

Kari Sainio

# Älypuhelimella käytettävän suunnistussovelluksen laajennus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinööriytyö

27.11.2013

Tekijä Otsikko	Kari Sainio Älypuhelimella käytettävän suunnistussovelluksen laajennus
Sivumäärä Aika	45 sivua + 1 liite 27.11.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	
Ohjaaja	lehtori Peeter Kitsnik
<p>OrientGame on älypuhelimille suunniteltu suunnistussovellus. Insinööriyön tavoite oli kehittää Android-älypuhelimelle tehtyä aikaisempaa versiota sovelluksesta uusilla ja parannetuilla ominaisuuksilla sekä laajentaa toiminnallisuutta tekemällä www-pohjainen suunnistusratojen suunnitteluohjelma.</p> <p>Työssä määriteltiin uudet vaatimukset sovelluksen laajennusta varten sekä Android- että www-sovellukselle. Vaatimusten perusteella valittiin käytettävät ohjelmistoteknologiat ja perusteltiin niiden käyttö. Toteutuksen pääalueita esitettiin kuvilla ja lähdekoodiesimerkeillä.</p> <p>Insinööriyön tuloksena syntyi laajennettu versio Android-älypuhelimille tarkoitettusta OrientGame-sovelluksesta sekä sen käyttöä tukeva www-sovellus. Määritellyt tavoitteet saavutettiin hyvin. Lopputuloksia on visualisoitu erilaisilla kuvaruutukaappauksilla. Lisäksi käsitellään mahdollisia jatkokehityskohteita sekä muutoin suunnittelun ja toteutuksen onnistumista.</p> <p>Laajennettu versio OrientGame-sovelluksesta on ollut Google Play-sovelluskaupassa ladattavissa maailmanlaajuisesti elokuusta 2013 lähtien. Älypuhelinsovellukseen liittyvä www-sovellus on ollut myös käytössä samanaikaisesti.</p>	
Avainsanat	Suunnistus, Web, PHP, Android, Java, Google Maps, Facebook

Author Title	Kari Sainio Extension of Orienteering Application for Smartphones.
Number of Pages Date	45 pages + 1 appendix 27 November 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	
Instructor	Peeter Kitsnik (Lecturer)
<p>OrientGame is a smartphone application for orienteering. The purpose of the project described in this thesis was to extend the capabilities of this Android application by adding new features and developing a special Web application which is for designing orienteering tracks.</p> <p>The new requirements were specified for both an Android application and a Web application. On the basis of these requirements, proper technologies were selected. The essentials of implementation principles are described in the thesis using pictures and source code examples. Subsequently, an extended version of the Android application and also the supporting Web application were developed.</p> <p>The goals were achieved well. New versions of the Android application and Web application were developed. Both applications had the Facebook user authentication integrated and they enabled the user to create their own orienteering tracks. In addition, some further ideas for development were also considered. These are using other authentication methods than Facebook and consideration of creating tracks on the Android device itself.</p> <p>The extended version of the OrientGame application has been available on the Google Play store since August 2013. The corresponding Web application has also been available since then.</p>	
Keywords	Orienteering, Web, PHP, Android, Java, Google Maps, Facebook

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Toiminnallisuuden määrittely	5
2.1	Suunnistuksen määritelmä	5
2.2	Haluttu toiminnallisuus	6
2.3	Teknologiavalinnat	8
2.4	Www-sovelluksen rakenne	9
2.5	Android-sovelluksen rakenne	13
3	Www-sovelluksen toteutus	17
3.1	Käyttäjätunnistuksen toteutus	17
3.2	Kartta- ja paikkatiedot	21
3.3	Tietokannan sovituserros ja Ajax-liityntä	26
3.4	Www-sovelluksen pääsivu	27
4	Android-sovelluksen toteutus	30
4.1	Sovelluksen rakenteelliset muutokset	30
4.2	Facebook-tunnistuksen toteutus	31
4.3	Facebook-kaveriluettelon hakeminen	35
4.4	Karttatiedon hakeminen	35
4.5	Muut muutokset	37
5	Testaus	39
6	Julkaisu	40
6.1	Web-hotelli	40
6.2	Google Play -kauppa	41
7	Yhteenveto	43
	Lähteet	45
	Liitteet	
	Liite 1. Esimerkki KML-tiedostosta	

## Lyhenteet

AJAX	<i>Asynchronous JavaScript And XML</i> ; Web-teknologia, joka mahdollistaa asiakassovelluksen asynkroniset kutsut Web-sovellukselle.
API	<i>Application Programming Interface</i> ; sovelluksen ohjelmointirajapinta.
DAL	<i>Data Abstraction Layer</i> ; ohjelmitava kerros, jolla peitetään alimman tason tiedonkäsittelyyn tarkoitettu toiminnallisuus.
Eclipse	Avoimeen lähdekoodiin perustuva integroitu ohjelmistokehitysympäristö.
GPS	<i>Global Positioning System</i> ; Amerikkalainen satelliittipohjainen järjestelmä, joka tarjoaa paikka ja -aikatietoja.
GLONASS	<i>Globalnaya Navigatsionnaya Sputnikovaya Sistema</i> ; Venäläinen satelliittipohjainen järjestelmä, joka tarjoaa paikka ja -aikatietoja.
HTML5	HTML5-kieli on uusin versio verkkosivujen tekemiseen yleisesti käytetystä HTML-merkintäkielestä.
HTTP	<i>Hyper Text Transfer Protocol</i> ; Internetissä käytössä oleva yleisin tiedonsiirtoprotokolla. Se pohjautuu Request/Response –sanomarakenteeseen.
JavaScript	JavaScript on yleinen enimmäkseen Web-selaimissa käytetty tulkkaava ohjelmointikieli, jonka rakenne perustuu Java-ohjelmointikieleen.
JSON	<i>JavaScript Object Notation</i> ; avoimen standardin mukainen tekstipohjainen ja rakenteellinen tiedon esitystapa.
KML	<i>Keyhole Markup Language</i> ; XML-merkintäkieleen perustuva notaatio, jolla voidaan ilmaista erilaisia kartoissa näkyviä merkintöjä, kuten paikkoja ja reittejä sekä visualisoida niitä Internet-pohjaisilla kaksi- ja kolmiulotteisilla varta vasten tehdyillä selaimilla.

OAuth	<i>Open standard for Authentication</i> ; tunnistusmenetelmä, jolla käyttäjä voi kirjautua luotettuun järjestelmään ja sallia kolmannelle osapuolelle oikeuden tiettyihin resursseihin tarvitsematta käyttäjän tunnusta ja salasanaa.
PHP	<i>Personal Home Page</i> ; yleinen Web-palvelimissa suoritettava tulkettava ohjelmointikieli- ja ympäristö.
URL	<i>Universal Resource Locator</i> ; www-osoite.
REST	<i>Representational State Transfer</i> ; HTTP-protokollaan perustuvien Web-palveluiden spesifikaatio.
SDK	<i>Software Development Kit</i> ; ohjelmistokirjasto, jonka avulla voidaan kehittää lisätoiminnallisuuksia.
SHA1	<i>Secure Hash Algorithm 1</i> ; 160-bittinen hash-algoritmi, joka korvaa nykyään MD5:n. Se muodostaa syötetiedosta 20-tavuisen hash-arvon.
XML	<i>Extensible Markup Language</i> ; rakenteellinen merkintäkieli.

## 1 Johdanto

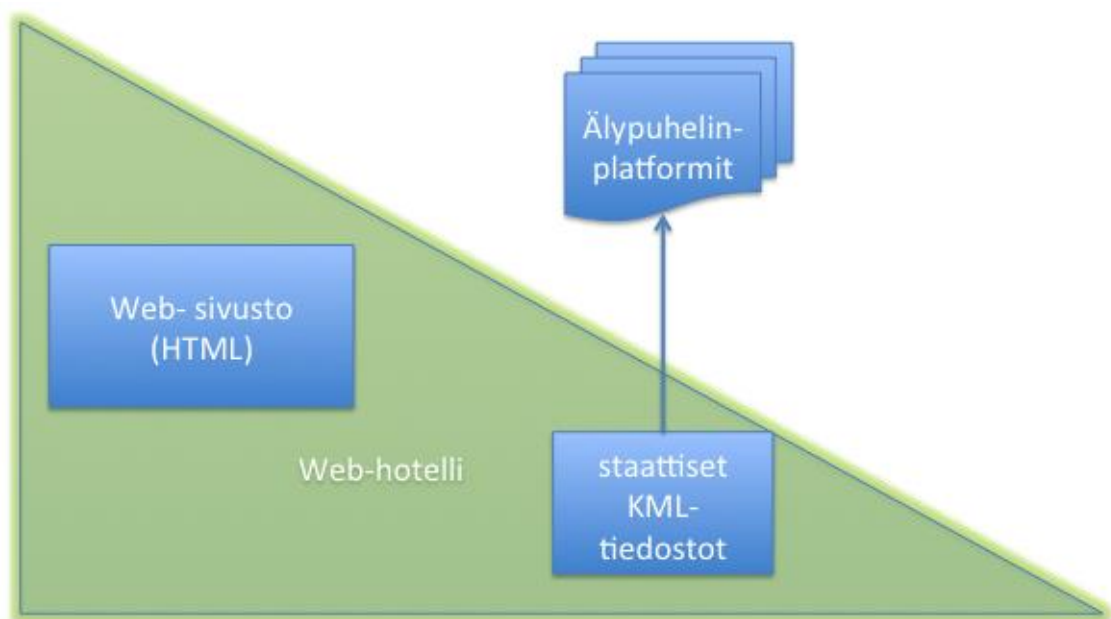
Suunnistus on yksilölaji, jossa kukin suunnistaja etsii kartalle ennakoita merkityt rastit itsenäisesti käyttäen apuna painettua tai tulostettua suunnistuskarttaa ja kompassia. Se perustuu pitkälti yksilön taitoon, kokemukseen ja kykyyn hahmottaa kolmiulotteinen maasto ja tulkita siitä kartalla näkyvä vastaava kaksiulotteinen projektiio. Suunnistuskartta on hyvin tarkka ja itse suunnistettava rata rasteineen tyypillisesti tulostetaan tai piirretään kartalle ennen suunnistukseen ryhtymistä. Nykyaikaiset älypuhelimet ja niiden sisältämät sähköiset kartat mahdollistavat eräänlaisen sähköisen suunnistamisen. Älypuhelin sisältää yleensä myös GPS-pohjaisen navigointi- ja paikannustoiminteen. Jossain malleissa on käytettävissä jopa magneettikentän tunnistava sensori, jota voidaan soveltaa sähköisenä kompassina. Nykyisten älypuhelimien valmiit kartat eivät yllä tarkkuudeltaan suunnistuskartan tasolle, mutta tarjoavat kuitenkin hyvän ja ilmaisen mahdollisuuden hyödyntää sen ominaisuuksia suunnistamiseen. Lisäksi laitteet ovat tyypillisesti kytkettynä Internetiin matkapuhelinverkon kautta, mikä mahdollistaa suunnistusratojen jakelun laitteisiin helposti. Lisäksi niiden tarjoaman sosiaalisen median verkostoa voidaan hyödyntää erilaisessa tiedottamisessa. Älypuhelimien tarkoitettu suunnistuksen mahdollistama sovellus voidaan myös jakaa helposti älypuhelin-alustan kauppapaikassa kaikille halukkaille.

Insinööriyön tarkoituksena on kehittää uusi versio olemassa olevasta älypuhelimissa toimivasta suunnistussovellusta nimeltä OrientGame sekä kartuttaa kokemusta Web- sekä älypuhelinohjelmiston kehityksestä ja yleensäkin kokeilla konseptin toimivuutta älypuhelimissa. Koska nykyiset älypuhelimet tarjoavat suhteellisen hyvän ja maailmanlaajuisesti kattavan karttasovelluksen, voidaan globaalisuutta hyödyntää ohjelmiston jakelussa sekä käytössä ympäri maailman. Lisäksi älypuhelimet tarjoavat oivan mahdollisuuden kytkeä suunnistusreittien ja jopa mahdollisen tulosseurannan jakelun Internetin kautta tapahtuvaksi. OrientGame-sovellus on tarkoitettu lähinnä muille kuin varsinaista suunnistusta harrastaville henkilöille mahdollistamaan helpon tavan tutustua lajiin tarjoten melko uudenlaisen käyttökokemuksen.

Lähtötilanne

OrientGame-älypuhelinsovellus on kehitetty ensimmäisenä iPhone-alustalle. Myöhemmin sen ohjelmisto on konvertoitu Android- ja Windows Phone -alustoille. Sovelluksen

käyttöliittymä on pyritty pitämään suhteellisen samana kaikilla alustoilla, mutta siinä on vivahdetta iPhone-käyttöliittymästä, koska versio on kehitetty ensimmäisenä. Tarkoitus ei kuitenkaan ole ollut kehittää esimerkiksi Sports Trackerin kanssa kilpailevaa urheilu-sovellusta. Sovellus tarjoaa käyttäjälle valmiita julkisia suunnistusratoja ympäri maailman, jotka on tallennettu web-palvelimelle. Käyttäjällä on ollut mahdollisuus oman radan lataamiseen. Tällöin hänen on täytynyt tehdä valmis rata käyttäen KML-notaatiota sekä asentaa tiedosto johonkin Internetin web-palvelimelle, josta se on voitu sovelluksen avulla hakea. KML-tietorakenne on vastaava, jota muun muassa Google Earth -ohjelmisto käyttää. Tietorakenteessa kuvataan eri paikkatietoja ja niiden suhteita.



Kuva 1. Lähtötilanteen arkkitehtuuri.

Kuvasta 1 voidaan nähdä, että lähtötilanteen arkkitehtuuri on hyvin yksinkertainen sekä staattinen. On ollut www-sivusto, joka on sisältänyt kuvauksen OrientGame-sovelluksesta, linkit kunkin älypuhelinsovelluksen lataussivuille sekä varsinaiset KML-tiedostot. Vain järjestelmän ylläpitäjä on voinut tuoda suunnistusratoja järjestelmään tallentaen ne tiedostoina palvelimelle käytetyn web-hotellin ylläpito-ohjelmiston avulla. Tiedostot ovat olleet myös staattisia, eli muutokset ratoihin ovat vaatineet aina uuden tiedoston tuonnin palvelimelle ja vanhan poistamisen. Yleensäkin on puuttunut sovelluksen käyttäjälle tarvittava suunnittelutyökalu, jolla KML-tiedostoja tai yleensäkin soveltuvia suunnistusratoja voitaisiin luoda helposti. Tarjolla on ollut julkisia työkaluja

esimerkiksi Google Earth, jolla KML-tiedostoja on voinut luoda. Nämä eivät sellaisenaan ole olleet suoraan käyttökelpoisia. Käyttäjän on lisäksi tarvinnut ymmärtää KML-tiedoston rakenne (ks. liite 1), joka sellaisenaan ei ole kauhean ymmärrettävä ja kuvainnollinen.

### Tavoitteet

Työn kohteena olevaa OrientGame-sovellusta (kuva 2) laajennetaan lisäämällä siihen verkkopalvelu, joka hyödyntää sovelluksen käyttäjän (suunnistajan) sosiaalisen median verkostoa ja mahdollistaa suunnistusharrastuksen jakamisen sosiaalisen median välityksellä helpolla ja mainostettavalla tavalla. Tämä toteutetaan siten, että käyttäjillä tulee olemaan mahdollisuus luoda itse suunnistusratoja ja jakaa niitä kisailtavaksi omassa rajatussa kaveripiirissään. Kuka tahansa voi suunnitella radan, mutta vain määritellyillä ystävillä on pääsy ratatietoihin. Tällä halutaan suojella sitä, että erilaiset tilaisuudet eivät kasva liian isoiksi ylimääräisten osallistujien halutessa suunnistaa kyseessä olevaa rataa. Tarkoitus on tarjota mahdollisuus myös eräänlaiseen sosiaaliseen ja leikkimieliseen kilpailmiseen omassa piirissä.



Kuva 2. Kuvaruutukaappaus lähtötilanteen Android-sovelluksen kotinäkymästä.

Työn alueeseen sisältyy tarvittavat Web-sovelluksen luominen sekä muutokset älypuheliin ladattavaan sovellukseen. Esimerkkinä tässä työssä tullaan käyttämään Android-ympäristöä, mutta tarkoitus on tuottaa vastaava toiminnallisuus myös iPhone-

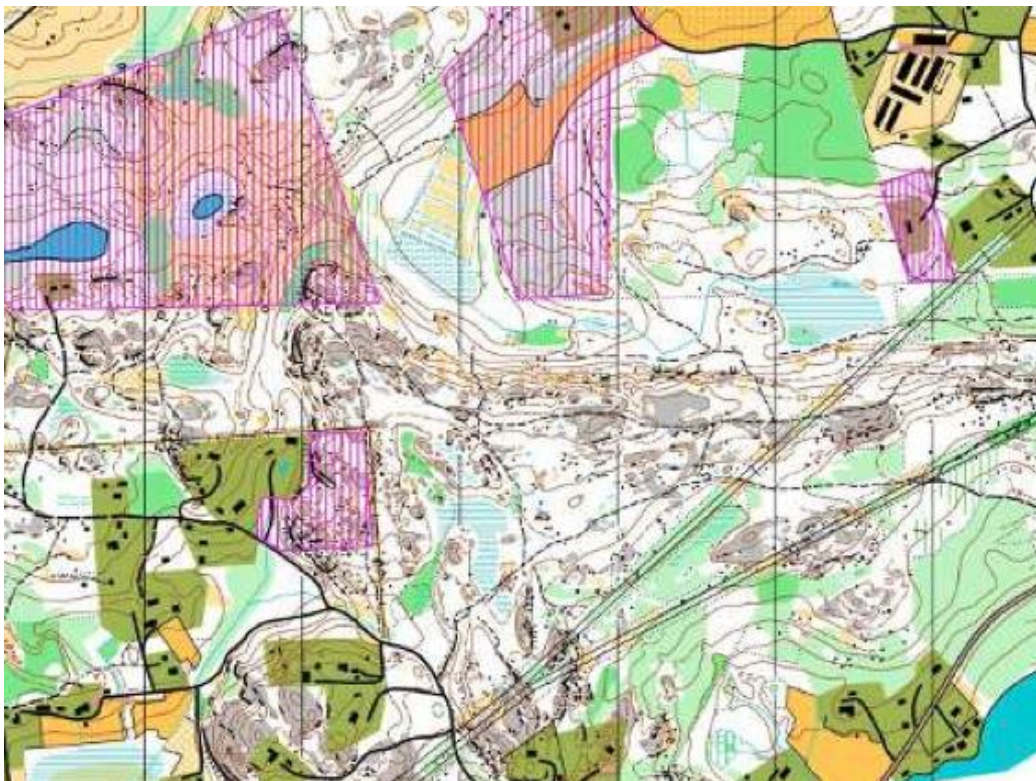
ja Windows Phone -laitteisiin, joiden toteutusten kuvaus rajataan pois tästä insinööri-työstä. Työssä keskitytään Android- ja www -sovelluskehitykseen sekä niihin liittyviin teknologioihin.

Tavoitteen saavuttamiseksi määritellään tarvittava toiminnallisuus, perustellaan teknologiavalinnat sekä käydään läpi toteutusta oleellisilta osilta. Lopuksi luodaan lyhyt katsaus tavoitteen onnistumisesta sekä pohditaan mahdollista jatkokehitystä.

## 2 Toiminnallisuuden määrittely

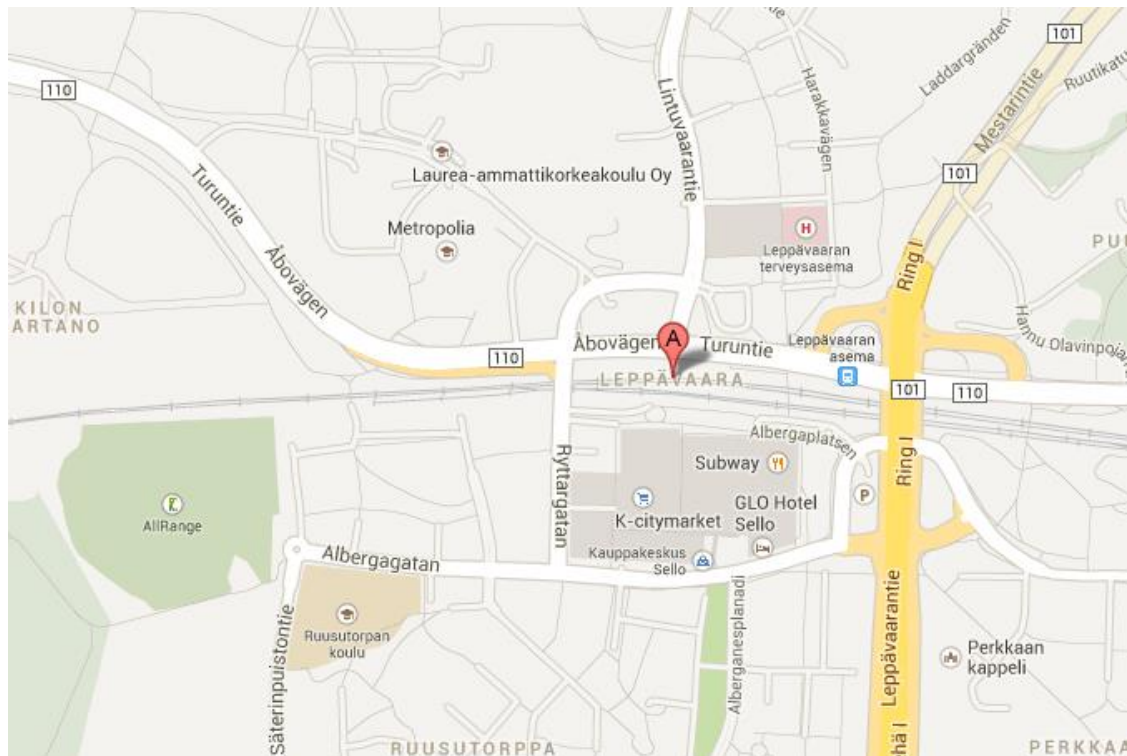
### 2.1 Suunnistuksen määritelmä

Suunnistus on kuntoilu- ja urheilulaji, jossa kuljetaan maastossa tarkoituksena löytää karttaan merkityt rastit oikeassa järjestyksessä mahdollisimman nopeasti edeten. Maasto voi olla rakentamatonta luontoa, esimerkiksi metsää, tai rakennettua puistoa ja taajamaa. Kartan lisäksi suunnistamisen apuna käytetään usein kompassia [1]. Kilpa- ja harrastesuunnistuskartat (kuva 3) on tyypillisesti tehty kiinteän mittakaavan mukaisiksi, ja ne vastaavat suhdetta 1:10 000. Tämä tarkoittaa sitä, että kartalla yksi senttimetri vastaa luonnossa 100 metriä. Suunnistuskartta pyrkii kuvaamaan luonnon sen tarjoamien maamerkkien perusteella. Kartoissa yksityiskohdat onkin kuvattu hyvin tarkasti. Esimerkiksi suuret kivet, jyrkänteet, kumpareet ja muut rakenteet löytyvät omina piirustusmerkkeinään kartalta. Suunnistuskartta sisältää myös ns. korkeuskäyrän, joka kuvaa maaston kolmiulotteisuuden kaksiulotteisessa kartassa.



Kuva 3. Ote suunnistuskartasta [11].

Tässä työssä käytettävässä suunnistus-sovelluksessa ja sen vaatima kartta-aineisto pohjautuu julkiseen karttatietoon, jota ylläpitävää muun muassa Google. Googlen tarjoama karttatieo ei sellaisenaan sovellu tarkkuussuunnistukseen kuten varsinaisessa suunnistuksessa käytetyt suunnistuskartat [2]. Googlen ynnä muita sähköisten karttojen tarkkuuksissa (kuva 4) on lähtökohtana ollut autoihin soveltuva navigointi. Näiden sovelluksen käyttäjältä ei odoteta kompassinkäyttötaitoja. Sovelluksessa tapahtuva suunnistus on yhdistelmä autonavigaation perustuvasta kartasta, mutta ilman navigointiohjeita. Suunnistus vaatii kuitenkin alkeellisen kartanlukutaidon, koska kartta-aineisto on hyvin suurpiirteinen eikä tyypillisesti sisällä maaston kuvausta asutusten ulkopuolella. Tämä suunnistus muistuttaa enemmän korttelisuunnistusta, jossa maasto on helppokulkuista, suuripiirteistä ja on kuvattu yleisimmillä karttamerkeillä kuten teillä, rakennuksilla ja suurimmilla kävelypoluilla.



Kuva 4. Ote Google Maps -kartasta [12].

## 2.2 Haluttu toiminnallisuus

Älypuhelin näytöt ovat suhteellisen pieniä (lukuun ottamatta uudempiä isompia Apple iPad- ja Android Tab -tyyppisiä laitteita) ja niissä radan suunnittelu on vaikea-

ta, koska radan pituus voi helposti ylittää näytölle rajatun kartta-alueen. Yleensäkin näytölle ei mahdu kovin isoa osaa kartasta. Lisäksi sormen käyttö osoittimena tarkkuutta vaativassa piirtämisessä on hankalaa. Radan suunnittelussa perinteinen tietokone hiirineen on tarkempi väline mahdollistamalla suuren näytön hyödyntämisen ja hiirellä tapahtuvan tarkkuuseditoinnin. Suunnittelu halutaan mahdollistaa myös erilaisilla tietokoneilla mallista ja käyttöjärjestelmästä riippumatta.

Työn vaatimukset voidaan määritellä seuraavasti:

- Suunnistusratoja pitää pystyä luomaan kenen tahansa, jolla on pääsy Internetiin ja käytössä tietokone.
- Sovelluksen käyttäjän pitää pystyä lataamaan omia sekä kavereiden ratoja.
- Ratatietojen jakelu perustuu suljetun kaveripiiriin käyttöön. Tehtyjä ratoja ei välttämättä haluta jakaa koko maailmalle.
- Palvelun pitää toteuttaa yksinkertainen tulospalvelu, jossa suunnistamiseen käytetty aika voidaan tallentaa palveluun muiden suunnistajien nähtäväksi mahdollistaen kilpailemisen. Tämä vaatimus ei ole pakollinen.
- Älypuhelinohjelmisto perustuu olemassa olevaan OrientGame-versioon, koska se tarjoaa valmiin suunnistamiseen sopivan alustan hyödynnettäväksi.
- Suunnistusradat pitää pystyä suunnittelemaan web-selaimessa, jolloin käytössä on suurempi näyttö kuin älypuhelimessa.
- Kaverit voidaan valita sosiaalisen median avulla.
- Karttatieto on julkisesti saatavilla olevaa käsittäen koko maailman mahdollistaen sovelluksen käytön globaalisti.
- On toteutettavissa yleisille älypuhelinlustoille (ensisijaisesti Android) sekä saatavilla julkisesti (esim. Google Play) mahdollistaen suuren kattavuuden älypuhelinlinten käyttäjissä.

- Käytössä on kaupallinen web-hotellin palveluntarjoaja tai virtuaalipalvelin ja sen rajaama teknologia.
- Halutaan käyttää olemassa olevaa verkko-osoitetta.

### 2.3 Teknologiavalinnat

Kehitettävä sovellus koostuu www-sovelluksesta sekä älypuhelimessa olevasta asiakassovelluksesta. Kyseessä oleva www-sovellus sijoitetaan kaupalliseen web-hotelliin, joka käyttää tällä hetkellä rekisteröityä osoitetta: [www.orientgames.com](http://www.orientgames.com).

Web-hotellin käyttö tarjoaa kustannustehokkaamman sekä ylläpidoltaan helpomman ympäristön verrattuna vuokrattuun virtuaalipalvelimeen, jota myös harkittiin. Virtuaalipalvelin vaatii käytännössä täyden palvelinohjelmiston asennuksen, konfiguroinnin sekä varusohjelmiston ylläpitoa. Web-hotellissa nämä asiat ovat tehty valmiiksi, ja palvelun pystyttämisessä voidaan keskittyä itse sisällön tuottamiseen. Web-hotellin palveluntarjoaja huolehtii tyypillisesti uusimman ohjelmiston päivityksestä, erilaisen tilastotiedon välityksestä sekä perustietoturvan takaamisesta. Insinööriyössä web-hotellin ylläpito on jätetty pois, koska käytännössä valmiit www-sivut yms. sisältö voidaan siirtää helposti tiedonsiirto-ohjelmistolla palvelimeen ja sisältö on heti käytettävissä eikä se näin ollen sisällä teknisesti mitään uutta. Palveluntarjoajan tilastopalveluita käytetään hyödyksi, jotta nähdään, miten palvelua käytetään sekä voidaan havaita mahdollisia tietoturvaongelmia.

Nykyään on saatavilla uudenaikaisia ja tehokkaita web-sovelluskehitysjärjestelmiä kuten:

- Vaadin [3] mahdollistaa sovelluksen käyttöliittymän kehityksen täysin palvelimeen ohjelmoidun Java-pohjaisen frameworkin avulla.
- Node.js [4] mahdollistaa palvelinohjelmiston kehityksen käyttäen hyväksi JavaScriptiä mahdollistaen käytön sekä web-selaimessa että palvelimessa.
- Muita erilaisia Web-julkaisualustoja (esim. Drupal ja Joomla).

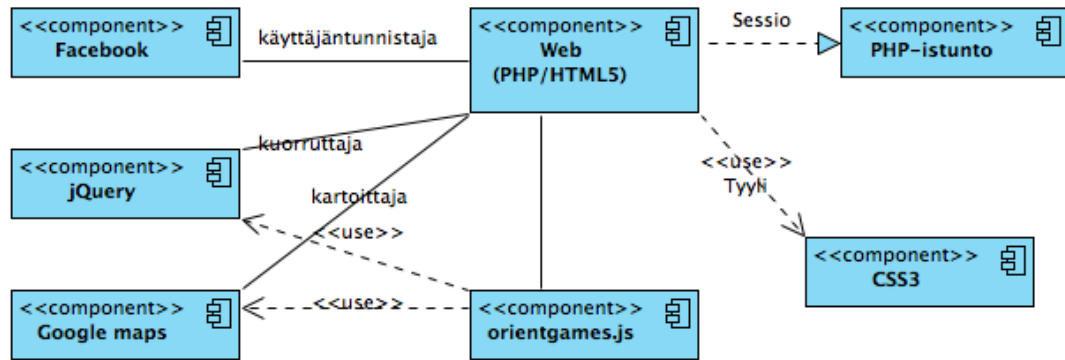
Vaikka uudemmat teknologiat ovat mielenkiintoisia, ei niiden käyttöä päästy kuitenkaan hyödyntämään, koska tarjolla oleva kaupallinen web-hotelli mahdollistaa käytännössä vain PHP-kielisen palvelinsovelluksen tekemisen.

Vartenotettaviksi älypuhelinlustoiksi voidaan tätä kirjoittaessa lukea Android-, iPhone- sekä Windows Phone -ekosysteemit. Kuten aikaisemmin todettiin, keskitytään työssä kuvaamaan Android-alustalla tehtävää sovellusta. Android on yksi eniten käytössä olevista älypuhelinlustojärjestelmistä ja mahdollistaa nopean sovelluskehityksen Java-ohjelmointikieltä käyttäen. Google tarjoaa myös sisäänrakennetun karttatuensa lisensoimille Android-valmistajille. Sokerina pohjalla on halpa kehitysympäristö (joka on käytännössä ilmainen) ja Googlen Play -sovelluskauppa, jonka kautta sovelluksen jakaminen eri maissa on helppoa. Android-sovelluskehitys on helppo aloittaa, koska sen vaatima kehitysympäristö on saatavilla ilmaiseksi kehittäjä sivustolta.

## 2.4 Www-sovelluksen rakenne

Www-sovelluksessa pyritään käyttämään yleisesti käytettyjä ja hyväksi havaittuja menetelmiä. Koska HTML5/CSS3-tuki on nykyaikaa, on sen tarjoamia mahdollisuuksia pyritty hyödyntämään käyttämällä sen sisällön määrittäviä uusia elementtejä: `<footer>`, `<header>`, `<nav>` ja `<section>` sekä CSS3-tyylimuotoilua. Lisäksi käytetään JavaScript-ohjelmointikieltä sekä yleisesti saatavilla olevia kirjastoja, kuten jQuery [8] ja Google Maps API.

Www-sovellus rakennetaan kuvan 5 osoittamista komponenteista. Web-sovellus muodostuu yhdestä PHP-kielisestä pääsivusta (`index.php`), jonka rakenne perustuu HTML5-mukaiseen määrittelyyn. Www-palvelin tulkitsee `index.php`-nimisen sivun pääsivuksi automaattisesti, ja näin ollen käyttäjän ei sovellusta käyttäessä tarvitse tietää URL-osoitteessa sivun nimeä, vaan voidaan viitata suoraan `www`-osoitteeseen [www.orientgames.com](http://www.orientgames.com). PHP-kielinen pääsivun toteutus mahdollistaa myös PHP-pohjaisen istunnon hallinnan. Www-istunto halutaan toteuttaa, jotta käyttäjä tunnustetaan eikä hänen tarvitsisi jokaisella kerralla kirjautua järjestelmään uudelleen. Koska Facebook-kirjaston avulla toteutetaan varsinainen käyttäjän tunnistus, täytyy rakentaa oma erillinen toiminnallisuus sille, että tämä käyttäjätunnistus voidaan sitoa `www`-istuntoon.

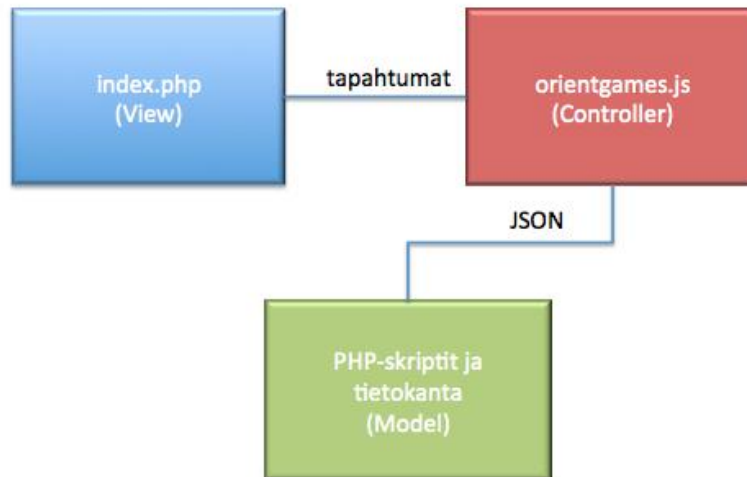


Kuva 5. Www-sovelluksen komponentit.

Www-sovelluksen pääsivu (index.php) pidetään yksikertaisena, ja sen tarkoituksena on vain tarjota pääasiallisesti sovelluksen käyttöliittymä sekä tarvittava sidos eri komponenttien välillä. Sovelluksessa käytetään hyväksi jQuery-kirjastoa yksinkertaistamaan koodia lähinnä Ajax-kutsujen osalta sekä tarjoamaan valmiita käyttöliittymäkomponentteja. Google Maps -komponentti on valmis JavaScriptillä toteutettu karttatoiminnallisuuden tarjoama komponentti, joka voidaan upottaa käytetylle www-sivustolle. Www-sivun tyyli määritellään erillisellä CSS3-tyylitiedostolla.

Www-sovelluksen arkkitehtuuri perustuu löyhästi ns. MVC-malliin, jossa rakenne on eroteltu Model-View-Controller-osioihin kuvan 6 osoittamalla tavalla. MVC-mallissa eri kerrokset ovat eroteltu toisistaan kutsurajapinnoilla ja kullakin kerroksella on karkeat tehtävät. View-kerros määrittelee käyttöliittymän näkymän ja painikkeet, Controller-kerros tulkistaa View-kerroksen tapahtumia ja toimii linkkinä Model-kerrokselle, jossa varsinainen tietorakenne on struktuurisessa muodossa. MVC-mallin ylläpito on kerrosten johdosta helpompaa kuin monoliittisen ohjelman ja jokaista kerrosta voidaan käsitellä erillisinä osina. Kuvassa 6 View-kerroksen toteuttaa pääsivu (index.php), joka siis huolehtii käyttöliittymän päivytyksestä sekä selaintapahtumien (esim. hiiren napinpainallukset) välittämisestä Controller-kerrokselle, joka toteutetaan omana JavaScript- ja jQuery-kirjastokokoelmana (orientgames.js). Tämä kirjasto toteuttaa käyttöliittymän käyttäjän valitsevat tapahtumat sekä alustaa tarvittavat kontrollioliot. Controller-osio pitää sisällään myös osan Model-kerroksesta, kuten käyttäjän tekemän suunnistusradan rastipisteet sekä Google Maps-välisen kommunikoinnin. Model-kerroksen pääasiallinen tarkoitus on toimia linkkinä tietokannan ja muistinvaraisen tietomallin välillä ja huolehtia näin ollen tarvittavien metodien avulla tiedon tallennuksesta ja hakemisesta

tietokantaan. Tiedonvälitys Controllerin ja Model-osioiden välillä hoidetaan JSON-tietorakennetta käyttäen. Model-kerros toteutetaan täysin PHP-kieltä käyttäen jakamalla se tarvittaviin Ajax-operaatioilla kutsuttaviin PHP-aliohjelmiin.



Kuva 6. Www-sovelluksen MVC-malli.

### Käyttäjätunnistus

Tärkeä tietoturvallisuuteen liittyvä vaatimus on, että käytännössä lähes minkä tahansa sovelluksen tarvitsee tukea käyttäjän tunnistusta [6], jotta tietoihin pääsy voidaan rajata ja sallia se vain sen oikealle omistajalle. Tällaisen toiminnallisuuden toteutus on vaativaa ja tarvitsee hyvän perehtymisen asiaan. Lisäksi eri käyttäjätietojen tallennus järjestelmään voi vaatia erilaisten kansallisten tietoturvallisuusmääräysten toteuttamisen tai mahdollisen henkilökisterin. Työn lähtökohtana ei ole keskittyä liiaksi käyttäjähallintaan vaan käyttää valmiina olevia ja julkisesti saatavilla olevia teknologioita. Nykyisessä sosiaalisessa mediassa on käytettävissä erilaisia järjestelmiä, kuten Facebook, Twitter ja Google+, jotka kaikki tarjoavat käyttäjätunnistuspalvelua. Tyypillisesti nämä kaikki perustuvat OAuth-menetelmään, jossa käyttäjän ei tarvitse luovuttaa omia tunnistustietojaan (käyttäjätunnus ja salasana) kolmannelle osapuolelle. Facebook ja sen tarjoama käyttäjätili tarjoaa olemassa olevista järjestelmistä valmiin käyttäjätunnistuksen sekä hyvän kytköksen sosiaalisen median käyttöön. Lisäksi se on tuotavissa eri älypuhelinjärjestelmiin ja www-palveluihin. Sen tarjoama ohjelmointirajapintaa (API) käytetään tässä työssä sovelluksen todennuksen toteuttamiseen. Facebook tuo mukanaan sovellukselle automaattisen valmiin sosiaalisen kaveripiirin. Tämä helpottaa oh-

jelmiston toteutusta, koska vaatimuksen mukainen rajattu käyttäjäryhmää halutaan käyttää ratojen jakelussa. Tätä toiminallisuutta ei tarvitse nyt itse tehdä vaan se voidaan hoitaa Facebookin tarjoamana valmiina palveluna. Koska palveluntarjoja mahdollistaa PHP-kielen käytön web-hotellissa, valitaan Facebook-tilin integrointiin sen PHP-kielinen ohjelmistokehityskirjasto (SDK). Lisäksi aikaisemmin todetun PHP-istunnon toteutus saadaan näin helpoiten integroitua Facebook-tunnistukseen.

### Karttatiedot ja Google Maps

Julkisia karttatietoja on saatavilla muun muassa Googlelta ja Microsoftilta, jotka kattavat lähes koko maailman. Työssä on päädytty käyttämään valmista Google Maps -kirjastoa, joka tarjoaa oivallisen JavaScript API:n www-sovellusten käyttöön [7]. Google Maps on myös hyvin tuettuna Android-sovelluksille. Lisäksi Google Maps -kirjastossa hyödynnetty paikkatietoinformaatio on käytettävissä myös muissa järjestelmissä, koska informaatio on kuvattu yleisillä maantieteellisillä koordinaateilla. Tätä hyödynnetään myös suunnistusratojen kuvauksessa, jossa käytetään XML-kieleen perustuvaa yksinkertaistettua KML-kieltä [13]. Liitteessä 1 on kuvattu esimerkkinä tiedosto, jota Orient-Game-sovellus hyödyntää suunnistusratojen jakelussa. KML-kuvauskieli mahdollistaa lisätä kartalle ns. Overlay-komponentteja, joilla voidaan karttaan sijoittaa lisäinformaatiota esimerkiksi reiteistä, alueista tai yleensä mielenkiintoisista paikoista. Nykyään erilaiset liikenne- ja säätiedot voidaan sijoittaa kartalle käyttämällä KML-kuvauskieltä.

### Tietokantaliityntä

Kuten aikaisemmin mainittiin, toteutetaan Web-sovelluksen Model-kerros PHP-ohjelmointikielellä. PHP-soveltuu hyvin vastaanottamaan Ajax-pyyntöt ja toimimaan linkkinä tietokantaan. Erilaiset Apache Tomcat -pohjaiset Java-toteutukset voidaan sulkea pois käytettävistä ohjelmointitekniikoista, koska näitä ei web-hotellin palveluntarjoaja tue. PHP on suhteellisen vanhaa teknologiaa web-sovelluksissa, mutta se on osoittautunut olevansa vielä nykyäänkin varteenotettava, ja pitkän historian turvin sillä on hyvä tuki erilaisten kirjastojen käytölle [7]. PHP:lle löytyy valmiina hyvä tuki tietokanta- ja Facebook-integraatioon. Se soveltuu hyvin web-sovelluksen käyttöön ja toteuttamaan Model-kerroksen toiminnallisuutta.

Palveluntarjoajalla on käytettävissä MySQL- tai PostgreSQL-tietokannat. Näiden kesken on ehkä makuasia, kumpaa halutaan käyttää, koska molemmat perustuvat re-

laati tietokantamalliin ja tarjoavat oivallisen PHP-tuen. Työssä on valittu käyttöön näin ollen MySQL, koska se on ollut markkinoilla jo kauan ja muutenkin yleisesti käytössä. Kantojen rakenteessa pyritään olemaan yleisen SQL-kieliselä tasolla ja välttämään MySQL:n ominaispiirteitä, jotta tietokanta on siirrettävissä helposti myös muille tietokanta-alustoille. PHP-kielillä toteutettavat tietokantafunktiot ovat helposti korvattavissa PostgreSQL- tai jopa dokumenttipohjaisen MongoDB-tietokannan tarvitsemilla funktioilla.

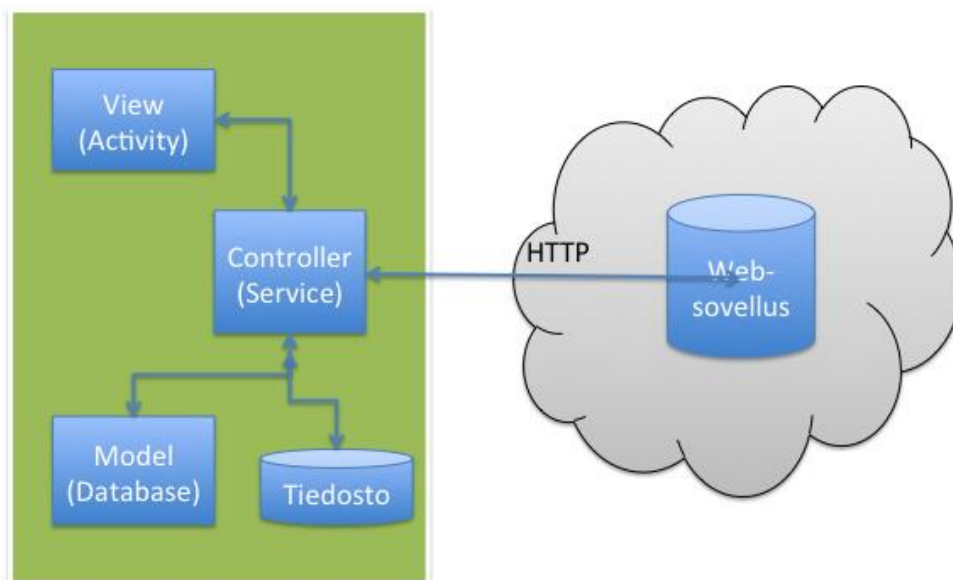
## 2.5 Android-sovelluksen rakenne

Älypuhelinsovelluksessa keskitytään Android-alustaan ja sen uusimpaan versioon 4.x [10], joka on ollut saatavilla jo pitkään. Tämän työn kirjoitushetkellä on versio 4.2 jo saatavilla. Koska vanhoja älypuhelimia on markkinoilla vielä paljon sekä nykyinen Oriengtame-versio tukeutuu Android-versioon 2.3, tutkitaan myös mahdollisuutta tukea sovellusta myös vanhemmassa Android-versiossa 2.3. Lähtötilanteen Android-sovellus on luokiteltu Androidin Google Play -kaupassa Sports-kategoriaan. Kuten taulukosta 1 havaitaan, löytyy vanhemmalla Android versiolla 2.3 vielä noin 26 % käyttäjiä Sports-kategoriassa. Android-maailmassa ei käyttöjärjestelmäpäivityksiä tueta yleensä kovin pitkään. Tämä tarkoittaa sitä, että noin 3-vuotta vanhoja Android-puhelimia on vieläkin markkinoilla paljon ja niissä on vanhemmat käyttöjärjestelmäversiot. On siis hyvä, että sovellus on käytettävissä mahdollisimman monessa laitteessa.

Taulukko 1. Google Play -kaupan urheilu-sovellusten käyttämät Android-versiot 09/2013 [16]

TOP 10 ANDROID VERSIONS FOR SPORTS	
<b>Android 4.1</b>	<b>38.98%</b>
<b>Android 2.3.3 - 2.3.7</b>	<b>25.98%</b>
<b>Android 4.0.3 - 4.0.4</b>	<b>17.41%</b>
<b>Android 4.2</b>	<b>9.73%</b>
<b>Android 2.2</b>	<b>3.59%</b>
<b>Android 4.3</b>	<b>2.00%</b>
<b>Android 3.2</b>	<b>1.07%</b>
<b>Android 2.1</b>	<b>0.76%</b>
<b>Android 3.1</b>	<b>0.26%</b>
<b>Android 1.6</b>	<b>0.09%</b>

Android-sovellus perustuu löyhästi Model-View-Controller-malliin kuvan 7 mukaisesti. Käyttäjätunnistus upotetaan Activity-komponenttiin, joka samalla muodostaa käyttöliittymän. Tiedonsiirrosta web-sovellukselle huolehtii Service-komponentti, joka samalla toimii mallin Controller-osana. Käyttäjän tarvitsemat tiedot tallennetaan Androidin omiin SQLite- ja tiedostorakenteisiin.



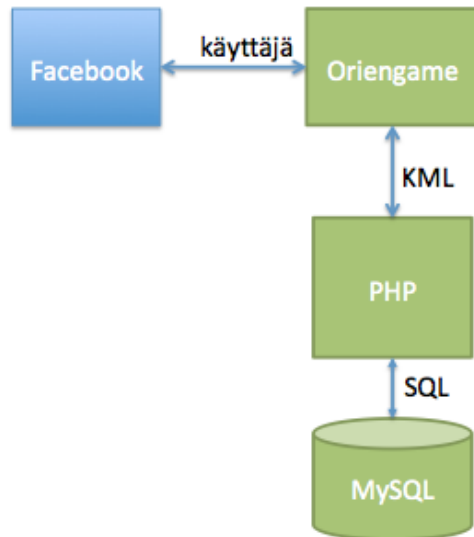
Kuva 7. Android-sovelluksen rakenne.

## Käyttäjätunnistus

Koska www-sovelluksen käyttäjätunnistus perustuu Facebook-tilin käyttöön täytyy vastaava tunnistusmekanismi tarjota myös Android-sovellukselle. Android-sovelluksille on tarjolla Java-pohjainen Facebook SDK -ohjelmistokehitysympäristö, jolla käyttäjätunnistus saadaan helposti toteutettua. Facebook SDK tarjoaa ylemmän tason funktiot Facebook-tilin käyttäjätunnistukseen. Näin ollen suorat REST- ja OAuth-pohjaiset alemmantason Facebook-funktioita voidaan jättää pois, joihin Facebook-ohjelmointirajapinnan käyttö perustuu ja joilla vastaava toiminnallisuus on tarvinnut aikaisemmin toteuttaa. Tarkoituksena on hyödyntää vain Facebook-käyttäjän nimitietoja sekä kaveriluetteloa jättäen muu Facebook-toiminnallisuus pois. Jatkossa voidaan toki hyödyntää Facebookin tarjoamaa informaatiota. Lisäksi sidos Facebookin käyttöön halutaan jättää löyhäksi mahdollistamalla joko sen poistamisen, korvaamisen jollain toisella käyttäjän tunnistusmenetelmällä tai sen laajentamisen muilla tunnistusmenetelmillä.

## Suunnistusradan hakeminen

OrientGame Android -sovelluksen suunnistusratojen haku perustuu olemassa olevan sovelluksen laajentamiseen. Radat haetaan KML-tiedostoina Web-palvelimelta HTTP-yhteyden avulla. Koska suunnistusratojen luominen ja editointi tapahtuu Web-sovelluksessa ja se huolehtii myös KML-muotoisen tiedon välittämisestä Android-sovellukselle, voidaan toiminnallisuus jättää ennalleen. Näin voidaan toteuttaa vain toiminnallisuus sille, että oikean käyttäjän suunnistustiedon hakeminen sallitaan vain sen tiedon oikealle käyttäjälle.

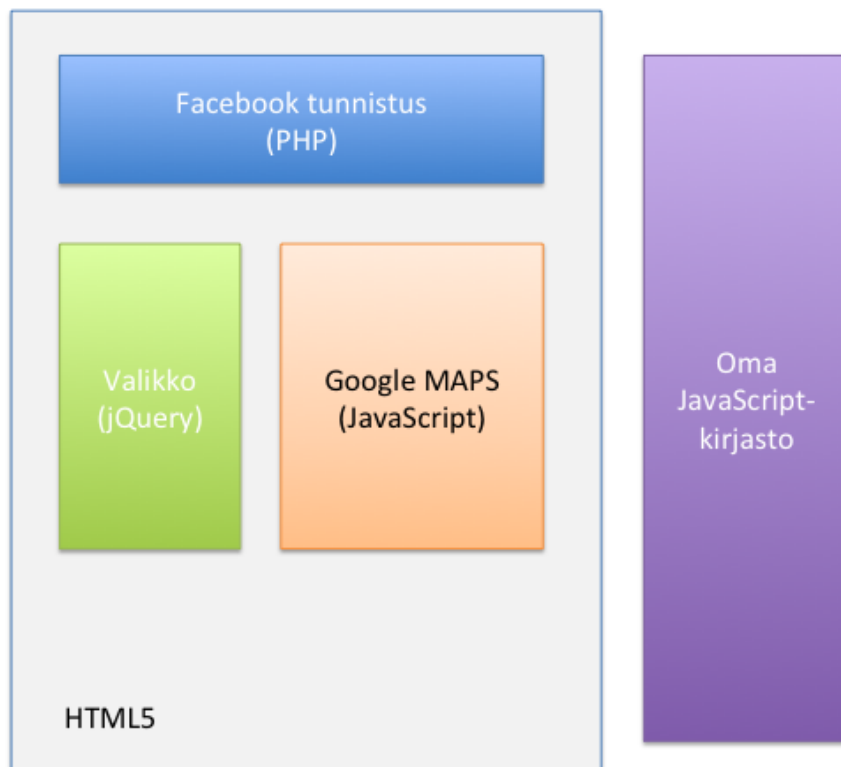


Kuva 8. OrientGame Android-sovelluksen suunnistusrastien haku

Kuvassa 8 on havainnollistettu komponentit, jotka osallistuvat suunnistusradan karttatiedon hakemiseen. Ideana on se, että OrientGame Android-sovellus linkittyy Facebook-käyttäjätiliin ja sallii vain käyttäjän ja hänen kaveritietonsa hakemisen. Facebook-komponenttia ei haluta kytkeä suoraan OrientGame-karttatietojen hakemiseen. Tällä mahdollistetaan se, että Facebook-käyttäjätili on riippumaton OrientGame-sovelluksen käyttämän käyttäjätiedon kanssa. Näin ollen voidaan jatkossa, mikäli tähän päädytään, vaihtaa Facebook-käyttäjätunnistus toisenlaiseen tunnistusmekanismiin. Tietokantaan tallennetaan käyttäjän Facebook-nimi, jonka perusteella tiedon haut toteutetaan.

### 3 Www-sovelluksen toteutus

Www-sovellus (kuva 9) on rakennettu kaupallisen web-hotellin tarjoaman alustan päälle, jossa on käytössä Apache Web -palvelinsovellus, PHP-kielen tuki sekä MySQL-tietokanta. Www-sivusto on kirjoitettu palveluntarjoajan teknologioiden pohjalta. Lisänä on käytetty JavaScript- sekä jQuery -kirjastoja. Web-selainten HTML5/CSS3 tukea on pyritty hyödyntämään ja sovellus on kehitetty Firefox- ja Google Chrome -selaimille yhteensopivaksi.



Kuva 9. Www-sovelluksen HTML5-pohjainen layout.

#### 3.1 Käyttäjätunnistuksen toteutus

Käyttäjätunnistusta ei kannata nykyaikana toteuttaa itse, koska se sellaisenaan olisi jo työläs ja oman aiheensa vaativa. Lisäksi sovelluksen käyttäjiä voisi olla vaikeata saada rekisteröitymään ja luottamaan vielä yhteen käyttäjätunnistukseen. Markkinoilla on useita vaihtoehtoja tarjolla käyttäjätunnistukseen, kuten Google, OpenID, Twitter ja Facebook. Käyttäjätunnistuksessa päädyttiin käyttämään Facebookin tunnistusta, koska se taitaa olla tunnetuin. Se tarjoaa tarvittavan API:n web- sekä tärkeimmille älypuhe-

linalustoille. Lisäksi Facebook-tilin käyttö mahdollistaa valmiin suljetun ryhmän eli näkymän ko. henkilön kaverilistaan, joka on mainio kohdeyleisö hyödyntää suunnistusratojen jakelussa. Samalla muodostuu myös riski niille käyttäjille, jotka eivät halua jakaa Facebook-kaverilistaansa kolmannelle osapuolelle tai yleensä omaa profiilitietoansa. Facebook-käyttäjätunnistus-menetelmä voidaan suhteellisen helposti vaihtaa toisenlaiseksi, kuten OpenID-pohjaiseksi. On jätetty mahdollisuus lisätä myös muita käyttäjätunnistusmenetelmiä, koska käyttäjätiedoista hyödynnetään vain käyttäjän omaa nimitietoa. Tässä piilee sovelluksen myös pieni heikkous, koska samanimisten henkilöiden erottaminen ei tässä toteutuksessa ole mahdollista. Haitta nähdään tietoturvamielessä pienenä, koska samanimiset käyttäjät eivät kuitenkaan pääse toistensa Facebook-tilitietoihin kiinni, mutta saattavat päästä käyttämään samaa suunnistusrataa.

Facebook API:n käyttöön löytyy verkosta Facebook Developer -sivustolta ja kirjallisuudesta [11] paljon materiaalia. Vuonna 2007 Facebook julkaisi oman alustan sovelluskehittäjille. Alusta perustuu HTML-pohjaiseen Facebook Markup Language (FBML), sovellusrajapintaan (API) sekä REST (Representational state transfer) -pohjaisten kutsujen käyttöön. Nykyisin Facebook API on toteutettu yleisimmille ohjelmointikielille ylemmän tason ohjelmointikirjastona. PHP-sovellusta varten on olemassa oma kirjastonsa.

Ennen kuin Facebook API:a voidaan käyttää, pitää sovelluksen kehittäjän rekisteröidä itsensä (kuva 10) ja sovelluksensa developers.facebook.com-sivustolla, jossa sovellukselle luodaan ns. API-avain. Avaimella Facebook tunnistaa sovelluksen. Koska Orient-Game-sovellus käyttää Facebook-käyttäjätunnistusta, on vaatimuksena kertoa rekisteröinnissä lisätietona URL-osoite (kuva 11), jonne Facebook-tunnistuksesta palataan onnistuneen tunnistuksen jälkeen. Facebookin tunnistus perustuu OAuthiin, jonka tarkoituksena on suorittaa käyttäjätunnistus ja valtuuttaa kolmannen osapuolen palvelu käyttämään tunnistetun henkilön tiliä ilman, että kolmannelle osapuolelle kerrotaan käyttäjän tunnusta ja salasanaa. Sovelluksen käyttäjälle Facebook-tunnistus tapahtuu sen omalla sivulla ja OrientGame-web-sovellus ei koskaan saa tietoonsa Facebook-käyttäjän tunnusta ja salasanaa.

## Sovellukset ▶ Orientgame

★ Markkinoi sovellusta
Muokkaa sovellusta
+ Create New App

---

**Asetukset** Edit Settings

Sovellustunnus/API-avain  
369674233143061

App Secret  
Näytä

App Namespace  
orientgames

Eristystila  
Pois

Käytössä olevat toiminnot  
Sivusto Facebook Login -toiminnolla, Android-sovellus

Last Mobile Install Reported

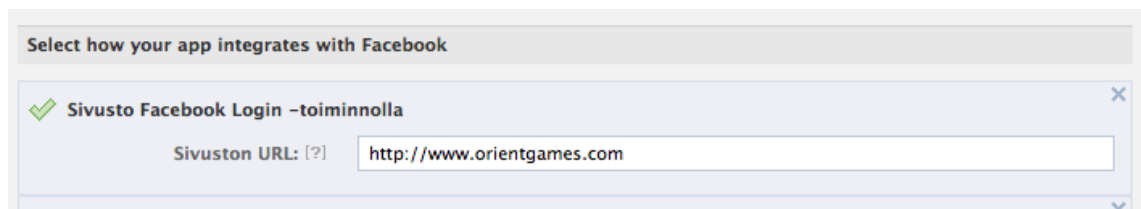
iOS: Eilen 18:07

Undetermined OS: 27. heinäkuuta 2013 kello 15:20

**Kehittäjän hälytykset** Näytä kaikki

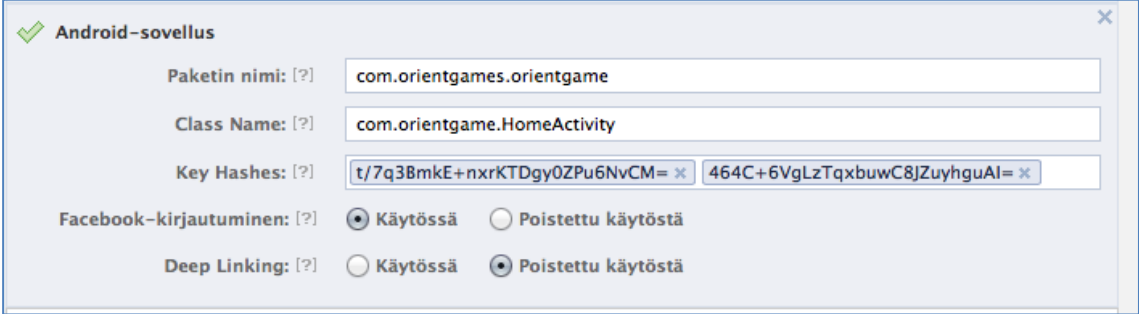
Sinulla ei ole kehittäjän hälytyksiä.

Kuva 10. Facebook-sovelluksen rekisteröinti kehittäjäisivustolla.



Kuva 11. Www-sovelluksen käyttäjätunnistuksen URL:n määrittely.

Facebook-sovelluksen rekisteröinnin yhteydessä voidaan myös määrittellä muiden alustojen käyttäjät. Android-sovelluksen tapauksessa tulee lisäksi määrittellä käytetyn sovelluksen paketin nimi sekä Android Activity -komponentti, joka tunnistusta käyttää (kuva 12). Lisäksi vaaditaan erillinen varmistus (Key Hashes), jolla sovellus voidaan tunnistaa ja näin ollen rajoittaa rajapinnan väärinkäyttöä. Muille älypuhelinlustoille on tehtävä vastaavanlaiset määrittelyt niiden ominaisuuksien mukaisesti.



Android-sovellus

Paketin nimi: [?]

Class Name: [?]

Key Hashes: [?]

Facebook-kirjautuminen: [?]  Käytössä  Poistettu käytöstä

Deep Linking: [?]  Käytössä  Poistettu käytöstä

Kuva 12. Android-sovelluksen rekisteröinti Facebookissa

Facebook PHP-pohjainen käyttätunnistus on toteutettu esimerkkilistauksen 1 mukaisella PHP-ohjelmalla. PHP-pohjainen API (facebook.php) istuu parhaiten käytettyyn web-alustaan, koska palvelinsovellus on muutoinkin kirjoitettu pitkälti PHP-kielellä. Ohjelmakoodissa voidaan nähdä, että siinä hyödynnetään PHP-istuntoa. Facebook-kirjasto esitellään PHP-kielen `require_once()`-metodilla, jonka jälkeen määritellään Facebook Developer -sivustolla luodut arvot. Nämä ovat Facebook-tunnistuksen tarvitsevat ja vaatimat `appId`, `secret` -tunnisteet, jotka luodaan Facebookin omilta kehittäjä sivuilta, kun tunnistus rekisteröidään. Samassa voidaan nähdä Facebook Graph API-toiminnallisuus, koska metodikutsulla `facebook-api('/me', 'GET')` kerrotaan, että halutaan hakea käyttäjän profiilitiedot ja se toteutetaan HTTP GET -metodilla. Facebook-käyttäjän nimi viedään myös PHP-istuntomuuttujaksi, jotta käyttäjän ei tarvitse tehdä uutta tunnistusta, mikäli Web-istunto on vielä voimassa.

```
// Luodaan tai palautetaan Web-istunto
session_start();

// vaaditaan Facebookin tarjoama valmis API-kirjasto
require_once("src/facebook.php");

// määritellään taulukkoon tarvittavat API:n vaatimat tunnisteet, jotka ovat luotu Facebookin kehittäjä sivustolla
$config = array();
$config['appId'] = '123456789043061';
$config['secret'] = 'abcdef01234567895abcdef0123456789';
$config['fileUpload'] = false; // optional
$config['test'] = true;
$username = '';
$facebook = new Facebook($config);
try {
    $user_profile = $facebook->api('/me','GET');
    //DEBUG echo "You are logged in as: " .
    $user_profile['name'];
    $username = $user_profile['name'];
}
```

```

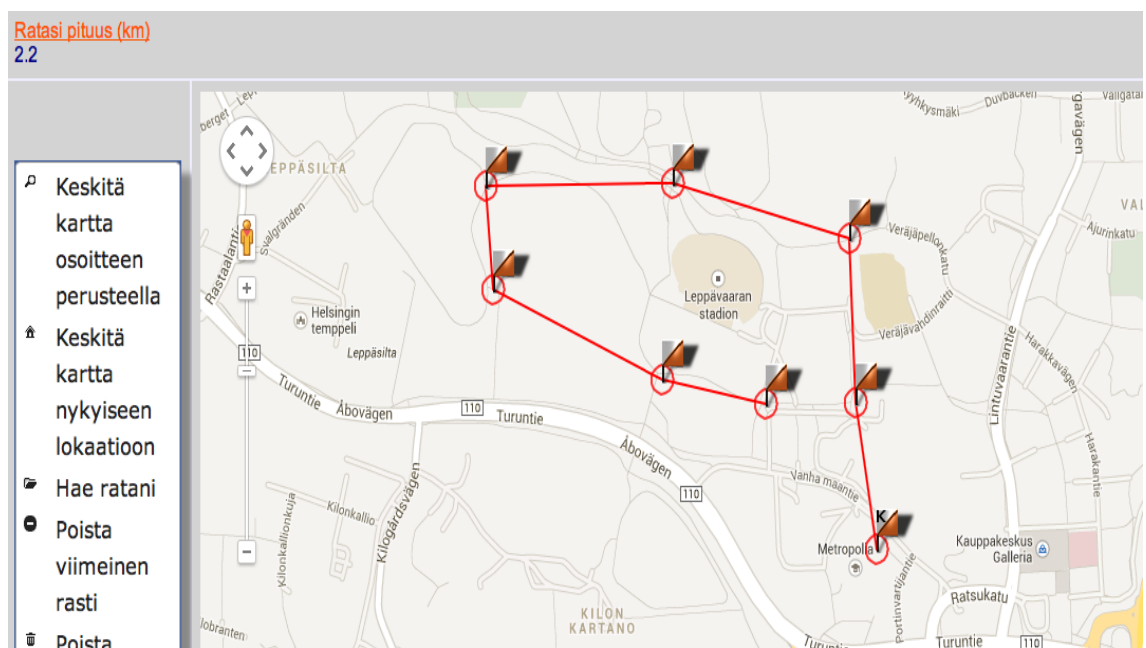
    $_SESSION['username'] = $user_profile['name'];
} catch(FacebookApiException $e) {
    $login_url = $facebook->getLoginUrl();
    //echo 'You need to <a href="' . $login_url . '">login with
face book to create track!</a>';
    // error_log($e->getType());
    // error_log($e->getMessage())
}

```

Esimerkkilistaus 1. Facebook API:n käyttöönotto www-sivulla.

### 3.2 Kartta- ja paikkatiedot

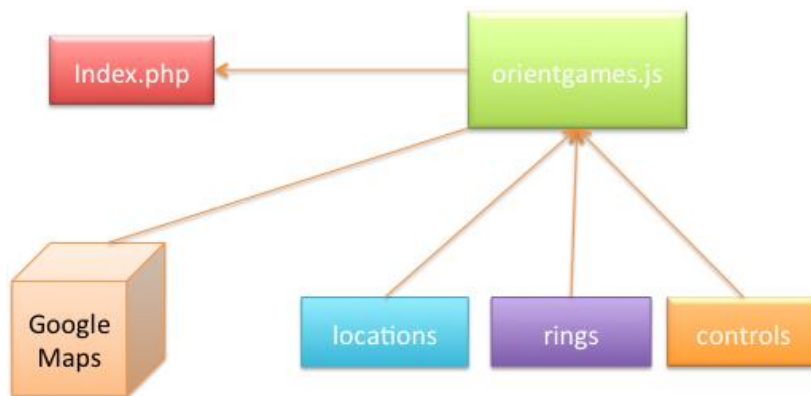
Www-sivustolle on toteutettu kuvan 13 mukainen näkymä, jolla suunnistusrata voidaan piirtää. Se on toteutettu jQuery- ja JavaScript -kielillä ja perustuu JavaScript-pohjaiseen Google Maps -kirjastoon. Se mahdollistaa, että omalla Web-sivustolla voi käyttää Googlen tarjoamaa kartta- ja paikkatietoa. Google Maps API tarjoaa myös suunnistusradan näyttämiseen sopivan graafisen kirjaston, jonka varaan on voitu rakentaa sopivat metodit.



Kuva 13. Web-näkymä suunnistusrata työkalusta

Google Maps -olio sisällytetään erillisenä www-elementtinä, jonka jälkeen sille voidaan erillisillä JavaScript-koodilla luoda niin sanottuja Overlay-olioita sekä linkittää www-sivun käyttöliittymän tapahtumia. Google Maps toiminnallisuus on toteutettu Ja-

JavaScript-kirjastossa OrientGames.js. Kirjasto hallinnoi ja ylläpitää suunnistusradan rakennetta. Oleelliset komponentit on kuvattu kuvassa 14. Käyttäjän muokatessa suunnistusrataa välitetään tapahtumatiedot Google Maps -oliolle, joka hallinnoi kartalla näkyviä graafisia ratakomponentteja. Controls-oliot sisältävät käytännössä Google Maps Marker -olion. Tämä olio pitää sisällään tiedon geografisesta paikasta sekä pisteen graafisen ulkoasun määrittämisen, joka toteutetaan ikonina. Ikoni on kuvatiedosto, joka sidotaan itse Marker-olioon ja näkyy käyttäjälle rastilippuna. Marker-olioon sidotaan myös erilaiset editointitapahtumien käsittely. Tämä tarkoittaa hiirellä tehtävien tapahtumien käsittelyä kuten rastin valinta hiiren painikkeella tai rastin siirto kartalla siirtämällä sijaintia hiirenpainike pohjassa. Rings-oliota käytetään sitomaan rastiympyrä käytetyn rastin kohdalle ja se piirretään erillisellä funktiolla. Locations-oliota käytetään taas tallentamaan rastipaikkojen koordinaatit, jota hyödynnetään radan viivojen piirtämisessä.



Kuva 14. Google Maps -karttaolion hyödyntäminen www-sovelluksessa.

Google Maps API saadaan käyttöön www-sivustolla sisällyttämällä tarvittava JavaScript-kirjasto, jossa on määritelty API:n käyttämä avain "key". Tämä avain pitää generoida Google Developer -sivustolla (esimerkkilistaus 2):

```

<script
src="https://maps.googleapis.com/maps/api/js?key=putHereYourKeyYouHaveGotFromGoogle&sensor=false&libraries=geometry">
  
```

Esimerkkilistaus 2. Google Maps -kirjaston alustus JavaScriptissä.

Samalla määritellään `sensor = false`, jolloin sovelluksessa ei ole erikseen paikannusta tarjoavaa sensoria. Sovellus hyödyntää myös lisänä kirjastoa "geometry", jota tarvitaan muun muassa kartalla olevien etäisyyksien laskemiseen. Tätä hyödynnetään, kun lasketaan suunnistusradan kokonaispituus linnuntietä käyttäen. Google Maps API:a varten on [www-sivustolle](http://www-sivustolle) kehitetty kirjasto, joka tarjoaa valmiit Ajax-pohjaiset funktiot radan piirtämiseen, editoimiseen ja tallentamiseen.

Sovelluksen tarvitsemat Googlen Maps -kirjaston alustus voidaan tehdä esimerkkilistauksen 3 mukaisesti. Ensin alustetaan kartan malli, joka tässä tapauksessa on Roadmap-tyyli. Sitten karttaoliolle lisätään käsittelijä, joka tutkii hiiren painallukset ja sen perusteella joko lisää graafisen rastilippua kuvaavan olion kartalle. Lisäksi lisättyyn kohtaan piirretään pieni ympyrä. Oliolle lisätään myös veto-siirto-tapahtuma, jolla mahdollistetaan graafisen olion siirtäminen, kun se on asetettu kartalle.

```
function initialize() {
    /* alusta sijainti nykyisten koordinaattien mukaisesti */
    var myLatLng = new google.maps.LatLng(currentLat, current-
    lng);

    var mapOptions = {
        center: myLatLng,
        zoom: 15,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

    map = new google.maps.Map(document.getElementById("map"),
    mapOptions);

    google.maps.event.addListener(map, 'click', function(e) {

        if (index == 0)
            trackPin = "res/trackpink.png"
        else
            trackPin="res/trackpin.png";

        marker = new google.maps.Marker({
```

```

        position: e.latLng,
        map:map,
        mapTypeControl:true,
        title:index.toString(),
        clickable: true,
        draggable: true,
        icon:{url: trackPin, anchor: new
google.maps.Point(0,30)}
    });

    index++;
    locations.push(e.latLng);
    insertCircle(e.latLng);
    controls.push(marker);

    // Add dragging event listeners.
    google.maps.event.addListener(marker, 'dragstart',
function(e) {

    });
    google.maps.event.addListener(marker, 'drag', func-
tion() {});
    google.maps.event.addListener(marker, 'dragend', func-
tion(e) {
        for (i in controls) {
            if (e.latLng == controls[i].getPosition()) re-
placeControl(e.latLng, i);

        }
    });
    refreshRoute(locations);
}); // 'click' listenerin loppu
geocoder = new google.maps.Geocoder();
} // inititalize loppu

```

**Esimerkkilistaus 3. Google Maps -kirjaston alustuksesta.**

MySQL-tietokanta (kuva 15) sisältää taulut kunkin käyttäjän ratatietojen tallentamiseen. Tässä vaiheessa tallennetaan vain kunkin käyttäjän Facebookissa käyttämä nimi sekä Web-sivustolla tehdyn radan rastien koordinaatit ja järjestys. Toiminnallisuus on pidetty mahdollisimman yksinkertaisena ja tehokkaana. Periaate on, että uuden radan tietoja ei päivitetä vanhan radan päälle vaan aina radan päivityksen yhteydessä vanha olemassa oleva rata tuhotaan tietokannasta ja uuden radan tiedot kirjoitetaan uusina.



Kuva 15. Tietokannan karkea rakenne.

Radan tallennusta varten on perustettu kaksi tietokantataulua, joilla on yksi-moneen suhde ts. Track-taulu toimii hakemistona radan käyttämiin rastipisteisiin. Control Point -taulu määrittää kaikki rastipisteet, johon tallennetaan koordinaatit sekä rastien järjestysradalla. Tästä taulusta on viittaus Track-tauluun, jotta sitominen oikealle käyttäjälle voidaan tehdä. Taulujen tietorakenne on kuvattu tarkemmin kuvissa 4 ja 5.

Id (Primary Key)	Int (11)
User	Varchar (50)
Description	Varchar (50)
Country	Varchar (50)
Timestamp	Date

Esimerkkilistaus 4. Tietokannan Track-taulun tietorakenne.

Id (Primary Key)	Int (11)
Track_id (Foreign Key)	Int (11)
N	Int (11)
Lat	Varchar (15)
Lng	Varchar (15)

Esimerkkilistaus 5. Tietokannan Control Point -taulun tietorakenne.

### 3.3 Tietokannan sovituserros ja Ajax-liityntä

Tietokantaoperaatiot on toteutettu PHP-kielillä ja toteuttaa perusfunktiot tietojen lukemiseen ja kirjoittamiseen tietokannasta. Tiedon lukeminen ja kirjoittaminen www-sivustolta hoidetaan JSON-tietorakenteita käyttäen ja jQueryn tarjoamaa Ajax-tukea hyödyntäen. Android-sovellus hakee tarvitsemansa tiedon KML-muodossa, koska tämä on ollut tietorakenteen muoto Android-sovelluksen varhaisemmassa vaiheessa.

Kukin tarvittavista tietokantaoperaatioista on toteutettu omana PHP-ohjelmalla ja niitä kutsutaan HTTP Request -metodilla. Web-käyttäjän tiedot ovat tallennettu PHP Sessio -muuttujaksi, jota käytetään oikean tiedon hakemiseen. Käyttäjän tiedot vastaavat Facebook -tunnistuksen käyttäjätietoja.

Esimerkkilistauksessa 6 on kuvattu PHP-pohjaisen tietokantayhteyden avaus:

```
// Kytkeydytään tietokantaan ja ilmoita mikäli tämä ei onnistu.
// Die -komento tulee poistaa tuotantoversiota.
mysql_pconnect("localhost:3306", $dbuser, $pwd)
    or die("Database connection failed : ".mysql_error());

// Valitaan oriengamesin käyttämä tietokanta.
mysql_select_db('OrientGames') or die("404 Not Found");
if (!isset($_SESSION['username'])) die("403 Forbidden");

// Poistetaan ylimääräiset merkit käyttäjän nimestä sekä haetaan käyttäjän nimi HTTP Sessiosta.
$user = strip_tags($_SESSION['username']);
```

Esimerkkilistaus 6. Esimerkki tietokantayhteyden avauksesta PHP-kieltä käyttäen.

PHP-ohjelman tarvitsevat muuttujat siirretään HTTP Request -sanomassa. Tätä varten PHP-kielessä voidaan käyttää \$\_REQUEST['muuttuja'] -komentoa, joka osaa hakea muuttujan sekä HTTP POST-, että HTTP GET Request -sanomissa. Tieto saadaan lisättyä tietokantaan esimerkkilistauksen 7 mukaisesti.

```
// Luodaan SQL INSERT lause uuden radan tallentamista varten.
```

```

$sql = "INSERT INTO Track (User, Description, Country,
Timestamp) VALUES ('$user', '$description', '$country',
'$date')";

mysql_query($sql) or die("400 Bad Request");

$sql = "SELECT Id from Track WHERE User = '$user'";
$res = mysql_query($sql);
$row = mysql_fetch_row($res);
$n = 0;
foreach ($track as $coord) {
    $coord = ltrim($coord, '(');
    $coord = rtrim($coord, ')');
    $coord = explode(',', $coord);
    $sql = "INSERT INTO ControlPoint (Track_id, N, Lat,
Lng) VALUES ('$row[0]', '$n', '$coord[0]', '$coord[1]')";
    mysql_query($sql) or die("400 Bad Request");
    $n = $n + 1;
}

```

Esimerkkilistaus 7. Tiedon lisääminen SQL-tietokantaan.

### 3.4 Www-sovelluksen pääsivu

Www-sovelluksen pääsivu perustuu PHP-kielen ja JavaScriptin käyttöön. Käyttöliittymän kuorurutukseen tarjoaa jQuery UI kirjaston, jonka avulla saadaan www-sivulle tehtyä hiukan normaalia sivua näyttävämpi valikkorakenne (esimerkkilistaus 8). PHP-koodin osalle jää http Session ylläpito, Facebook-kyselyt sekä halutun käyttöliittymäkielen valinta (englanti, suomi). Www-sivustolla on pyritty yhden sivun toteutukseen, jolloin kaikki toiminnallisuus tapahtuu käytännössä index.php-sivun sisällä.

```

<div class="menu">
<ul align="top" id="menu">
<li><a id="locate"><span class="ui-icon ui-icon-
search"></span>Keskitä kartta osoitteen perusteella</a></li>
<li><a id="locateme"><span class="ui-icon ui-icon-
home"></span>Keskitä kartta nykyiseen lokaatioon</a></li>
<li><a id="load"><span class="ui-icon ui-icon-folder-
open"></span>Hae ratani</a></li>
<li><a id="dellast"><span class="ui-icon ui-icon-circle-

```

```

minus"></span>Poista viimeinen rasti</a></li>
<li><a id="clear"><span class="ui-icon ui-icon-
trash"></span>Poista kaikki rastit</a></li>
<li><a id="save"><span class="ui-icon ui-icon-
disk"></span>Tallenna ratani</a></li>
<li><a id="help"><span class="ui-icon ui-icon-
info"></span>Ohjeet radan tekemiseen</a></li>
<script>
    $("#menu").menu();

    $("#locate").click(function() {
        $("#dialog-address").dialog({
            height: 200,
            width: 500,
            modal: true,
            buttons: {
                "Center": function()
                    centerToAddress($('#address').val());
                    $(this).dialog( "close" );
            }
            ,
            Cancel: function() {
                $(this).dialog( "close" );
            }
        }
    });

    $("#locateme").click(function() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(function
(position) {
                setLocation(position);
            });
        }
    });
    $("#dellast").click(function() {
        delLastPoint();
    });
    $("#clear").click(function() {
        clearTrack();
    });
    $("#load").click(function() {
        loadTrack('noname');
    });
    $("#save").click(function() {
        saveTrack('noname');
    });
    $("#help").click(function() {
        $("#dialog-help" ).dialog({
            height: 400,
            width: 600,
            modal: true
        });
    });
};

```

```

$(function() {
    $( document ).tooltip();
});

$(function() {
    $( "#language radio" ).buttonset();
});
$(document).ready( function(){
    $("dialog-help").hide();
});
</script>

```

Esimerkkilistaus 8. jQuery-pohjaisen menun tekeminen.

Pääsivu on lokalisoitu englannin ja suomen kielelle (esimerkkilistaus 9). Tämä on toteutettu PHP-kielisillä funktioilla sekä www-sivun session hallinnalla. Näin käyttäjän tullessa uudelleen sivulle hänen ei tarvitse valita käytettyä kieltä uudelleen.

```

if ($_SESSION[lang] == "") {
    $_SESSION[lang] = "en";
    $currLang = "en";
} else {
    $currLang = $_GET[lang];
    $_SESSION[lang] = $currLang;
}

switch($currLang) {
    case "en":
        define("CHARSET", "utf-8");
        define("LANGCODE", "en");
        break;
    case "fi":
        define("CHARSET", "utf-8");
        define("LANGCODE", "fi");
}

```

Esimerkkilistaus 9. PHP-kielinen lokalisoitivalinta.

Kun lokalisoititietoa pidetään yllä PHP-kielisessä muuttujassa `$currLang`, voidaan tätä käyttää myöhemmin sivustolla eri tulostusvaiheissa, jolloin muuttujan perusteella tulostetaan kyseessä olevaa kieltä vastaava sisältö.

## 4 Android-sovelluksen toteutus

Seuraavassa kappaleessa käydään läpi OrientGame Android -sovellukselle tehtyjä muutoksia, jotta siinä voidaan käyttää www-sovelluksessa tehtyjä ratoja.

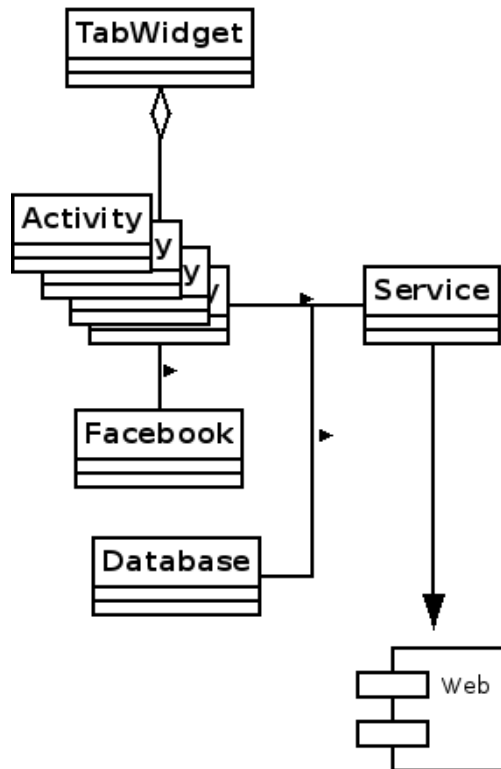
### 4.1 Sovelluksen rakenteelliset muutokset

Alkuperäinen OrientGame-sovellus ei tarjoa käyttäjälle minkäänlaista käyttäjän tunnistusmekanismia, joten se on ensimmäinen toiminnallisuus, joka sovellukseen lisättiin. Lisäksi tarvittiin toiminnallisuus, jotta Facebookin kautta tunnistetun henkilön ja hänen kavereidensa nimitietoja voidaan käyttää oikean tiedon hakemiseen palvelimelta.



Kuva 16. Android TabWidget -toteutus.

OrientGame-sovellus on jaettu useaan Androidissa käytettyyn Activity-komponenttiin, joilla jokaisella on oma elinkaarensa (kuva 17). Activity-komponentti on myös olio, joka tyypillisesti tuhoetaan Android-sovelluksessa, kun sen näyttämä käyttöliittymä katoaa sovelluksen näytöltä. Eri Activity-komponentit on sidottu yhteen erityisen TabWidget Activity -komponentin avulla (kuva 16), joka mahdollistaa käyttäjän valita eri Activity-komponenttien ja toiminnallisuuksien välillä. Tätä varten OrientGame-sovelluksella on käytössään Service-komponentti, joka mahdollistaa sovelluksen tausta-ajon ja tiedon jakamisen eri Activity-komponenttien välillä, kun ne aktivoituvat ja passivoituvat (eli katoavat näytöltä muun toiminnallisuuden tullessa näytölle) käyttäjän toimenpiteiden seurauksena.

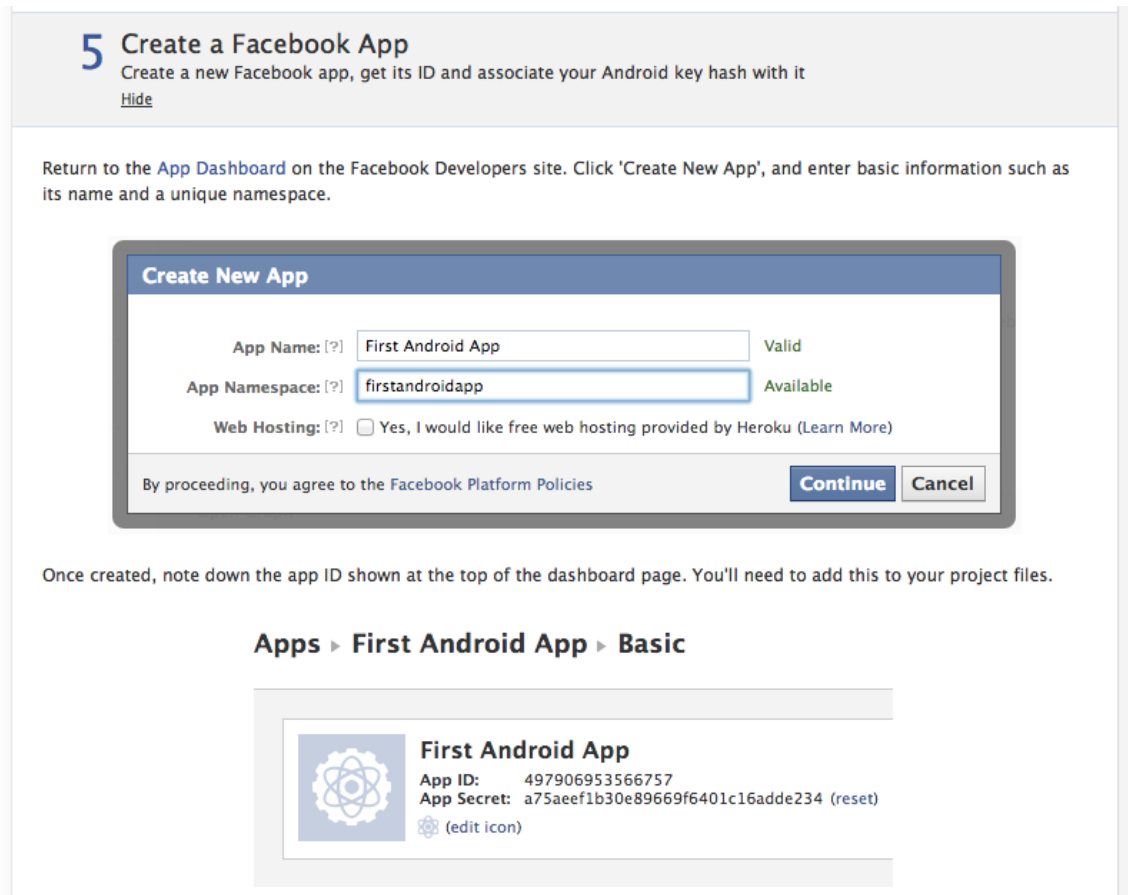


Kuva 17. Android-sovelluksen rakenne

#### 4.2 Facebook-tunnistuksen toteutus

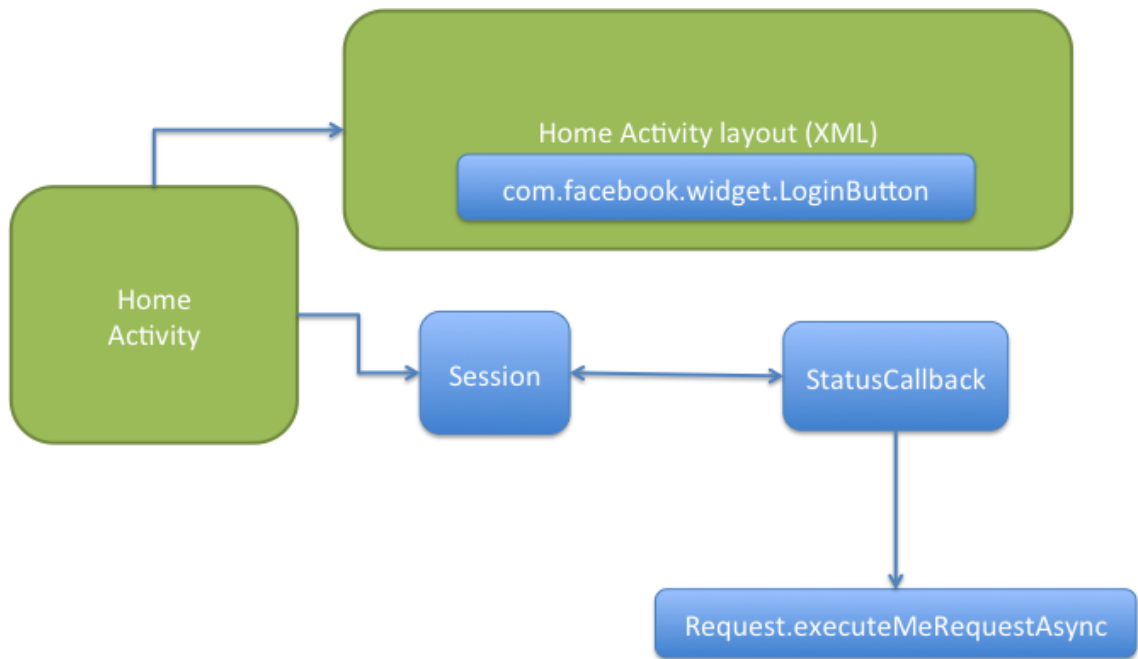
OrientGames-sovelluksen käyttäjätunnistus toteutetaan siten, että sovellus rekisteröidään ensin Facebook-sovelluksena. Tämän jälkeen OrientGames-sovelluksessa voidaan hyödyntää täysin Facebook-käyttäjätiliä ja sen tarjoamia ominaisuuksia, kuten kaveriluetteloa. Tämä on näppärää, koska sovellukseen ei tarvitse ohjelmoida vastaavaa toiminnallisuutta itse. Facebook-integraatio voidaan toteuttaa helposti Android-sovelluksille tarkoitetulla ohjelmakirjastolla (Facebook SDK), joka tarjoaa perusfunktiot.

OrientGames on rekisteröity Facebook-sovellukseksi kuvan 18 osoittamalla tavalla. Tämä on tehty Facebook-kehittäjäportaalissa. Jotta portaalilla voi käyttää täytyy myös kehittäjän rekisteröityä ensin Facebook-sovelluskehittäjäksi. Facebook-kirjasto tuodaan Android-kehitysyökaluun (esim. Eclipse) kirjastona ja sisällytetään OrientGames-sovellukseen.



Kuva 18. Android Facebook API:n käyttöönotto (kuvan tunnistetiedot eivät ole käytössä).

OrientGame-sovelluksen Facebook-integraatio on toteutettu HomeActivity-komponentti-aktiiviteetissa. HomeActivity-komponentti on istutettu aikaisemmin kuvattuun Android TabWidget -komponenttiin, jota kuvaa käyttöliittymässä näkyvä vihreä talon kuva. Tähän aktiviteettiin on ohjelmoitu graafinen Facebook-painike, jota koskettamalla käyttäjä voi kirjautua Facebook-tilille. Sovellusta ensimmäistä kertaa käytettäessä uutta käyttäjää pyydetään sallimaan pääsy Facebook-tilin tietoihin. Kuvassa 19 on kuvattu tarvittavat oliot Facebook-tilille kirjautumisen toteuttamiseen.



Kuva 19. Facebook-tilin kirjautumiseen liittyvät oliot.

Android-sovelluksen käyttöliittymä määritellään yleensä erillisessä XML-tiedostossa, joka käännetään osaksi sovelluksen ohjelmatiedostoa. Facebook Login -painike saadaan helposti otettua käyttöön määrittelytiedostossa esimerkkilistauksen 10 osoittamalla tavalla. Se saadaan toimimaan myös niin, että mikäli sovelluksen käyttäjät eivät ole kirjautuneet Facebook-tilille, voidaan heille näyttää erillinen Facebookin Login-kirjautumisnäkyvä.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
...

<com.facebook.widget.LoginButton
android:id="@+id/authButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
android:layout_marginTop="1dp"/>
  
```

```
...
</RelativeLayout>
```

Esimerkkilistaus 10. Facebook Login -painikkeen määrittely XML-tiedostossa.

Kuten esimerkkilistauksen 10 mukaisesta XML-kielisestä käyttöliittymän esittelystä huomaa, toteutetaan Facebook-painike ns. Widget-komponenttina. Facebook SDK sisältää tälle valmiin toteutuksen, jolloin Login-painike voidaan ottaa sovelluksessa helposti käyttöön. Facebook Login widget ylläpitää henkilön kirjautumista tilille. Tilan vaihtuessa kirjautuneeksi, osaa widget muuttaa painikkeen automaattisesti Facebook Logout -painikkeeksi. Login-widget sisältää valmiina Facebook Session -olio, jota käytetään ns. Singleton-tyyppisenä luokkana. Sessio-olio hallitsee Facebook-sovelluksen istuntoa. Tämä tarkoittaa sitä, että vaikka Session -olioita luodaan useita, viittaavat sen hallitsemat tiedot samaan olioon. Activity-komponentin Facebook-luokat tuodaan mukaan normaalisti import-lausekkeilla ja tämän jälkeen voidaan Sessio-oliota kutsua luokkaoliona.

Facebook Login Widget toteuttaa sellaisenaan kirjautusmispainikkeen. Mikäli käyttäjä ei kuitenkaan ole kirjautunut Facebook-tilille painikkeella, täytyy ohjelmakoodiin määritellä Session-luokkametodilla kirjautumisen tarkistus. Tämä voidaan toteuttaa kuvan 25 mukaisella metodilla.

```
// Avataan Facebook Sessio.
Session.openActiveSession(this, true,
    new Session.StatusCallback() {

        // Määritellään callback metodi, jota kutsutaan, kun Face-
        // book login on palauttanut tiedon asynkronisesti.
        @Override
        public void call(Session session, SessionState state, Ex-
        ception exception) {
            // Tarkista onko Sessio auki
            if (session.isOpened()) {

                // hae Facebook /me -komennolla käyttäjän tiedot
                Request.executeMeRequestAsync(session, new Re-
                quest.GraphUserCallback() {

                    // määritellään takaisinkutsu Graph API vastauselle
                    @Override
                    public void onCompleted(GraphUser user, Response
                    response) {

                        // Tarkista onko käyttäjä olio palautettu
```

```

        if (user != null) {
            // Lue käyttäjän nimi omaan muuttujaan
            userIs = user.getName();
        }
    }
});
}
});

```

Esimerkkilistaus 11. Facebook-istunnon avaus ohjelmallisesti.

Facebook Sessio-oliota käytetään luokkametodeilla. Kun käyttöliittymässä on Login-painike esitelty, voidaan samaan Sessio-olioon viitata helposti ohjelmakoodissa kutsuamalla `Session.getActiveSession()`, joka palauttaa tilatiedon aktiivisesta Facebook-kirjautumisesta.

#### 4.3 Facebook-kaveriluettelon hakeminen

Facebook-kaveriluettelon hakemiseen käytetään Facebook SDK:n mukana tulevaa `FriendPicker Activity` -komponenttia. Kun Facebook-tilille on kirjauduttu, voidaan `FriendPicker Activity` laukaista sovelluksessa, jolloin aktiviteetti tarjoaa käyttäjälle oman kaveriluettelon ja mahdollisuuden poimia haluttu kaveri listalta. Tyypillisesti Facebook-kaveriluettelosta voi poimia useita henkilöitä, mutta sovelluksessa tätä on rajoitettu niin, että vain yksi (tai ei ketään) henkilö voidaan valita. Kun henkilö on valittu, voidaan se Android-sovelluksessa käsitellä `onActivityResult()`-metodilla, jonka paluuarvoista voidaan lukea valitun henkilön nimi ja käyttää sitä myöhemmin ohjelmassa.

#### 4.4 Karttatiedon hakeminen

Android-sovellus käyttää Googlen karttapalvelua näyttääkseen sen sovelluksessa. Jotta karttapalvelua voidaan käyttää, täytyy sovellus erikseen rekisteröidä Googlelle ja saada sieltä tarvittavat tunnistetiedot sovelluksen tekemiseen. Koska sovelluksen ensimmäinen versio on jo kehitetty ennen tämän työn aloitusta, keskitytään tässä vain kertomaan miten sovellus saadaan adaptoitua käyttämään www-sivustolla olevaa suunnistusradan tietoa. Tarvittavat toimenpiteet Googlen karttapalveluiden käyttä-

miseksi on tehty jo sovelluksen aikaisemmassa versiossa. Esimerkkilistauksessa 12 on näytetty miten suunnistusratoja kuvaava KML-tiedosto noudetaan Web-palvelimelta. Periaate on yksikertainen ja ehkä hiukan prototyypimäinen. Tietoturva on hiukan paranneltu suoralta Tamper [6] yritykseltä, luomalla tarvittavaan Web-palvelun pyyntöön hash-tyyppinen tarkistus. Hash tehdään SHA1-tyylisenä, jonka avaimena käytetään sekä Web-palvelimelle, että Android-sovellukselle sovelluskehitysvaiheessa kirjattua ennakkoon jaettua yhteistä avainta. Tällä hetkellä Web-pyyntöt eivät ole salattu, mutta toteutukseen on suhteellisen helppoa lisätä HTTPS-tyyppinen kerros. Tämä vaatisi kuitenkin palveluntarjoajalta SSL-sertifikaatin hakemista, joka on tässä vaiheessa koettu tarpeettomaksi.

```
// Hae rata vain, jos henkilö on valittu. Servicen pitää olla
// päällä
if (trackServiceBound && fbuser != null) {
    try {
        fbuserEn = URLEncoder.encode(fbuser, "UTF-8");
        System.out.println(fbuser);
    } catch (UnsupportedEncodingException uee) {
        System.err.println(uee);
    }
    String urlToLoad =
    "http://www.OrientGames.com/readkmltrack.php?fbuser=";
    // Luo hash (SHA1) http requestille
    try {
        MessageDigest digest = MessageDigest.getInstance("SHA1");
        urlToLoad +=fbuserEn;
        String hash = fbuser + "<secret>";
        byte[] passwordBytes = hash.getBytes();
        digest.reset();
        digest.update(passwordBytes);
        byte[] padding = digest.digest();
        StringBuffer hexString = new StringBuffer();
        for (int i=0; i<padding.length; i++) {
            String s = Integer.toHexString(0xFF & padding[i]);
            if (s.length() < 2)
                hexString.append("0"+Integer.toHexString(0xFF &
padding[i]));
            else
                hexString.append(Integer.toHexString(0xFF & pad-
ding[i]));
        }
        urlToLoad += "&og="+hexString.toString();
        //System.out.println(urlToLoad);
        trackPoint = trackService.fetchTrackDataAsync(urlToLoad);
    } catch (Exception ex) {
        System.err.println(ex);
    }
}
}
```

Esimerkkilistaus 12. KML-tiedoston lataus sekä SHA1-hash luonti.

## 4.5 Muut muutokset

Koska Android-sovellus ei automaattisesti tallenna siinä käytettyjen Activity-komponenttien käyttämiä tietoja, on OrienGame-sovellukseen toteutettu tiedon säilytystoiminne. Tämä näkyy käyttäjälle, kuin tieto ei olisi koskaan muistista kadonnutkaan. Tieto siis säilyy näennäisesti sovelluksessa, kun se suljetaan ja taas avataan seuraavalla kerralla. Radan tallennus välimuistiin on toteutettu käyttäen Androidin perustiedostontiedostonkäsittelyrutiineja hyödyntäen. Käytännössä Android-sovellus käyttää Javasta tuttua tyyliä ja tiedostoa voidaan lukea ja kirjoittaa niin sanotussa streammuodossa. Toteutuksen vaatimat metodit on esitelty listauksessa 13. Koska verkosta ladattava suunnistusratatieto on KML-muodossa, eli käytännössä merkkipohjainen tiedosto, voidaan tallennus tehdä suoraan tässä muodossa. Tiedoston lukeminen voidaan myös toteuttaa vastaavalla tavalla, koska ohjelmassa on olemassa oleva logiikka KML-mallisen tiedon käsittelyyn.

```

/*
 * Tallentaa palvelimelta ladattu rata tiedostoon.
 * Syöttötieto on xml-jono
 * Palauttaa arvon true tai false onnistumisen merkiksi
 */
private boolean cacheTrack(String xmlContent) {

    String filename = "ogtrack";
    FileOutputStream outputStream;

    try {
        outputStream = openFileOutput(filename, Context.MODE_PRIVATE);
        outputStream.write(xmlContent.getBytes());
        outputStream.close();
    } catch (Exception e) {

        e.printStackTrace();
        return false;
    }
    return true;
}

/*
 * Lataa tallennettu rata.
 * Ratatieto palautetaan XML-jonona.
 */
private String loadCachedTrack() {

    String filename = "ogtrack";
    String xmlData = "";

```

```

try {
    File myFile = new File(this.getFilesDir(), filename);
    FileInputStream fIn = new FileInputStream(myFile);
    BufferedReader myReader = new BufferedReader(
        new InputStreamReader(fIn));
    String aDataRow = "";
    while ((aDataRow = myReader.readLine()) != null) {
        xmlData += aDataRow + "\n";
    }
    myReader.close();
} catch (Exception ex) {
    System.err.println("OrientGames: "+ex.getMessage());
}

return xmlData;
}

```

### Esimerkkilistaus 13. KML-tiedoston tallennus ja lataus

Tiedostojen virheenkäsittelyyn ei ole toteutettu sen kummempaa mekanismia. Käyttäjä näkee virheilmoitukset siinä tapauksessa, että luultua tallennettua rataa ei löytynyt. Tämä saattaisi vaatia hiukan huolellisempaa käyttäjän huomioimista esimerkiksi kertomalla tämä Android-laitteen status-rivillä.

Android-sovelluksessa uusittiin valikkoikonit, jotta sovellus näyttäisi hiukan ammattimaisemmalta. Sovelluksen uusina ikoneina käytetään vapaasti saatavilla olevia graafisia elementtejä, jotka noudattava erillistä avointa lisenssiehtoa [15]. Ikoneihin päädyttiin, koska sovelluksen graafinen suunnittelu on jo sellaisenaan oma työläs asia ja vaatii asiaan perehtymistä ja myös taiteellisia kykyjä.

## 5 Testaus

### Www-sovellus

Www-sovelluskehityksessä voidaan hyödyntää nykyisten www-sovellusten JavaScript- ja HTML -vianselvitystä. Esimerkiksi Firefox-selaimelle on saatavilla Firebug-lisäosio, joka soveltuu mainiosti www-sivujen testaukseen. Firefox-selaimelle on myös saataville erilaisia testaustyökaluja, joilla voidaan tutkia HTTP Request/Response-rajapintaa. Testausvaiheessa on myös hyödynnetty sitä, että www-palvelimelle on luotu ns. Staging-hakemisto, jolle pääsyä on rajoitettu. Tämä on mahdollistanut www-sovelluksen testauksen rajatulla käyttäjäkunnalla.

### Android-sovellus

Kehitysvaiheessa sovellusta on testattu käyttäen hyväksi kehitysympäristöä. Android-sovelluskehitys tarjoaa hyvän testiympäristön ja se voidaan integroida osaksi Eclipse-alustaa. Android-sovelluksen testaus onnistuu helposti ADB-työkalua käyttäen, joka on testaus- ja vianselvitystyökalu Androidin sisältämälle Dalvik-virtuaalikoneelle. ADB:llä on helppo tutkia ohjelmaan tehtyjä sisäisiä tulostuksia sekä nähdä Java-ympäristölle tyyppillisiä poikkeusten käsittelyä.

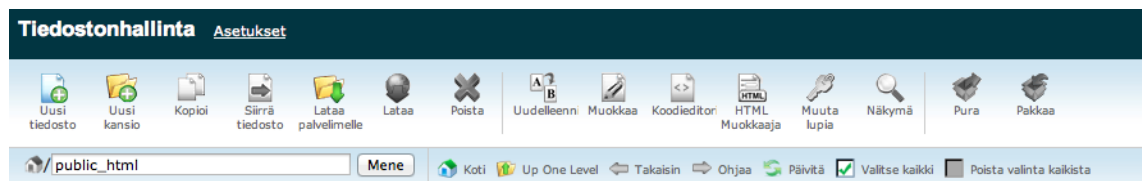
### Kenttätestaus

Testaus on perustunut enimmäkseen omaehtoiseen kokeiluun sekä nopeaan tuotanto-levitykseen. Web-sovellus on voitu jakaa aikaisemmin todetussa Staging-osiossa. Google Play -kauppa mahdollistaa myös nopean Android-sovelluksen jakamisen, koska Google ei tee kaupan ohjelmille käytännössä mitään tarkistuksia. Näin ollen uusi Android-versio saadaan jakeluun käytännössä vuorokaudessa siitä, kun se on ladattu sovelluskauppaan. Vertailun vuoksi iOS- ja Windows Phone -sovellusten jakaminen kestää noin yhden viikon, koska näille sovelluksille ajetaan automaattiset ja manuaaliset testit, jotka sovellusten pitää läpäistä ennen kauppaan pääsyä. OrientGames esi-versiota on ladattu erinäisistä sovelluskaupoista useita satoja kappaleita älypuhelin-alustaa kohden. On osoittautunut, että iOS-alusta on ollut näistä kaikista suosituin.

## 6 Julkaisu

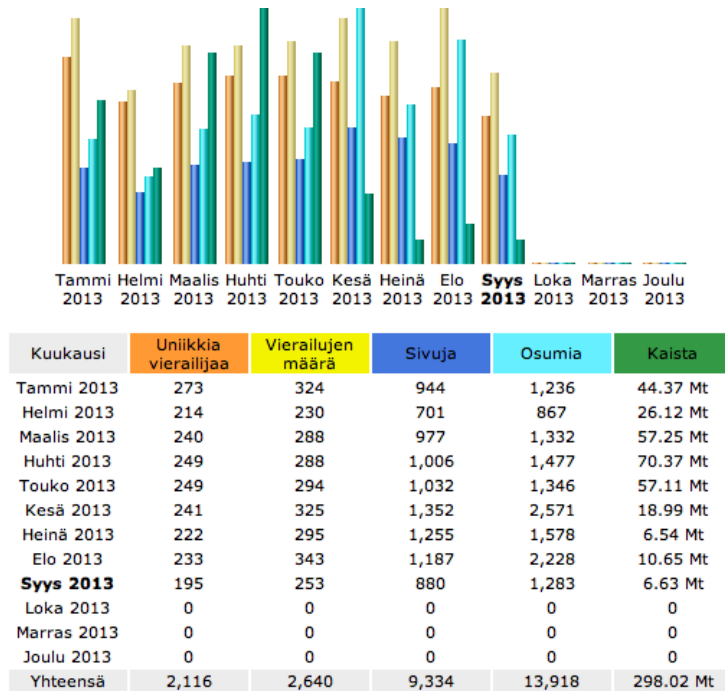
### 6.1 Web-hotelli

Varsinainen palvelu on toteutettu Web-hotellissa, joka on julkaistu osoitteessa [www.orientgames.com](http://www.orientgames.com). Käytetyn web-hotellin julkaisujärjestelmä ei perustuu mihinkään tyypilliseen web-julkaisujärjestelmään kuten esim. Joomla tai Drupal, vaan sivut siirretään palvelimelle, kuten ne siirrettäisiin oman palvelimen www-sivustoille (kuva 20).



Kuva 20. Web-hotellin tiedostonhallintakäyttöliittymä.

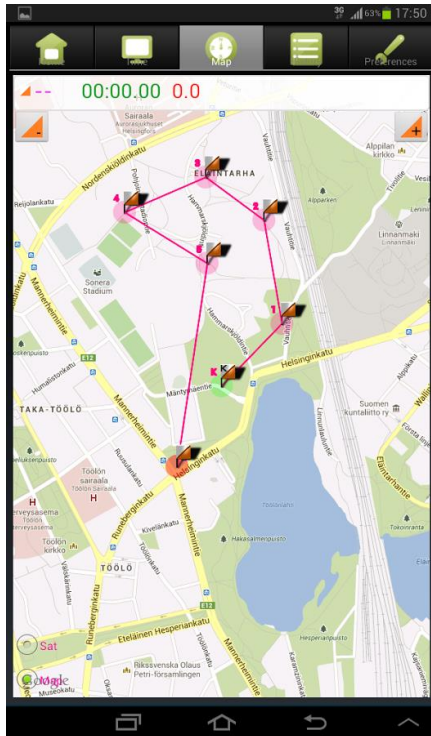
Web-hotellin hallintapaneeli mahdollistaa myös sovelluksen käytön seuraamisen sekä tietokannan tilankäytön kehittymisen. Kuvassa 21 on esimerkki, josta voidaan seurata www-sovelluksen käyttöastetta.



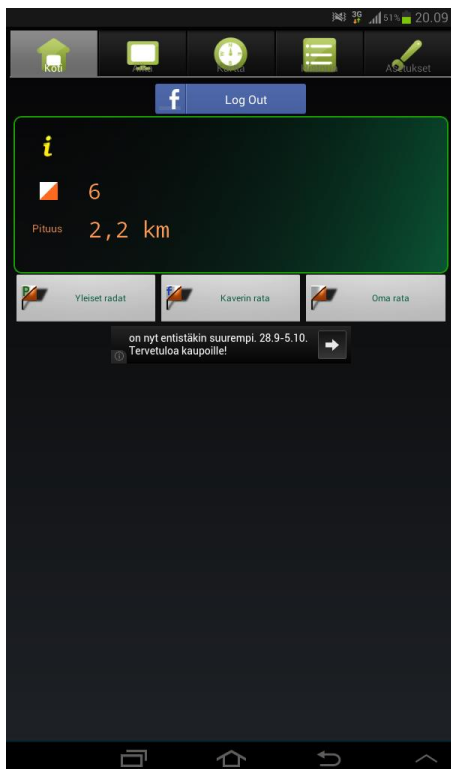
Kuva 21. Ote Web-hotellin liikennöintitilastosta.

## 6.2 Google Play -kauppa

Android-versio (kuvat 22 ja 23) tätä kirjoittaessa julkaistu Google Play -kaupassa. Koska olemassa oleva sovellus on jo aikaisemmin julkaistu, on uusi sovellus päivitetty vanhan päälle. Google Play -kaupassa riittää päivittää kuvaus sovelluksen uusista ominaisuuksista sekä näkymä sovelluksesta.



Kuva 22. Lopullisen Android-sovelluksen karttanäkymä.



Kuva 23. Uusi Home Activity-komponenttinäkymä.

Kuvassa 24 on esimerkki Google Play -sovelluskaupan kehittäjänäkymästä, jossa on määritelty sovelluksen julkaisun historiatietoa.

### What's New

1.7.0 Added facebook connection to use tracks created in www.orientgames.com. Cannot use own storage anymore.  
 1.7.1 Possible to use own track created in www.orientgames.com, facebook account identification  
 Miles support  
 Current track is cached for next opening  
 Settings tab  
 Usability tuning  
 1.7.2 New icons <http://creativecommons.org/licenses/by/3.0/>  
 Bug fixing  
 1.7.3 Major bug fix related to friend track loading

---

### Additional information

Updated	Size	Installs	Current Version	Requires Android	Content Rating
August 14, 2013	1.2M	100 - 500	1.7.3	2.3.3 and up	Low Maturity

**Contact Developer**  
[Visit Developer's Website](#)  
[Email Developer](#)  
[Privacy Policy](#)

Kuva 24. Näkymä OrientGame-sovelluksen historiasta Google Playssä.

## 7 Yhteenveto

Uusi sovellus on ollut käytössä elokuusta 2013 alusta lähtien, ja sille on tehty tarvittavat muutokset myös iPhone- ja Windows Phone -alustoille. Aika näyttää, miten sovellus saa menestyä. Sovelluksen latausmäärät eri sovelluskaupoissa eivät ole lisääntyneet muutosten myötä merkittävästi. Android-sovelluksen latausmäärät ovat olleet kaikista pienimmät, mikä saattaa johtua siitä, että ko. Android-alustalle on saatavilla hiukan vastaavanlaisia suunnistussovelluksia. Koska Android-alustalle on nopeinta julkaista ohjelmia, on se hyvä prototyyppialusta ja erilaiset korjaukset ovat nopeita julkaista.

Ratojen tekeminen www-sivustolla koetaan vielä aika vaikeaksi ja palautettu testihenkilöiltä on tullut siitä, että suunnistusrataa ei voi tehdä itse älypuhelimessa. Suunnistus lajina on maailmalla suhteellisen tuntematon laji, ja varsinaiset suunnistusharrastajat haluavat suunnistaa käyttäen mekaanista karttaa ja kompassia. Näin ollen suunnistajien piirissä sovellus ei tule saamaan suurta suosiota. Suunnistus on myös haasteellista ja osaamista vaativa laji.

Sovelluksen käyttöliittymä vaatisi myös paremman suunnittelun, koska nyt se on aika insinöörimäinen ja voi olla vaikeata hahmottaa lopulta, mistä on kysymys. Älypuhelinsovelluksen ja www-sovelluksen kytkeminen yhteen on ollut joka tapauksessa mielenkiintoinen ja opettavainen harjoitus. Olemassa olevaa koodikantaa ja ideaa voidaan hyödyntää muissa samantyyppisissä sovelluksissa, jotka vaativat paikkatietojen jakamista käyttäjille. Tämän idean pohjalle on jalostumassa paljon muita ideoita.

#### Jatkokehitys

Tehdyn sovelluksen rajoittavat tekijät ovat ehkä pakollisen Facebook-tunnistuksen käyttö omien karttojen tekemisessä. Jatkokehityksenä voisi tulla kysymykseen tarjota erilaisia tunnistusmenetelmiä, koska kaikki eivät käytä tai halua käyttää Facebookia. Oman tunnistusmenetelmän kehitys ei kuitenkaan tunnu luontevalta, koska tällöin törmätään huomattavaan työmäärän lisääntymiseen. Oman tunnistusmenetelmän luominen tarkoittaisi erilaisiin tietosuoja-asioihin perehtymistä sekä erittäin hyvään tietoturvasioiden ratkaisemisen sekä mahdollisesti kansallisten säädösten huomioimista. Oma tunnistusmenetelmä tarvitsisi myös mahdollisesti paljon luotettavaa tietovarastointia. Kuten aikaisemmin on jo mainittu, käyttäjätunnistus tarvitsee joka tapauksessa parantelua, koska tällä hetkellä sovelluksen tunnistus perustuu henkilön Facebook-sovelluksessa käyttämään nimeen ja samannimisiä ihmisiä voi olla Facebook-palvelussa rekisteröityneenä useita.

Toinen merkittävämpi kehityskohde voisi olla omien karttojen tekeminen itse älypuhelinsovelluksessa. Tämä saattaisi lisätä käyttäjien innostusta, koska tällöin koko sovellus olisi käytettävissä hyvin omasta älypuhelimesta eikä käyttäjätunnistusta tarvitsisi välttämättä tehdä. Tämän toiminnallisuuden rakentaminen vaatisi hyvää käyttöliittymäsuunnittelua, koska suunnistusradan piirtäminen suhteellisen pienellä älypuhelinnäytöllä on haastavaa.

## Lähteet

- 1 Suunnistus. 2013. Verkkodokumentti. Wikipedia.  
<<http://fi.wikipedia.org/wiki/Suunnistus>>. Luettu 11.5.2013.
- 2 Häggman Bjarne, Mäkinen Matti, Oikarinen Erkki. 1981. Suunnista Luontoon. Suunnistajan käsikirja. WSOY.
- 3 Grönroos Marko, 2013, Book of Vaadin, Vaadin 7 edition. Verkkodokumentti.  
<<http://www.vaadin.com/download/book-of-vaadin/vaadin-7/pdf/book-of-vaadin.pdf>>. Luettu 11.5.2013.
- 4 Node.js. Verkkodokumentti. <<http://www.nodejs.org>>. Luettu 11.5.2013.
- 5 Gauchat J.D. 2012. HTML5 for Masterminds. 2nd Edition. Minkbooks.
- 6 Stuttard Dafydd, Pinto Marcus, 2013. The Web Application Hacker's Handbook. Second Edition.
- 7 Svennerberg Gabriel. 2010. Beginning Google Maps API 3. Apress.
- 8 Forber Alan. 2013. The Joy Of PHP Programming.
- 9 Lindley Cody, St. Laurent Simon. 2010. jQuery Cookbook. O'Reilly.
- 10 Meier Reto, Wrox. 2012. Professional Android 4 Application Development, Wrox.
- 11 Graham Wayne. 2008. Facebook API. Developer Guide. Apress.
- 12 Suunnistuskartta. 2013. Verkkodokumentti. <<http://www.espoonakilles.fi>>. Luettu 10.7.2013.
- 13 Google Maps. 2013. Verkkodokumentti. <<https://maps.google.fi>>. Luettu 10.7.2013.
- 14 KML Tutorial. 2013. Verkkodokumentti.  
<[https://developers.google.com/kml/documentation/kml\\_tut](https://developers.google.com/kml/documentation/kml_tut)>. Luettu 13.6.2013.
- 15 Creative Commons -lisenssi. 2013. Verkkodokumentti.  
<<http://creativecommons.org/licenses/by/3.0/>>. Luettu 15.7.2013.
- 16 Google Play Developer -sivusto. 2013. Verkkodokumentti.  
<<http://developer.google.com>>. Luettu 21.10.2013.

## Esimerkki KML-tiedostosta

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://earth.google.com/kml/2.1">
  <Document>
    <name>Leppavaara, Espoo</name>
    <description />
    <Placemark>
      <name>K</name>
      <description />
      <Point>
        <coordinates>24.801560585426525,60.22275403417404</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>1</name>
      <description />
      <Point>
        <coordinates>24.804238404865273,60.2264272213685</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>3</name>
      <description />
      <Point>
        <coordinates>24.786943521137246,60.22877130679105</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>4</name>
```

```
<description />
<Point>
  <coordi-
nates>24.779240216846474,60.22612887119793</coordinates>
</Point>
</Placemark>
<Placemark>
  <name>5</name>
  <description />
  <Point>
    <coordi-
nates>24.789947595233926,60.227396840668604</coordinates>
</Point>
</Placemark>
<Placemark>
  <name>6</name>
  <description />
  <Point>
    <coordi-
nates>24.79662093126298,60.22426412110643</coordinates>
</Point>
</Placemark>
<Placemark>
  <name>0</name>
  <description />
  <Point>
    <coordi-
nates>24.80683478319169,60.22141884085814</coordinates>
</Point>
</Placemark>
</Document>
</kml>
```