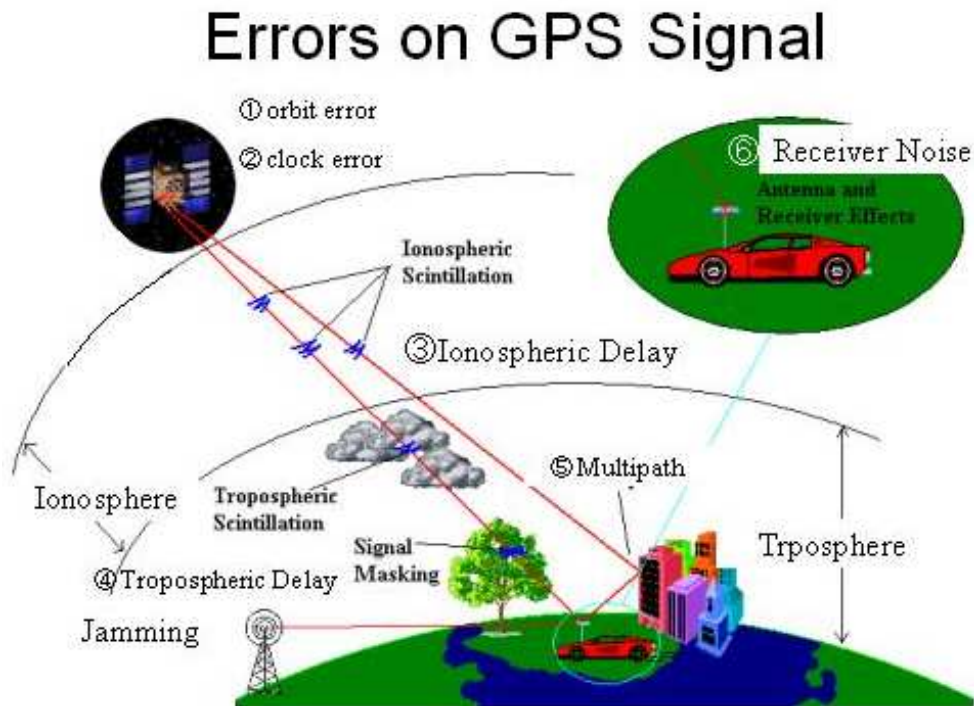
Tämän työn kannalta tärkeää on tietää melko tarkka arvio erilaisten kohteiden, kuten aukeiden, mäntymetsien ja tiheiden kuusikkojen, paikannuksen tarkkuuksista, jotta niiden avulla voidaan laskea rastiympyrä, jossa suunnistajan pitää olla ollakseen sovelluksen mielestä rastilla, riittävän suureksi. Mikäli käyttäjä olisi rastilla, mutta paikannuksen tarkkuuden vuoksi sovellus kuvittelisi käyttäjän olevan rastiympyrän ulkopuolella, joutuisi käyttäjä liikkumaan rastin lähistöllä etsimässä sopivaa koordinaattia.

2.2.1 Paikannuksen perusteita

Satelliittipaikannusjärjestelmät perustuvat satelliitteihin, jotka lähettävät vastaanottimille atomikellon ajan sekä navitaatiosignaalin. Siviilisignaaleissa voidaan laskea paikkaa pseudoetäisyydellä, eli lasketaan satelliitin signaalin kulkemiseen kulunut aika ja ajasta etäisyys. Kun tiedetään riittävän monen satelliitin sijainti ja etäisyys vastaanottimesta, voidaan laskea myös vastaanottimen sijainti. Sijainnin tarkkuus kasvaa, mitä useamman satelliitin signaali saadaan.

Jos oletetaan, että vastaanotin on maapallon pinnalla, yhden satelliitin signaalista voidaan laskea ympyrä, jonka kehällä laite on. Toisella signaalilla voidaan laskea tältä kehältä kaksi pistettä, joista toisessa laite on. Vasta kolmannella signaalilla voidaan laskea laitteen sijainti yksiselitteisesti. Kuitenkin, koska vastaanottimien kellot eivät ole yhtä tarkkoja kuin satelliittien atomikellot, tarvitaan ainakin neljännen satelliitin signaali, ennen kuin sijainti voidaan laskea tarkasti.

Suurimman epätarkkuuden paikannukseen luo vastaanottimen kellon epätäsmällisyys. Toinen tärkeä epätarkkuuden lähde on ionosfääri ja troposfääri. Ongelmia saattavat aiheuttaa myös monitie-eteneminen [19] sekä tiheän kasvillisuuden aiheuttama katve. Mobiililaitteissa virhettä aiheuttaa myös laitteen oma taustakohina. Salaamatonta signaalia voidaan myös häiritä tarkoituksellisesti, mutta se ei liene ongelmana tässä työssä. Kuvassa 1 on esitetty GPS:n suurimmat virhelähteet.



Kuva 1: GPS:n virhelähteet

Paikannuksen tarkkuutta ja ensimmäisen luotettavan sijainnin saamista voidaan parantaa erilaisin menetelmin. DGPS (Differential GPS) on menetelmä, jossa tunnetaan jonkin paikan tarkka sijainti maapallon pinnalla. DGPS-asema laskee koko ajan satelliittien epätarkkuuksia omaan sijaintiinsa nähden, ja lähettää eri tavoin korjaussignaalia vastaanottimille. DGPS:llä voidaan korjata lähettimen ja ilmakehän aiheuttamaa häiriötä. WAAS (Wide Area Augmentation System) on pohjois-Amerikassa toimiva DGPS järjestelmä [20] ja EGNOS (European Geostationary Navigation Overlay Service) WAAS:in Eurooppalainen vastine [21]. Kumpikaan järjestelmä ei toimi tehokkaasti Suomessa, johtuen pohjoisesta sijainnistamme.

A-GPS (Assisted GPS) nopeuttaa ensimmäisen luotettavan sijainnin saamista kertomalla vastaanottimelle sen summittaisen sijainnin, aikatiedon sekä satelliittien ratojen korjaustiedot, jonka jälkeen vastaanotin osaa kuunnella oikeiden satelliittien taajuuksia sekä korjata niiden ratatietoja. RTK (Real Time Kinematic) on menetelmä, jossa kuunnellaan satelliitin kantoaaltoa sen sisältämän datan sijaan [22].

2.3 Mobiilialustat

Mobiilialustoja on useita, mutta niistä tärkeimmät ovat Android, iOS sekä Windows Phone [23]. Taulukossa 1 esitellään mobiilialustojen tärkeimmät tiedot, jossa jokaisesta alustasta käsitellään viimeistä versiota.

Taulukko 1: Mobiilikäyttöjärjestelmien eroavaisuuksia[24][25]

	Android	iOS	WP
Lähdekoodi	Avoim	Suljettu	Suljettu
Julkaisija	Google Inc.	Apple Inc.	Microsoft
Käyttäjät (2012)	68,8%	18,8%	2,5%
Ennuste: käyttäjät 2016	63,8%	19,1%	11,4%
Julkaisu	5.11.2007	Kesäkuu 2007	29.10.2012
Viimeinen versio	4.4 Kitkat	iOS 7	8.0.10512.142
Viimeisen version julkaisu	31.10.2013	18.9.2013	14.10.2013
Sovelluskauppa	Google Play	App Store	WP Store
Sovelluksia, 11/2013	Yli miljoona	Yli miljoona	175 000
HTML 5	Kyllä	Kyllä	Kyllä

Android on alkujaan Android Inc:n ja myöhemmin Google Inc:n kehittämä käyttöjärjestelmä, joka on rakennettu avoimen lähdekoodin Linux-ytimen päälle ja on itsekin lisensoitu Apache-lisenssillä avoimen lähdekoodin ohjelmistoksi [26]. Marraskuussa 2013 Android on laitteiden myyntimäärissä laskettuna johtavassa asemassa älypuhelinmarkkinoilla. Useat valmistajat valmistavat Android-älypuhelimia.

iOS on Apple Inc:n kehittämä suljetun lähdekoodin käyttöjärjestelmä [27]. Kuten muutenkin Applen tuotteissa, on myös iPhone-älypuhelimissa, sekä niiden sisältämässä iOS-käyttöjärjestelmässä ajateltu paljon tyyliä ja helppokäyttöisyyttä. Vain Apple valmistaa iPhone-älypuhelimia. Windows Phone on näistä uusin tulokas. WP on Microsoftin kehittämä suljetun lähdekoodin käyttöjärjestelmä [28], joka muistuttaa monessa suhteessa Microsoftin Windows 8 –käyttöjärjestelmää. Vaikka WP:n markkinaosuus on vielä pieni, uskotaan yleisesti markkinaosuuden kasvavan, kun WP on ollut hieman kauemmin markkinoilla. Windows Phoneja valmistavat useat valmistajat.

Mobiililaitteisiin voi tehdä sovelluksia eri tavoin [29]. Natiivisovellus on tehty laitteen vaatimalla kielellä juuri kyseiselle käyttöjärjestelmälle, esimerkiksi WP:lle voi ohjelmoida C#-ohjelmointikielellä. Web-sovellus on tehty käyttäen web-tekniikoita, kuten HTML:ää (Hypertext Markup Language) ja Javascriptia. Hybridisovellus on näiden kahden väliltä, siinä luodaan natiivikielellä selainikkuna, johon luodaan sisältöä webtekniikoilla. Kahdesta viimeisestä enemmän luvussa 2.4. Jokaisella tavalla on luonnollisesti omat hyvät ja huonot puolensa. Natiivisovellus voi hyödyntää kaikkia laitteen tarjoamia ominaisuuksia, mutta sovellus pitää ohjelmoida jokaiselle alustalle erikseen.

2.4 jQuery Mobile ja Cordova

Puhdas web-sovellus ei voi yleensä käyttää esimerkiksi notifi kaatioita, eli sovellusten antamia viestejä sovelluksen ollessa suljettuna, mutta sovellus ei vaadi asentamista ja ohjelman osasia tai jopa koko ohjelmaa voi käyttää muilla alustoilla [30]. Hybridillä on mahdollisuus käyttää kaikkia laitteen ominaisuuksia, mutta pitää osin ohjelmoida natiivikielellä. Suuren osan hybridisovelluksesta voi tehdä web-tekniikoilla ja siten käyttää uudestaan toisilla alustoilla. Molemmissa voi käyttää hyväksi niin kutsuttuja frameworkkeja, eli kehikkoja, joiden avulla on helpompi tehdä itse sovellusta.

Frameworkkeja, eli valmiita sovellusten käyttöliittymien kehikoita, on paljon ja moneen eri tarkoitukseen, useat liittyvät jollakin tavalla webtekniikoihin. Tämä työ vaatii mobiililaitteisiin tarkoitettuja frameworkkejä, joita on esimerkiksi iUI, jQuery mobile ja Sencha Touch [31]. Tämä työ toteutettiin jQuery mobilella, koska se todettiin hyväksi vaihtoehdoksi jQuerysta saatujen kokemusten perusteella ja jQuery mobile on laajasti käytetty myös muissa sovelluksissa.

jQuery mobile on jQuerysta tehty mobiiliframework [32], joka auttaa luomaan tehokkaan ja tyylikkään mobiilisovelluksen. jQuery mobile antaa oman CSS-tiedostonsa (Cascading Style Sheets), jossa kaikkien elementtien ulkoasu on muokattu yhteneväksi ja siistiksi. jQuery mobilen javascript-tiedosto antaa paljon lisää metodeita varsinkin kosketusnäytön syötteiden käsittelyyn.

Apache Cordova on Oraclen kehittämä hybridisovelluksen välitaso natiivikoodin ja web-sovelluksen välillä [33]. Cordova luo sovellukseen oletuksena selain-ikkunan ja antaa valmiudet käyttää laitteen ominaisuuksia, kuten kameraa ja paikannusta. Cordovan päälle tehty web-sovellus asentuu kuten kaikki muutkin sovellukset ja sovellus itse hallinnoi sisältöään, joten sovellus on käytettävissä ainakin osittain offline-tilassa.

2.5 Selaintietokannat

Eri mobiilialustat tukevat erilaisia tietokantaratkaisuja, eikä internetselaimillekaan ole vielä kaikkien käyttämää standardia. Koska työ tehtiin näiltä osin webtekniikoilla, käydään tässä läpi eri mobiiliselainten tukemat muodot. W3C:n (The World Wide Web Consortium) vuonna 2013 suositama tietokantamalli on indeksitietokanta (IDB, Indexed database) [34], ja se onkin tuettuna Internet Explorerissa versiosta 10.0 lähtien, Mozilla Firefoxissa, Chromessa ja Operassa sekä edellisten mobiiliversioissa [35]. Myös Androidin selain alkaa tukea IDB:tä seuraavassa versiossaan (4.4). Toinen yleisesti käytetty tietokantaratkaisu on Web SQL (Structured Query Language), jota tukevat Chrome, Safari, Opera, Android Browser ja BlackBerry Browser sekä Safarin, Operan ja Chromen mobiiliversiot. W3C lopetti Web SQL:n tukemisen marraskuussa 2010. Kolmas mahdollinen tietokantaratkaisu on hieman vähemmän käytetty, vaikkakin useat pienemmät, erilliset tietokantaratkaisut, kuten CouchDB ja MongoDB käyttävät sitä [36]. Siinä JSON-objektiin (JavaScript object notation) tallennetaan halutut tiedot ja itse objekti tallennetaan Web Storageen, joka on kaikissa HTML5:ttä tukevissa selaimissa.

Indeksitietokantaan tallennetaan tietoa yksittäisiin objektivarastoihin (Object Store), jotka luodaan tietokannan luomisen yhteydessä. Jokaisessa objektivarastossa on yksi avainarvo (Key), jonka on oltava yksilöllinen jokaiselle datariville. Tietoa voidaan hakea tietokannasta joko objektivarastoista tai erityisistä indekseistä (Index), joita voidaan luoda tietokantaa luodessa. Ennen tiedon hakemista pitää luoda transaktio (Transaction)

tietokantaan ja objektivarastoon, josta tietoa ollaan hakemassa sekä pyyntö (Request). Tämän jälkeen tietoa voidaan hakea avainarvoilla tai arvojoukolla (Key Range). Haku avaimella palauttaa yhden arvon, koska avain on uniikki, mutta arvojoukko voi palauttaa useita, jotka voidaan käydä läpi kursorin (Cursor) avulla. Seuraavassa koodiesimerkki, miten indeksitietokannasta voidaan hakea rivi, jonka avain on muuttujan id arvo objektivarastosta ”tracks”.

```
var db = openedDB;  
var trans = db.transaction("tracks", "readonly");  
var store = trans.objectStore("tracks");  
var request = store.get(id)  
request.onsuccess = function (event) {  
    document.getElementById("name").value = event.target.result.name;  
};  
Request.onerror = requestError(event);
```

Ensimmäisellä rivillä alustetaan muuttujaan db sovelluksen indeksitietokanta, joka on jo auki. Toisella rivillä tehdään nopea viittaus objektivarastoon ”tracks” ja ilmoitetaan, että sieltä halutaan vain lukea tietoa. Kolmannella rivillä valitaan viittaus objektivarastoon ”tracks” ja neljännellä haetaan tietokannasta riviä, jonka avainarvo on id:n arvo. Mikäli tiedonhaku onnistuu, kutsutaan 5. rivillä funktiota, joka tulostaa id:tä vastaavan nimen HTML:n tekstikenttään ”name”. Mikäli tiedonhaku epäonnistuu, kutsutaan viimeisellä rivillä requestError-funktiota, jonka tehtävänä on huolehtia ongelmasta.

Indeksitietokannan hyviä puolia ovat tietokantarakenteen nopeus ja tehokkuus, sekä se, ettei indeksikannalla ole teoreettista maksimikokoa. Huonojakin puolia kuitenkin löytyy. Indeksitietokannan rakennetta ei voi päivittää ilman, että uusi versio olisi alustettu. Tietoa ei voi tallentaa kuten relaatiotietokannoissa, vaan yksittäinen objektivarasto on yksittäinen, ja jos haluaa tehdä kyselyitä useampaan varastoon, pitää kyselyt ohjelmoida erikseen. Lisäksi kyselyitä voi tehdä vain asynkronisesti, eli kyselyn tulos ei pala samalle funktiolle, kuin mistä kysely tehtiin. Myös asynkronisuus aiheuttaa helposti suuria ongelmia.

Web SQL on SQLliten pohjalle tehty SQL-tietokanta, jossa tietoa voidaan hallita SQL:n komendoilla. WebSQL:n tukeminen on lopetettu W3C:n puolesta ja WebSQL:n tuki luultavasti häviää tulevaisuudessa selaimista [37]. Syynä lopettamiseen on ollut itsenäisten toteutusten puute, koska suurin osa selaimista käytti SQLiteä Web SQL:n pohjana oman toteutuksen sijaan. Tämän vuoksi standardointi oli vaikeaa, eikä Web SQL voinut edetä standardointiin. Seuraavassa koodiesimerkki, miten Web SQL -tietokannasta voidaan hakea rivi, jonka avain on muuttujan id arvo.

```
var db = openedDB;
db.transaction(function(tx) {
    tx.executeSql('SELECT * FROM tracks WHERE id = ?', [id], showName,
    requestError);
});
```

Ensimmäisellä rivillä alustetaan muuttujaan db sovelluksen Web SQL -tietokanta, joka on jo auki. Toisella rivillä luodaan transaktio. Transaktioita käytetään tässä, jotta SQL-haun mennessä pieleen mahdolliset päivitykset tietokantaan perutaan ja perutaan mahdollisesti tehty tuho. Kolmannella rivillä haetaan tietokannasta riviä, jonka avainarvo on id:n arvo. Mikäli tiedonhaku onnistuu, kutsutaan funktiota, joka tulostaa id:tä vastaavan nimen HTML:n tekstikenttään "name". Mikäli tiedonhaku epäonnistuu, kutsutaan requestError-funktiota, jonka tehtävänä on huolehtia ongelmasta.

JSON on tiedonsiirtoa ajatellen rakennettu tiedonjäsentelyformaatti. JSON-objektiin voi tallentaa JavaScript objekteja avain-arvopareina. Vaikka JSON on tarkoitettu tiedonsiirtoformaatiksi, se ei tarkoita etteikö siihen voisi myös säilöä dataa. JSON on melko yksinkertainen ihmissilmälle ja myös tietokoneiden on helppo tulkita sitä. JSON objektia ei sinällään voi tallentaa Web Storageen, vaan objekti pitää muuttua merkkijonoksi, kuten myös lähetettäessä verkon yli. Käytännössä JSON-merkkijono pitää muokata aina sovelluksen käynnistyessä objektiksi ja aina sovelluksen sulkeutuessa takaisin merkkijonoksi. Seuraavassa koodiesimerkki, miten JSON-objektista voidaan hakea rivi, jonka avain on muuttujan id arvo.

```

for(var i = 0; i <=database.tracks.length;i++){
    if(database.tracks[i].id == id){
        document.getElementById("name").value=database.tracks[i].name;
        break;
    }
}

```

Ensimmäisellä rivillä asetettava for-silmukka toistuu yhtä monta kertaa, kuin JSON-objektissa on ratoja. Toisella rivillä on ehtolause, joka toteutuu, jos toistorakenteen osoittaman radan id on sama, kuin annettu id. Mikäli id on oikea, tulostaa koodi id:tä vastaavan nimen tekstikenttään "name" ja lopettaa toistorakenteen suorittamisen.

JSON-tietokantamallin etuja ovat sen yksinkertaisuus ja jo valmiiksi oikea muoto lähettämiseen JSON-merkkijonona. JSON:ia on helppo tulkita ja muokata. Huonoja puolia ovat kuitenkin, ettei JSON-tietokantaratkaisu ole yleinen ja että JSON-merkkijono pitää aina muuttua objektiksi, jolloin se syö muistista tilaa. Tämän työn sovelluksen vaatima mobiililaitteen tietokanta on kuitenkin niin pieni, ettei muistin loppuminen tule kysymykseen.

2.6 Tietokantojen synkronointi

Laitteen tietokannan ja serverin tietokannan synkronointiin esitellään tässä kaksi melko läheistä vaihtoehtoa, synkronointi JSON:in ja XML:n (Extensible Markup Language) avulla. Molemmissa data lähetetään AJAX:in (Asynchronous JavaScript And XML) POST-metodilla mobiililaitteesta serverille ja serveri vastaa tähän vastausviestillä, jonka mobiililaitteeseen vastaanottaa. Eri tavat eroavat vain muodoissaan [38], ja siinä, että XML:ää lähetettäessä Ajaxin "dataType" asetetaan muotoon "text/xml" ja JSON:na lähetettäessä "dataType" asetetaan muotoon "application/json". JSON pitää myös muuttaa tekstimuotoon ennen lähetystä stringify-funktiolla. Kummankin idea on se, että laite lähettää ensin kaikki viimeisen synkronoinnin jälkeen muutetut tiedot ja serveri vastaa lähettäen myös edellisen synkronoinnin jälkeen muuttuneet tiedot.

XML on melko vanha merkintästandardi, jonka on kehittänyt W3C. XML on hieman HTML:n kaltainen elementeistä koostuva dokumentti, johon voidaan tallentaa dataa

useissa eri muodoissa. XML:n tulkitseminen on melko helppoa sekä ihmiselle, että tietokoneille. JSON taas on yksi Javascript standardiin kuuluva objekti, jonka sisään voi tallentaa objekteja. JSON:in tehokkuus perustuu siihen, että sen sisään voidaan tallentaa myös listoja, joihin voi tallentaa objekteja. Koko objekti voidaan tämän jälkeen muuttaa tekstimuotoon, ja lähettää tekstimuotoisena verkon yli. Tekstimuotoista objektia on myös sekä ihmisen, että tietokoneen, helppo tulkita.

2.7 Samankaltaiset sovellukset

Täysin työtä vastaavaa sovellusta ei tiettävästi ole olemassa, mutta muutama hieman vastaavankaltainen on. V-Orienteering on beta-vaiheessa oleva sovellus, jossa sovelluksen ympärille luotu yhteisö tekee suunnistusratoja ja ratoja voi suunnistaa Google Maps – kartoilla [39]. Radat ovat helppoja ja kartta-aineiston tarkkuuden vuoksi rastipisteet ovat suuria, esimerkiksi risteyksiä. Taulukossa 2 esitellään sovelluksien eroavaisuuksia.

Taulukko 2: Eri sovellusten ominaisuudet

Sovellus	Kartta	Rastin tarkistus	Rastipisteen määrittäminen
Opinnäytetyön toteutus	Suunnistuskartta	Paikannus	Rastipisteellä tai valmiiden koordinaattien avulla
V-Orienteering	Google Maps	Paikannus	Kartalle
GPS Orienteering	Suunnistuskartta	Paikannus	Kartalle
Dip Orienteering Dipper	Suunnistuskartta	NFC / QR	Viemällä rasti paikoilleen
OrientGame	Google Maps	Paikannus	Kartalle
MOBiiliga Orienteeruma	Suunnistuskartta	QR	Viemällä rasti paikoilleen

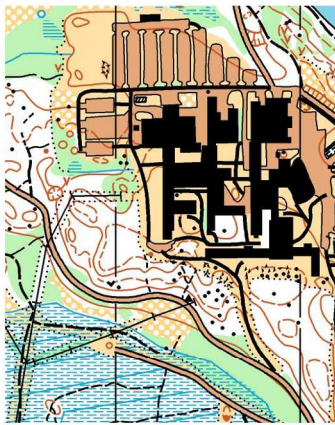
Toinen hieman vastaava sovellus on GPS Orienteering, jonka web-palvelussa voidaan luoda rata lataamalla sovellukseen JPEG-muotoinen (Joint Photographic Experts Group) kartta ja merkitsemällä siihen rastit sekä sovittamalla kartta oikeisiin koordinaatteihin [40]. Dip Orienteering Dipper on sovellus, jossa rastit on korvattu NFC (Near field communication) tageilla tai QR-koodeilla (Quick response) [41]. Sovellus ei ota kantaa

suunnistajan karttaan, koska itse rastia ei kiinnosta suunnistajan sijainti, vaan rasti vain antaa tietoa, kun joku tulee riittävän lähelle.

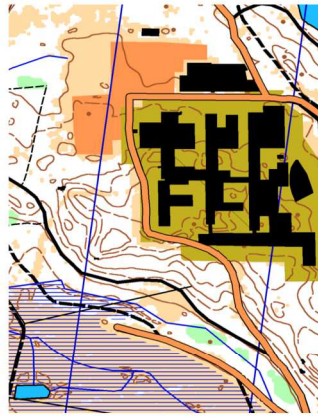
OrientGame on sovellus, jonka sivustolla voi luoda itselleen radan, ja suunnistaa paikannusta apuna käyttäen mobiilisovelluksessa [42]. Ratoja voi lisäksi jakaa Facebookin välityksellä. Karttana käytetään Google Mapsia. MOBiiliga Orienteeruma on sovellus, joka käyttää myös QR-koodeja rasteilla [43]. Tässä sovelluksessa suunnistuskartan saa puhelimeensa, ja karttanäkymässä näkyy myös kompassinuoli.

Internetissä on saatavilla ilmaista karttamateriaalia, kuvassa 2 on esitetty eri karttalähteiden karttoja. Ilmaista karttamateriaalia tarjoavat esimerkiksi Google Maps [44], kartta f, ja maanmittauslaitoksen kehittämä Oskari [45], kartta d. Ilmaiset palvelut eivät kuitenkaan tarjoa riittävän tarkkaa materiaalia suunnistuskartaksi. Lähimmäksi suunnistuskarttaa voidaan päästä Karttapullautin-sovelluksen avulla. Karttapullautin-sovellus osaa tehdä laserkeilausaineistosta, jota on osasta Suomea saatavilla ilmaiseksi, suunnistuskartan pohjan [46], kartta c. Karttapullautin-sovellus ei kuitenkaan osaa määrittää esimerkiksi pienimpiä polkuja, peltoja eikä kiviä. Luotuun karttaan voidaan lisätä Maanmittauslaitoksen tarjoama maastotietokanta, jolloin karttaan saadaan muun muassa talot, pellot, tiet ja vesistöt, kartta b. Siltikään kartassa ei ole vielä esimerkiksi kiviä.

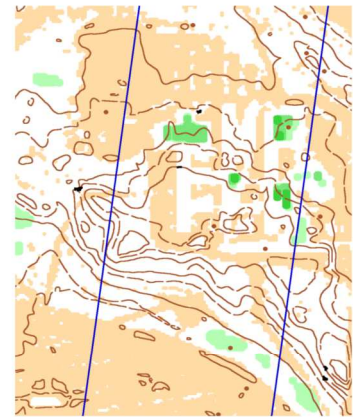
Vertailemalla kuvan 2 karttoja a ja b, eli suunnistuskarttaa ja pullautuskarttaa maastotietokannalla, voidaan havaita tiettyjä eroavaisuuksia. Pullautuskartasta puuttu kokonaan kaikki kivet ja polut, sekä yliopiston sisäpihalla olevat tiet. Yliopiston ympärillä olevat asfaltoidut ja kivetyt alueet eivät ole merkittyinä pullautuskarttaan, vaan koko yliopiston alue on merkitty yksityisalueeksi. Eräs suuri pullautuskartan ongelma on myös se, että kartan korkeuskäyrät ovat tarkkoja. Suunnistajaa ei yleensä kiinnosta korkeuden tarkka, esimerkiksi 5 senttimetrin nousu yli tietyn korkeuden keskellä aukeaa, vaan vaikka aukean reunassa oleva parin metrin nousu. Pullautuskartta merkitsee korkeuskäyrän tarkasti juuri sen oikeaan sijaintiin, kun taas käsin piirretyssä suunnistuskartassa kartoittaja useimmiten tietää, mitä suunnistaja kartalta haluaa, ja osaa sen karttaan piirtää. Tätä kutsutaan yleistämiseksi.



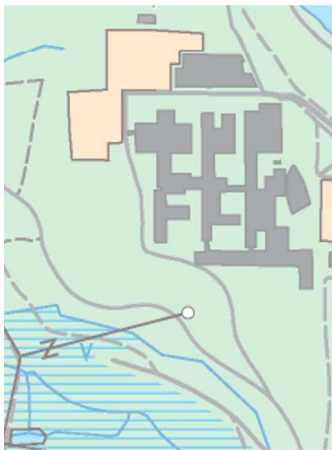
(a)



(b)



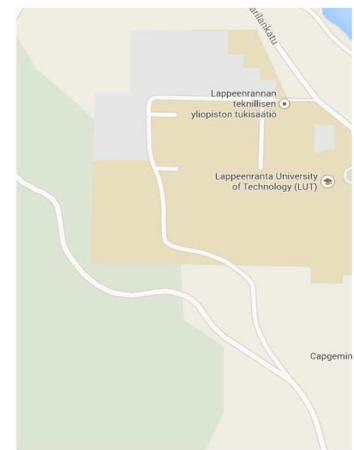
(c)



(d)



(e)



(f)

Kuva 2: Karttojen eroavaisuuksia: a) suunnistuskartta, b) pullautuskartta ja maanmittauslaitoksen maastotietokanta, c) pullautuskartta, d) Oskarin kartta, e) Maanmittauslaitoksen peruskarttarasteri, f) Google Maps-kartta. Kartoista a) ja c) puuttuu yliopiston pohjoispuolelle rakennettu Saimaan ammattikorkeakoulun rakennus.

3 SOVELLUKSEN TOTEUTUS

Sovelluksesta toteutettiin demo, jolla voi luoda ja suunnistaa ratoja sekä synkronoida niitä palvelimelle. Demo ei ole valmis sovellus, mutta se on riittävä osoittamaan, että tämän työn kuvailema sovellus voitaisiin tehdä. Tässä luvussa kerrotaan, miten edellisessä luvussa esiteltyjä asioita sovellettiin käytäntöön, ja miten sovelluksen tekeminen onnistui. Sovellusta verrataan muihin samankaltaisiin sovelluksiin ja sovelluksen antaman paikkatiedon tarkkuutta analysoidaan.

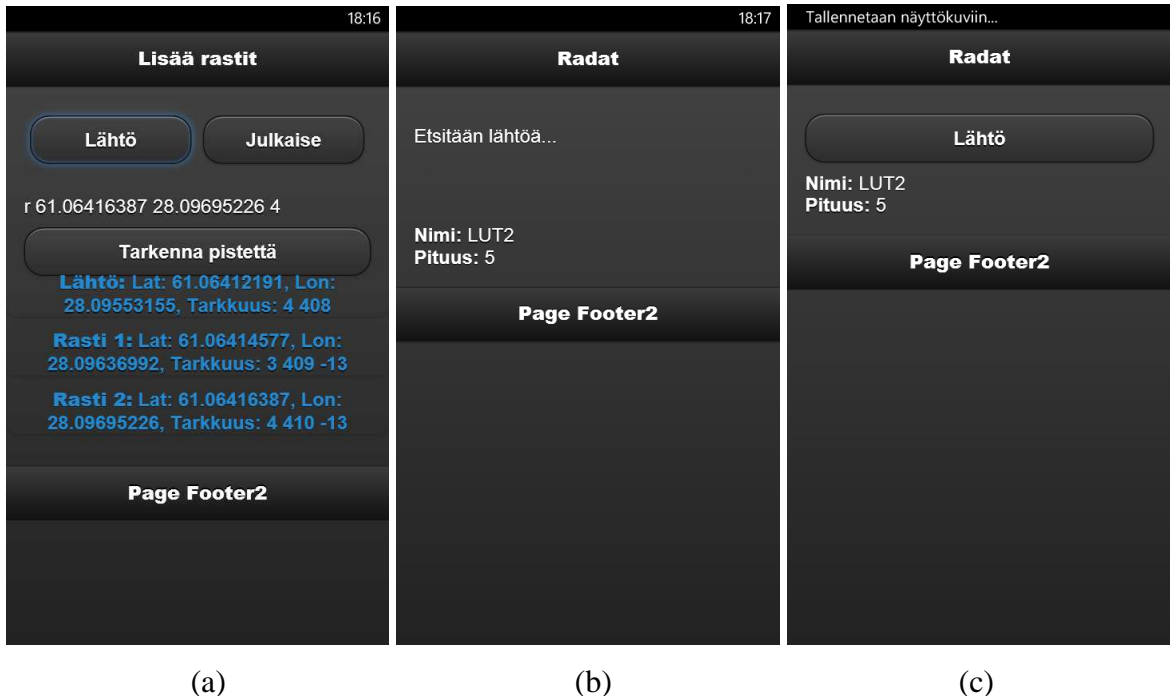
3.1 Paikannus

Suunnistus aiheuttaa paikannukselle haasteita, koska siinä tarvitaan sijainti muutaman metrin tarkkuudella, jotta voidaan olla varmoja, että suunnistaja on oikean rastin pisteessä. Suunnistuksessa maasto vaihtelee aukeista paikoista hyvinkin synkkiin kuusikkoihin ja kallioreunojen läheisyyteen, jossa näkyvyys taivaalle on huono. Tämän vuoksi pitää kyetä arvioimaan riittävän suuri rastiympyrä, jossa suunnistajan pitää olla ollakseen sovelluksen mielestä rastilla. Mikäli suunnistaja on sovelluksen mielestä rastiympyrän ulkopuolella, vaikka olisikin rastipisteessä, tapahtuu ns. false negative, jossa sovellus virheellisesti väittää, ettei suunnistaja ole rastilla. Toisaalta ympyrä ei saa olla liian iso, jolloin käyttäjä voisi olla rastin lähistöllä löytämättä itse rastia ja silti löytää rastin sovelluksen mielestä. Tällaista tapahtumaa kutsutaan false positiveksi, eli suunnistaja on sovelluksen mielestä rastiympyrän sisällä, vaikka olisikin sen ulkopuolella.

HTML5 ja Cordova tarjoavat mobiililaitteeseen paikannuspalvelun, mikäli laite tukee paikannusta. Laitteelta saa leveys- ja pituuspiirin sekä korkeuden lisäksi myös edellisten tarkkuuden, puhelimen suunnan pohjoiseen nähden sekä nopeuden muodossa metriä/sekunti. Ikävä kyllä laitteelta ei saa tietoa esimerkiksi satelliittien määrästä, joilta paikkatietoa saadaan, vaikka tieto olisi hyödyllinen rastiympyröiden koon määrittämisessä dynaamisesti.

Sovellus aloittaa paikannuksen, kun aloitetaan rastipisteiden tallentaminen tai kun käyttäjä valitsee radan, jonka hän haluaa suunnistaa. Paikannus lopetetaan, kun jompikumpi aktiviteeteista lopetetaan. Paikannusta pidetään koko rastipisteiden tallennuksen, tai suunnistuksen ajan päällä, jotta sijainti saadaan mahdollisimman tarkasti, vaikka tämä

käyttääkin enemmän akkua. Kuvassa 3 on esitetty sovelluksen rastien tallennuskuva sekä reitin tiedot sekä lähtöpaikan ulkopuolella, että lähtöpaikan löydyttyä.

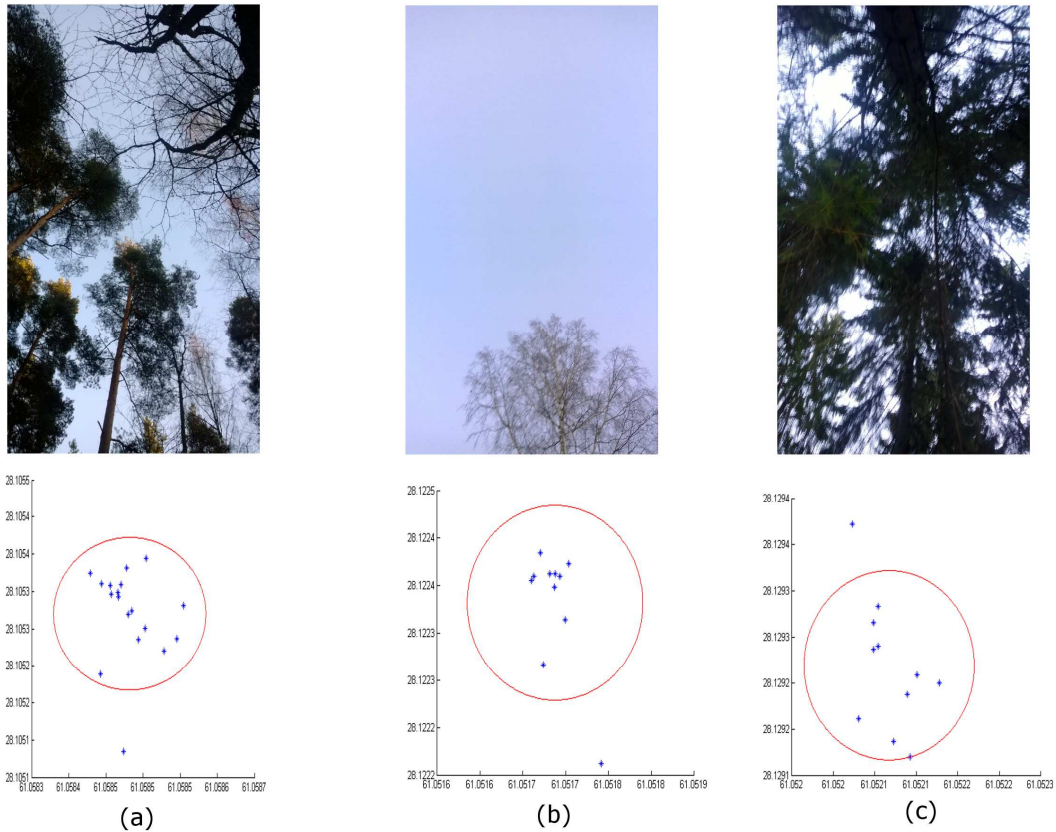


Kuva 3: Kuvaa sovelluksesta, a) rastien tallennus, b) reitin tiedot, lähtöpaikan ulkopuolella, c) reitin tiedot, lähtöpaikalla

Kuvassa 4 esitetään eritiheyksissä metsissä saatuja pisteitä. Pisteitä on tallennettu useana päivänä eri kellonaikoihin. Jokaisessa pisteessä seistiin noin minuutti ennen pisteen tallennusta, jotta laite löytäisi mahdollisimman monta satelliittia. Kuvassa ylhäällä on taivaskuva kohteesta ja alhaalla on esitetty pisteiden hajontakuvio. Alakuvassa olevan ympyrän säde on noin 5 metriä ja siniset pisteet yksittäisiä tallennuspisteitä. Kuvasta voi huomata, että itä-länsisuunnassa pisteet ovat melko hyvin kasassa, mutta pohjois-eteläsuunnassa on enemmän hajontaa. Jokaisessa kuvassa on myös yksi piste, joka ei mahdu kehän sisälle. Nämä pisteet oli tallennettu eri aikoina.

Kohteessa b, aukea, ovat pisteet parhaiten kasassa, kun taas pisteessä c, tiheä metsä, ne ovat eniten hajallaan. Pisteistä voidaan todeta, että rastiympyrän, eli kehän, jonka sisällä käyttäjän pitää olla ollakseen rastilla, säteen pitää olla suurempi, kuin 5 metriä. 10 metrin säteellä kaikki havaintoaineiston pisteet osuisivat ympyrän sisälle. Kuvista voidaan todeta, että rastiympyrän säteen tulee olla vähintään 10 metriä. Tämän kokoinen alue ei tuota

vielä ongelmia suunnistuksellisesti, koska suunnistajan ollessa 10 metrin säteellä rastista hän yleensä on jo löytänyt rastipisteen. Toki pitää muistaa, että piste saattaa heittää myös rastia kohti, jolloin ympyrän ulkopuolella oleva suunnistaja voi sovelluksen mielestä olla rastiympyrän sisällä.



Kuva 4: Tallennuspisteet: a) normaali tiheys, b) aukea, c) tiheä metsä

3.2 Mobiilialusta

Alunperin työ oli tarkoitus tehdä WP7 (Windows Phone 7) alustalle, ja sille tehtiinkin melko valmis demo. Koko sovellus tehtiin natiivikoodina, joten koodin hyödyntäminen muille alustoille, kuin WP8:lle, olisi ollut mahdotonta. Tämän vuoksi työssä siirryttiin webtekniikoihin, joiden hyödyntäminen toisilla alustoilla on huomattavan helppoa. Työtä ei vielä voi käyttää suoraan kuin Windows Phone 8:lla. WP7:n selain ei tue indeksitietokantaa ja Androidin selain alkaa tukea sitä vasta versiosta 4.4, joka julkaistiin 31.10.2013, alkaen. Työn valmistuessa, loppuvuodesta 2013, suunnitelmissa on kuitenkin muuttaa tietokantarakenne indeksitietokannasta JSON:in ja Webstoragen yhdistelmäksi, jolloin sovelluksen tekeminen muille alustoille olisi Cordovan avustuksella huomattavan helppoa.

Sovelluksen asentaminen WP8:lle oli helppoa, kun koko sovellus oli rakennettu Cordovan päälle. Cordova tarjoaa erillisen www-kansion, johon laitetaan kaikki web-sovelluksen osaset. Tämän jälkeen vain määrätään aloitussivu, jotta Cordova osaa avata oikean sivun ensimmäisenä. Myös sovelluksen tekeminen Android 4.4:lle olisi helppoa, koska Android 4.4:n selain tukee indeksitietokantaa. Voisi vain luoda Androidille kopion Cordovasta ja kopioida WP8:n www-kansion sisällön Androidin www-kansion sisälle.

3.3 jQuery Mobile ja Cordova

jQuery mobilen käyttö on helppo oppia, jos on kokemusta jQuery:n käytöstä. jQuery mobile tarjoaa helposti siistin ulkoasun sovellukselle, jolloin demon tekeminen on helppoa. Lisäksi tässä työssä luontevalta tuntui käyttää jQuery mobilen tarjoamaa sivurakennetta, jossa yhdessä HTML-tiedostossa voi olla useita sivuja erotettuina omiin div-tägeihinsä. Seuraavana yksinkertainen esimerkki yhdestä sivusta:

```
<div data-role="page">
  <div data-role="header">Tässä on sivun otsikkopalkki</div>
  <div data-role="content">Tässä on sivun sisältö</div>
  <div data-role="footer">Ja tässä on sivun alapalkki</div>
</div>
```

Kuten esimerkistä huomataan, on sivun rakenne melko yksinkertainen. Keskimmäisen div-tägin sisään voi laittaa kaikkia HTML:n elementtejä ja rakentaa sovelluksen näkymän. Eri sivuja voi vaihdella ilman tarvetta ladata sivuja uudestaan.

Myös Cordova oli melko helppo ja yksinkertainen käyttää. Varsinkin PhoneGap:llä, joka on palvelu, jolla nettisivun voi Cordovan avulla luoda mobiilisovellukseksi [47], on helppo luoda sovelluksia. Palveluun vain kirjaututaan, määritetään paketti, jossa on web-sovellus ja josta halutaan mobiilisovellus, ja päätetään, mille alustoille sovellus halutaan luoda. Muutamassa sekunnissa palvelu luo paketit, jotka voi ladata ja asentaa laitteeseen.

3.4 Selaintietokannat

Sovelluksessa päädyttiin käyttämään indeksitietokantaa, koska Internet Explorer, WP8:n selain, tukee sitä. Kuitenkin koko kehityksen ajan tähän oltiin tyytymättömiä lähinnä indeksitietokannan hankaluuden ja tähän sovellukseen sopimattomuuden vuoksi. Sovelluksen tietokanta on hieman monimutkainen käsiteltäväksi yksinkertaisesti indeksitietokannalla. Tietokantaan ei voi asettaa relaatioita, vaan ne on tehtävä ohjelmallisesti, joka aiheuttaa huomattavasti lisää työtä. Tiedon lisäys, lukeminen ja poistaminen ovat kaikki melko hankalia operaatioita, ja toimintojen asynkronisuus tuottaa hankaluuksia sovelluslogiikkaan. Tietokanta tullaan tulevaisuudessa muuttamaan käyttämään JSON-objektia, joka tallennetaan selaimen Webstorageen. Vain WP8 ja Android 4.4:n ja uudemmat tukevat indeksitietokantaa, kun taas web storagea tukevat kaikki mobiiliselaimet. Tämä mahdollistaa entistä tehokkaamman koodin hyödyntämisen tulevaisuudessa, kun miltei kaikki Javascriptillä kirjoitettu ohjelmakoodi voidaan hyödyntää kaikilla alustoilla.

3.5 Synkronointi

Tietokantojen synkronointi suoritettiin JSON-menetelmällä, joka esiteltiin edellisessä luvussa. JSON koettiin yksinkertaisemmaksi ja nopeammaksi verrattuna XML:ään. Demoluontoisesti kehitettiin myös serverille MySQL-tietokanta sekä ohjelma, joka avustaa tietokantojen synkronisoinnissa. Työn valmistuessa, loppuvuodesta 2013, synkronointi tapahtuu aina, kun käyttäjä painaa synkronointinappia. Julkaistavassa sovelluksessa sovelluksen pitäisi itse huolehtia synkronoinnista tietyin väliajoin. Työn valmistuessa käytetty tapa, jossa tieto edellisestä synkronoinnista säilytetään laitteessa, on hyvä, koska tällöin serverin ei tarvitse huolehtia yksittäisistä laitteista. Vaikka käyttäjä vaihtaisi mobiililaitettaan, osaa sovellus synkronoida aina kaikki tarvittavat tiedot laitteeseen. Myöhemmin, kun sovellukseen on kehitetty myös käyttäjätilit, voi serveri tarkistaa, että lähettää vain käyttäjälle sallitut tiedot, eli käyttäjän tekemien tai ostamien reittien tiedot.

4 POHDINTA JA TULEVAISUUS

Koska sovellus on tällä hetkellä, loppuvuodesta 2013, vielä demoasteella, pitää sitä vielä kehittää. Sovelluksen tärkeimmät toiminnallisuudet toimivat jo, mutta vielä on paljon paranneltavaa ennen kuin sovelluksen voi julkaista. Todennäköisesti jQuery mobilen käyttöä jatketaan, koska se on todettu hyväksi. jQuery mobilella on ollut yksinkertaista tehdä toimiva sovellus. Tässä luvussa käsitellään, mitä sovellukseen olisi vielä ennen julkaisua hyvä lisätä ja millaisilla lisäominaisuuksilla voisi lisätä sovelluksen arvoa käyttäjälle julkaisun jälkeen.

4.1 Samankaltaiset sovellukset

Kaikissa muissa sovelluksissa on jokin tekijä, jonka vuoksi sovellus on erilainen, kuin tämän työn sovellus. Useissa tarkastettiin rastilla olo NFC, tai QR –tekniikoilla. Nämä kuitenkin luovat rastin leimaamiseen usean sekunnin työn. Lisäksi puhelin pitää olla helposti saatavilla jokaisella rastilla. Tämän työn tarkoituksena on saada rastillakäynti mahdollisimman helpoksi, ja sen vuoksi asetuksista voi määrätä, haluaako käyttäjä ilmoittaa sovellukselle, että on rastilla, vai tarkastaako sovellus rastillaoloa koko ajan. Mikäli sovellus tarkastaa suunnistajan rastillaoloa koko ajan, voi sovellus ilmoittaa käyttäjälle rastin löytymisestä.

Monessa sovelluksessa käytettiin Google Maps –karttoja, jotka ovat huomasti suunnistuskarttoja epätarkempia. Epätarkkuuden vuoksi näiden sovellusten rastipisteiden pitää olla helppoja ja selkeissä kohteissa, jotka näkyvät Google Maps –kartoissa. Suunnistuksessa kuitenkin tarkoituksena on laittaa rasteja vaikeisiin paikkoihin, jolloin suunnistustaidot korostuvat.

Osassa suunnistuskartta oli käyttäjän puhelimesta. Vaikka tämän työn sovellukseenkin sellainen vaihtoehto tehdään, on suunnistuksen lähtökohtana kuitenkin paperinen kartta. Puhelimen näytöllä olevassa kartassa on useita ongelmia, joista suurin lienee mittakaava, joka on suunnistajalle tärkeä. Mittakaava muuttuu karttaa tarkennettaessa, ja alkuperäiseen mittakaavaan on vaikea päästä takaisin. Toisaalta pitkällä rastivälillä voi tulla ongelmia näytön resoluution kanssa, koska koko rastiväli ei mahdu kerralla näytölle. Tämä voi olla ongelma suunnistajan reitinvalinnan suunnittelulle.

4.2 Sovellukseen tehtävät muutokset ja lisäykset

Sovellusta tulisi kehittää ja lisätä siihen puuttuvia ominaisuuksia, joita käydään läpi tässä luvussa. Työn valmistuessa sovelluslogiikka on joiltakin kohdilta hieman sekava, esimerkiksi yksittäiseen rataan voi vielä lisätä rasteja julkaisun jälkeen, joten sitä olisi hyvä selkiyttää ja luoda selkeät tavat, joilla esimerkiksi voi luoda sovelluksen nollasta julkaisuun. Sovelluksen olisi hyvä myös laskea automaattisesti radan pituus ja ilmoittaa pituus radan tekijälle viimeistään käyttäjän merkittyä maalin. Laskettu pituuden voisi myös tallentaa serverille, kuten sovelluksen demossa käyttäjältä kysyty radan pituus. Sovellus voisi kysellä käyttäjältä enemmän tietoja tämän luomasta radasta, esimerkiksi liittyen radan noususummaan, eli radalla vaadittava nousemis- ja laskeutumismäärään.

Sovellukseen voisi myös lisätä käyttäjän saamaa tietoa. Työn valmistuessa, loppuvuodesta 2013, käyttäjä ei voi selata serverillä sijaitsevia ratoja, vaan kaikkien ratojen tiedot ladataan aina synkronoinnin yhteydessä laitteelle. Käyttäjän olisi hyvä pystyä selaamaan erikseen jo ostamiaan sekä serverillä olevia ratoja ja tekemään serverin radoista mahdollisen ostopäätöksen. Radat voisivat esiintyä käyttäjälle etäisyyden mukaan, lähin rata ensimmäisenä. Serverillä sijaitsevia ratoja selatessaan käyttäjältä voisi olla poistettuna kyseisestä näkymästä jo ostetut tai suunnitellut radat, jolloin jo suunnitellut radat eivät häiritse toimintaa.

Myöskään ratojen ostoa ei ole vielä toteutettu. Todennäköisesti kauppa toteutetaan niin, että uudella käyttäjällä on tietyn verran ”rahaa” sovelluksessa, jolloin hän voi suunnistaa ilmaiseksi vaikka ensimmäisen radan. Tämän jälkeen käyttäjä voisi ostaa joko yksittäisen radan, tai hieman halvemmalla tietyn summan sovellukseen eri kauppojen tarjoamilla sovellusten sisäisillä ostoilla. Sovellukseen voidaan joko luoda oma rahayksikkönsä, tai käyttää kulloinkin käytössä olevaa valuuttaa. Ostetun reitin voi suunnistaa sovelluksella kerran, ja reitin suunnistamiseksi uudestaan pitää käyttäjän maksaa reitti uudestaan. Tällä tavalla kukaan ei voi huijata järjestelmää siten, että useampi suunnistaisi yhdellä puhelimella vuoronperään.

Koska kompassi on tärkeä apuväline suunnistuksessa, on se hyvä saada myös tähän sovellukseen. Suunnistusnäkyseen saa melko helposti pyörimään kompassin HTML5:den

Geolocation API:lla (Application programming interface). Kompassiin voi myös tehdä ominaisuuden, jossa kompassikuvaa painamalla saa kompassin taustalle jäätyään kuvan kompassin asennosta painallushetkellä. Näin käyttäjä voi ottaa myös suunnan puhelimellaan. Suunnistusnäkyymään voi tehdä aloittelijoita ajatellen vinkkinapin, josta käyttäjä voi saada tietoa seuraavan rastin suunnasta tai matkasta seuraavalle rastille. Pienen vinkin antaminen helpottaa suunnistusta varsinkin aloittelijoilla ja jokaisella on mahdollisuus selvittää vaikeammastakin radasta hyväksytysti, joka taas luonnollisesti kohottaa käyttäjän itsetuntoa ja itsevarmuutta. Toisaalta tuloksissa voidaan esittää vinkkien käyttömäärät, jolloin kukaan ei voi huijata vinkkien avulla nopeimpia aikoja.

Sovellukseen voi myös luoda lisäominaisuuden, jolla voi tallentaa oman reittinsä. Käyttäjä voi halutessaan laittaa asetuksista reitin tallentamisen päälle, ja hän voi myöhemmin tarkistaa reitin sovelluksen internetpalvelusta. Käyttäjä voi myös verrata omaa reittiään vaikka ystävänsä reittiin tai jokaisen rastivälin nopeimman ajan tehneen reittiin ja näin kehittää tietämystään nopeimmasta reitinvalinnasta.

Sovellukseen pitää myös luoda houkutteleva ulkoasu. Sovelluksen nimeksi on luultavasti tulossa Orienttraining. Sovelluksen ulkoasussa voi käyttää suunnistajalle tuttuja termejä ja symboleita, esimerkiksi radan suunnistamisen aloittamista voi lähdön lisäksi kuvata kartoissa esiintyvällä lähdön symbolilla, joka on esitetty kuvassa 5. Symbolien käytöllä voisi osittain tehdä sovelluksesta universaalia, koska symbolit ovat samanlaiset kaikille suunnistajille, kielestä riippumatta. Sovelluksen ulkoasussa pitää pelkän ulkoasun lisäksi ajatella myös sitä, että sovellusta voidaan käyttää osittain vaikka kotisohvalla ja osittain maastossa.



Kuva 5: Suunnistuskartan ratasymboleita

Edellä käytyjen lisäysten lisäksi sovellukseen voi toteuttaa vaikka matka- ja suuntataitopelit, joissa sovellus kertoo, että käyttäjän pitää kulkea vaikka sata metriä tai suoraan pohjoiseen. Matkan kuljettuaan käyttäjä ilmoittaa tästä sovellukselle, joka laskee matkan ja suunnan ja informoi tästä käyttäjää. Näin suunnistaja voi harjaannuttaa erikseen näitä suunnistuksessa tarvittavia taitoja.

4.3 Selainsovellus

Sovellukseen kuuluu tärkeänä osana myös internet-sivut, joissa voi tarkastella reittivaihtoehtoja, ostaa niitä sekä selailta omia tuloksiaan ja verrata niitä muiden tuloksiin. Työn valmistuessa, loppuvuodesta 2013, niitä ei vielä ole olemassa, koska kandidaattityöhön rajattiin mobiilisovelluksen tekeminen. Tämä luku käsittelee, mitä kyseisillä nettisivuilla tehdään ja mitä sinne voi sovelluksen julkaisun jälkeen tehdä.

Tärkein tehtävä sivuilla on tietenkin eri reittivaihtoehtojen selailu ja osto sekä tulosten esittäminen erilaisin tavoin. Selailu ja osto tapahtuu samoin periaattein, kuin mobiilisovelluksessakin, toki käyttäjän suurempaa näyttöä käytetään mahdollisimman tehokkaasti hyödyksi. Reiteistä voidaan ilmaista väliajat sekä keskimääräisiä nopeuksia. Lisäksi voidaan esimerkiksi esittää käyttäjän kulkema reitti, mikäli sovellus reitin on tallentanut. Käyttäjän reittejä voidaan myös vertailla esimerkiksi tietyn rastivälin nopeimpaan reittiin, jolloin käyttäjä saa tärkeää tietoa, mikä olisi ollut nopeampi reitti ja miten kannattaa seuraavalla kerralla valita. Toki paras reitinvalinta on yksilöllinen sekä jokaiselle radalle ja rastivälille, kuin myös jokaiselle suunnistajalle riippuen esimerkiksi juoksukunnon ja suunnistustaidon tasapainosta. Käyttäjälle siis luotaisiin palvelu, jossa hän voisi verrata omia tuloksiaan muiden tuloksiin.

Tärkeä apuväline reittien tekijöille olisi, jos he voisivat tuoda OCAD:ssä tehdyn radan rastipisteet suoraan tekeillä olevaan reittiin. Useat kartat vuonna 2013 tehdään paikannuksen avulla, ja jokaisen pisteen tarkka sijainti tiedetään. Tämän vuoksi luodaan selainsovellukseen toiminto, jolla käyttäjä voisi tuoda reitin rastipisteet suoraan OCAD:stä käymättä itse pisteillä. Tämä helpottaisi huomattavasti radantekijän työtä ja pienentäisi kynnystä tehdä lisää ratoja.

Nykyaikaiset älypuhelimet maksavat huomattavia summia. Vaikka normaali käyttäjä voisikin ottaa sellaisen suunnistusradalla mukaan, olisivat kilpasuunnistajat silti haaste. He tuskin haluaisivat ottaa riskiä kalliin älypuhelimien rikkoutumisesta usein kivikkoisessa tai kosteassa maastossa. Toisaalta huippusuunnistajilla kuitenkin on usein jonkinlainen reitintallennuslaite mukanaan, koska he haluavat dataa omasta suorituksestaan. Internetsovellukseen olisikin hyvä tehdä toiminto, jolla jo ostetun radan reitin voisi tallentaa selainsovellukseen. Tallennuksen jälkeen sovellus tarkistaisi, onko rasteilla käyty. Tallennusmahdollisuus vähentäisi riskiä rikkoa kallis älypuhelin maastossa, mutta samalla myös kilpasuunnistajat voisivat harjoitella ja leikkimielisesti myös kilpailla harjoittellessaan toisiaan vastaan ja saada tarkat tiedot eri rastien rastiväliajoista. Myös internet-sivuilla pitää huomioida samankaltainen ulkoasu, kuin mobiilisovelluksessa. Suunnistusteemaa voi käyttää sielläkin, ja painikkeiden tekstien lisäksi voi käyttää myös suunnistuksesta tuttuja kuvia ja symboleita.

Seuroille voisi myös luoda oman yhteisöpalvelun, jossa seuran aktiivit voisivat jakaa mielipiteitään, jokainen seura omalla, erillisellä, kanavallaan. Yhteisöpalvelun voisi yhdistää tämän työn sovellukseen siten, että seuran aktiivit voisivat myös jakaa tekemiään reittejä vain omille jäsenilleen niin, ettei jokaisesta suunnistetusta radasta tarvitsisi maksaa. Esimerkiksi valmentaja voisi tehdä nuorille radan, ja nämä voisivat käydä suunnistamassa reitin silloin, kun parhaiten ehtisivät. Samalla valmentaja voisi saada tärkeää tietoa jokaisen reitinvalinnoista ja väliajoista ja käyttää näitä harjoistusten suunnittelussa. Tietenkin voidaan myös analysoida erilaisia reitinvalintoja ja niiden nopeuksia, kun dataa on tarpeeksi. Tällaista tietoa voitaisiin myydä eteenpäin erilaisille tahoille, joiden intresseihin kuuluvat paremmat suunnistajat ja tehokkaampi valmennus.

YHTEENVETO

Työssä luotiin suunnistussovelluksen demo. Täysin vastaavaa sovellusta ei vielä tiettävästi ole, vaan kaikki samankaltaiset sovellukset joko toimivat suunnistuskarttoja selvästi heikkolaatuisemmilla kartoilla, tai niiden rastina toimii pisteessä oleva QR-koodi tai NFC-tägi. Suunnistuskarttoja vastaavaa materiaalia ei ole ilmaiseksi saatavilla. Tässä sovelluksessa pisteen sijainti tiedetään muutaman metrin tarkkuudella ja työssä havaitaan, että rastiympyrän säteen tulisi olla vähintään 10 metriä, riippuen paljon rastipisteestä, käytettävästä tekniikasta, sekä siitä, missäpäin maapalloa sovellusta käytetään.

Työ tehtiin Windows Phone 8:lle. Siinä käytetään jQuery mobile- mobiiliframeworkkiä, Cordovaa, joka on välitaso natiivisovelluksen ja web-sovelluksen välillä sekä laitteen tietokantana IndexedDB:tä, eli web-indeksitietokantarakennetta. Käytetty indeksitietokanta tullaan tulevaisuudessa vaihtamaan WebStorageen tallennetuksi JSON-objektiksi sen helppokäyttöisyyden vuoksi. Tuloksena on demo sovelluksesta, jolla voi luoda reittejä, suunnistaa niitä ja synkronoida reittejä serverille. Sovellus on esitys siitä, miten teknologia voi auttaa suunnistustaitojen kehittämisessä.

Lopuksi työssä käydään vielä läpi, mitä sovellukseen voisi tehdä ennen julkaisua, ja mitkä ominaisuudet voisivat tuoda sovellukselle lisäarvoa julkaisun jälkeen. Sovelluslogiikka on vielä melko sekava, ja sitä tulisi selventää. Käyttäjien pitäisi kyetä kirjautumaan sovellukseen, jolloin käyttäjistä ja heidän suorituksistaan voisi pitää kirjaa. Käyttäjä voisi tallentaa sovellukseen rahaa ja ostaa sillä reittejä. Suunnistussivulla voisi olla suunnistuksessa tärkeä kompassi. Käyttäjälle voisi myös luoda mahdollisuuden tallentaa kuljettu reitti.

Internet-sivuja ei tämän työn valmistuessa, loppuvuodesta 2013, vielä ole, mutta myös nettisivuilla pitäisi pystyä selaamaan ja ostamaan reittejä sekä analysoimaan tuloksia ja väliaikoja ja vertailemaan niitä muiden suorituksiin. Ratoja olisi hyvä myös pystyä syöttämään suoraan OCAD-muodossa, jolloin jo ennestään tunnettuja koordinaatteja ei tarvitsisi hakea enää maastosta. Suunnistajat voisivat ilahtua myös ominaisuudesta, jolla reitin voisi syöttää suoraan laitteesta, joka tallentaa reitin. Lisäksi seuroille voisi tehdä yhteisöpalvelun, jossa voisi omalle seuralleen jakaa tekemiään reittejä.

LÄHTEET

1. Piedallu, Christian, ja Jean-Claude Gégout. ”Effects of Forest Environment and Survey Protocol on GPS Accuracy”. *Photogrammetric Engineering & Remote Sensing* 71, nro 9 (1. syyskuu 2005): 1071–1078. doi:10.14358/PERS.71.9.1071.
2. ”About Orienteering: International Orienteering Federation”. Viitattu 26. marraskuu 2013. <http://orienteering.org/about-orienteering/>.
3. He, Xiaozhi. ”Application of Geographic Information System in Orienteering Sports”. Teoksessa *Business, Economics, Financial Sciences, and Management*, 637–642. Springer, 2012. http://link.springer.com/chapter/10.1007/978-3-642-27966-9_86.
4. ”SSL-Suomen Suunnistusliitto”. Viitattu 22. marraskuu 2013. <http://www.ssl.fi/ssl/sslwww.nsf/sp2?open&cid=content2F334A>.
5. ”Tietoja tapahtumasta | Jukolan viesti”. Viitattu 26. marraskuu 2013. <http://www.jukola.com/tietoja-tapahtumasta/>.
6. ”VDST: Was ist Orientierungstauchen?” Viitattu 26. marraskuu 2013. <http://www.vdst.de/tauchen-sport/orientierungstauchen/was-ist-orientierungstauchen.html>.
7. ”Negev Desert Championship 2011 | Israel Orienteering”. Viitattu 26. marraskuu 2013. <http://israelorienteering.org/championship-events/desert-championship-2011/>.
8. Zentai, László. ”Implementation of Cartographic and Digital Techniques in Orienteering Maps”. Teoksessa *Cartography from Pole to Pole*, toimittanut Manfred Buchroithner, Nikolas Prechtel, ja Dirk Burghardt, 19–29. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. http://link.springer.com/10.1007/978-3-642-32618-9_2.
9. ”OCAD Inc. - Home”. Viitattu 26. marraskuu 2013. <http://www.ocad.com/en/>.
10. Miettinen, Samuli. *GPS vie vaivatta perille: ohjekirja luonnossa liikkujille*. Helsingissä: Ajatus, 1999.

11. Poutanen, Markku. *GPS-paikanmäärittäminen*. Helsingissä: Tähtitieteellinen yhdistys Ursa, 1998.
12. "GPS constellation status for 11/26/2013". Viitattu 26. marraskuu 2013. <http://www.navcen.uscg.gov/?Do=constellationStatus>.
13. "Lockheed Martin · Global Positioning System (GPS)". Viitattu 26. marraskuu 2013. <http://www.lockheedmartin.com/us/products/gps.html>.
14. "NIS GLONASS / Main page / GLONASS / History of GLONASS development". Viitattu 26. marraskuu 2013. http://www.nis-glonass.ru/en/glonass/development_history/.
15. "Nokia ja Glonass sopivat satelliittipaikannuksesta | It-viikko". Viitattu 26. marraskuu 2013. <http://www.itviikko.fi/ratkaisut/2011/11/18/nokian-alypuhelimiin-tulee-venalainen-gps/201117217/7>.
16. "Galileo - What do we want to achieve ? - Satellite navigation - Enterprise and Industry". Viitattu 26. marraskuu 2013. http://ec.europa.eu/enterprise/policies/satnav/galileo/index_en.htm.
17. "BeiDou Navigation Satellite System — Home". Viitattu 26. marraskuu 2013. <http://en.beidou.gov.cn/index.html>.
18. "Welcome To Indian Space Research Organisation - Home Page". Viitattu 26. marraskuu 2013. <http://www.isro.org/index.aspx>.
19. Braasch, M.S. "Performance comparison of multipath mitigating receiver architectures". 3:3/1309–3/1315. IEEE. Viitattu 16. marraskuu 2013. doi:10.1109/AERO.2001.931361.
20. Pinker, Aron, Charles Smith, ja James Day. "Wide-Area Augmentation System (WAAS) – The Metamorphosis of a Major FAA Program". *GPS Solutions* 3, nro 4 (huhtikuu 2000): 48–57. doi:10.1007/PL00012815.
21. Taverna, Michael A. "Egnos In The Air." *Aviation Week & Space Technology* 172.29 (2010): 45. *Business Source Complete*. Web. 26. marraskuu 2013.
22. International Symposium on Distributed Computing and Artificial Intelligence, ja Juan Manuel Corchado. *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*. Berlin: Springer, 2009. <http://public.eblib.com/EBLPublic/PublicView.do?ptiID=417437>.

23. Allen, Sarah, Graupera, Vidal, ja Lee Lundrigan. *Pro Smartphone Cross-Platform Development iPhone, BlackBerry, Windows Mobile, and Android Development and Distribution*. [New York, N.Y.]: Apress : Distributed to the Book trade worldwide by Springer Science + Business Media, 2010. <http://public.eblib.com/EBLPublic/PublicView.do?ptiID=645524>.
24. ”Global mobile statistics 2013 Part A: Mobile subscribers; handset market share; mobile operators | mobiThinking”. Viitattu 27. marraskuu 2013. <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#smartphoneos>.
25. ”The Smartphone OS Complete Comparison [Chart]”. Viitattu 27. marraskuu 2013. <http://myphonedeads.co.uk/blog/33-the-smartphone-os-complete-comparison-chart>.
26. Jordan, Lucas L, ja Pieter Greyling. *Practical Android Projects*. [New York]: Apress : Distributed by Springer Science+Business Media, 2011. <http://dx.doi.org/10.1007/978-1-4302-3244-5>.
27. ”Apple - iOS 7”. Viitattu 27. marraskuu 2013. <http://www.apple.com/ios/>.
28. Eddy, Nathan. "Android, Apple Lead Smartphone OS Market Share: IDC." *Eweek* (2013): 1. *Academic Search Elite*. Web. 27. marraskuu 2013.
29. CAMPBELL, JAY. "A Note On Mobile Apps: Web, Native And Hybrid." *Business Travel News* 30.6 (2013): 26. *Business Source Complete*. Web. 27. marraskuu 2013.
30. Cordeiro, José, ja Karl-Heinz Krempels. *Web Information Systems and Technologies 8th International Conference, WEBIST 2012, Porto, Portugal, April 18-21, 2012, Revised Selected Papers*. Berlin, Heidelberg: Springer, 2013.
31. ”20 Useful Mobile Frameworks for Developers | Code Geekz”. Viitattu 27. marraskuu 2013. <http://codegeekz.com/mobile-frameworks-for-developers/>.
32. ”jQuery Mobile | jQuery Mobile”. Viitattu 27. marraskuu 2013. <http://jquerymobile.com/>.
33. ”Apache Cordova”. Viitattu 27. marraskuu 2013. <http://cordova.apache.org/>.
34. ”Indexed Database API”. Viitattu 27. marraskuu 2013. <http://www.w3.org/TR/IndexedDB/#object-store>.

35. "Can I use... Support tables for HTML5, CSS3, etc". Viitattu 27. marraskuu 2013. <http://caniuse.com/>.
36. Lennon, Joe. *Beginning CouchDB: [starting Apache CouchDB, the Open Source Database for the Cloud]*. New York, NY: Springer, 2009.
37. "Web SQL Database". Viitattu 27. marraskuu 2013. <http://www.w3.org/TR/webdatabase/>.
38. Goncalves, Antonio. "XML and JSon Processing". Teoksessa *Beginning Java EE 7*, Antonio Goncalves, 387–416. Berkeley, CA: Apress, 2013. http://link.springer.com/10.1007/978-1-4302-4627-5_12.
39. "Virtual Orienteering - Find your way!" Viitattu 24. toukokuu 2013. <http://www.vorienteering.com/home>.
40. "GPS Orienteering - Android-sovellukset Google Playssa". Viitattu 27. marraskuu 2013. <https://play.google.com/store/apps/details?id=se.hippsoomapp.gpsorientering&hl=fi>.
41. "dib - Orienteering Dibber". Viitattu 27. marraskuu 2013. <http://www.appindesign.com/dib/>.
42. "OrientGame - Android-sovellukset Google Playssa". Viitattu 27. marraskuu 2013. <https://play.google.com/store/apps/details?id=com.orientgames&hl=fi>.
43. "MOBO". Viitattu 27. marraskuu 2013. <http://mobo.osport.ee/>.
44. "Google Maps". Viitattu 27. marraskuu 2013. <https://www.google.fi/maps/preview#!data=!1m4!1m3!1d149222!2d28.1431493!3d61.0088989>.
45. "Avoimien aineistojen tiedostopalvelu". Viitattu 27. marraskuu 2013. <https://tiedostopalvelu.maanmittauslaitos.fi/tp/kartta>.
46. "Karttapullautin - toolbox/workflow for generating O training maps without field checking". Viitattu 27. marraskuu 2013. <http://routegadget.net/karttapullautin/>.
47. "PhoneGap | Home". Viitattu 27. marraskuu 2013. <http://phonegap.com/>.