

Mobiilisovelluksen teko eri alustoille

Timo Jurvelin

Kaupan ja kulttuurin toimialan opinnäytetyö
Tietojenkäsittelyn koulutusohjelma
Tradenomi

TORNIO 2012

TIIVISTELMÄ

KEMI-TORNION AMMATTIKORKEAKOULU

Koulutusohjelma:	Tietojenkäsittely
Opinnäytetyön tekijä(t):	Timo Jurvelin
Opinnäytetyön nimi:	Mobiilisovelluksen teko eri alustoille
Sivuja (joista liitesivuja):	20 (0)
Päiväys:	26.11.2012
Opinnäytetyön ohjaaja(t):	Yrjö Koskenniemi
<p>Perehdyin eri web-rajapintoihin ja olen selvittänyt miten hyvin ne tukevat erilaisia mobiili-käyttöjärjestelmiä. Rajapinnat auttavat ongelmassa, joka tulee alustojen erilaisesta tuesta tekniikoita kohtaan.</p> <p>Tutkin miten rajapinnat auttavat mobiilisovelluksen siirrettävyydessä toisille alustoille. Pohjana testauksessa käytän mobiilisovellusta, joka on tehty Androidille. Sovelluksen teossa käytän jQuery Mobilea, joka yhtenäistää web-standarditukea eri laitteiden selaimille. Sovelluksessa on käytössä myös PhoneGap. PhoneGap on sovelluskehys, jolla voi tehdä natiivisovelluksia monille mobiilialustoille.</p> <p>Web-rajapintojen tuki eri mobiililaitteille on aika hajanaista ja ominaisuudet eivät ole täysin samoja kaikissa alustoissa. Kuitenkin tuki uusille ja suosituimmille alustoille on hyvä. Sovelluksen siirrettävyys alustalta toiselle sovelluskehyyksen avulla oli vaivatonta. Täytyy kuitenkin ottaa huomioon, että sovellus oli pieni ja yksinkertainen, joten laajempi sovellus useammilla ominaisuuksilla voisi olla eri asia.</p>	
Asiasanat: mobiililaitteet, ohjelmointi, verkko-ohjelmointi	

ABSTRACT

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES

Degree programme:	Bachelor of Business Administration
Author(s):	Timo Jurvelin
Thesis title:	Making mobile application for different platforms
Pages (of which appendixes):	20 (0)
Date:	26.11.2012
Thesis instructor(s):	Yrjö Koskenniemi
<p>I studied different web interfaces in order to find out how well these interfaces support different mobile operating systems. Interfaces help to solve problems created by from different platform support in view of technologies.</p> <p>I studied how interfaces help move mobile application to different platforms. As a base for testing I used a mobile application made for Android. I used jQuery Mobile to do the application which unifies web standard support between different devices browsers. The application also uses PhoneGap. PhoneGap is an application framework which can be used to do native applications to multiple mobile platforms.</p> <p>Web interface support for different mobile devices is quite fragmented and features are not fully the same in all platforms. However, support for new and the most popular platforms is good. Moving the application to a different platform from another was quite easy by using the application framework. However, it has to be taken into account that the application was small and simple and therefore a larger application with more features could be a different thing.</p>	
Keywords: Mobile Devices, Programming, Network Programming	

SISÄLLYS

TIIVISTELMÄ.....	2
ABSTRACT.....	3
SISÄLLYS.....	4
1 JOHDANTO.....	5
1.1 Pohjustus.....	5
1.2 Tutkimuskysymys.....	5
1.3 Tutkimusmenetelmät.....	5
2 WEB-RAJAPINNAT MOBIILISELAIMILLE.....	7
2.1 Rajapintojen tarkoitus.....	7
2.2 Rajapinnat.....	8
2.3 Tuki eri alustoille.....	10
3 SOVELLUSKEHYSTEN TARKASTELU.....	12
3.1 Toiminta.....	12
3.2 Phoneygap.....	12
3.3 Sencha Touch.....	13
4 SOVELLUS VERSIOT.....	14
4.1 Esittely.....	14
4.2 Android.....	14
4.3 Windows Phone.....	16
5 POHDINTA.....	19
LÄHTEET.....	20

1 JOHDANTO

1.1 Pohjustus

Tässä opinnäytetyössä tutkin mobiilisovellusten tekoa eri alustoille. Mobiilisovelluskehitys on kovassa kasvussa vieläkin, mutta useat eri alustat ja alustoiden eri versiot vaikeuttavat sovelluskehittäjän työtä. Tilanne on johtanut siihen, että yhä enemmän haetaan rajapintoja, joiden avulla voitaisiin tehdä sovellus toimivaksi monille alustoille pienillä muutoksilla.

Aluksi tutkin teoriaa mobiilisovelluksiin liittyvistä rajapinnoista sekä kahta suosittua sovelluskehystä. Lopuksi toteutan sovellusversion Android- ja Windows Phone -alustoille. Tarkastelen toimivuutta, sekä hyödyllisyyttä kehittäjän näkökulmasta.

1.2 Tutkimuskysymys

Työssä minulla on tavoitteena tutkia, miten rajapinnat auttavat web-pohjaisen mobiilisovelluksen teossa. Tarkastelen asiaa kehittäjän näkökulmasta, mitä hyötyä kehittäjälle on käyttää web-rajapintoja. Tutkin myöskin raportissa sitä, miten sovelluskehikset auttavat mobiilisovelluksen siirrettävyydessä eri alustoille.

Rajapintojen tarkoitus olisi vähentää työn määrää kehitettäessä useille alustoille. Mahdollisia ongelmia ovat rajapinnoissa olevat bugit tai puutteet, mutta uusia versioita tulee usein, jotka toivottavasti korjaavat puutteita.

1.3 Tutkimusmenetelmät

Sovelluksen teossa käytän PhoneGap-sovelluskehystä, jolla sovelluksen siirrettävyys eri alustoille pitäisi olla helppoa. Web-rajapintana käytössä on jQuery Mobile, jolla korjataan selaimien puutteita, kuten myös erilaisuutta. Android -version teossa käytän Eclipseä ja Notepad++:aa. Sovellusversion teossa Windows Phonelle käytän Visual Studio 2010 Express for Windows Phone -kehitysalustaa.

Tarkkailen asiaa siis kehittäjän näkökulmasta, jotta saan selville parhaita mahdollisia tapoja suorittaa kehitystyötä. Mobiilisovellukset ovat huimassa kasvussa ja ovet ovat avoinna monille yrittäjille, joten mobiilialan tutkimus on hyvin ajankohtaista.

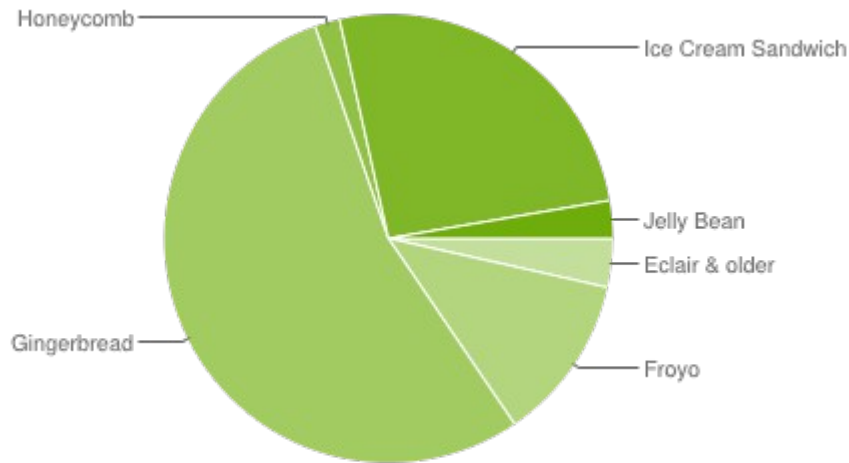
2 WEB-RAJAPINNAT MOBIILISELAIMILLE

2.1 Rajapintojen tarkoitus

Rajapintojen tarpeellisuus on muodostunut, kun eri selaimet eivät suorita koodia täysin samalla standardilla tai jokin ominaisuus puuttuu tietystä selaimesta. Tällöin tehtävään tarvitaan jokin korvaava suoritustapa. Varsinkin juuri mobiililaitteiden selaimissa on eroa ominaisuuksissa, yhdessä oleva tärkeä ominaisuus voi puuttua kokonaan toisesta. Esimerkiksi jQuery Mobilen ohjeissa kerrotaan, miten osa selaimista tukee css-komentoa `position: fixed`, mutta ne, jotka eivät tue jäävät ilman tuota ominaisuutta, koska rajapinta toteuttaa tuon eri lailla niille selaimille, jotka eivät tue sitä (jQuery Mobile 2012a, hakupäivä 8.11.2012).

Käyttämällä erilaisia rajapintoja vältetään mahdollista koodin uudelleen kirjoittamista kun jokin selain ei tuekaan samaa käskyä kuin toinen. Ominaisuuden puuttuessa tarvitaan kiertotie suorittaa haluttu toiminto, jos mahdollista. Rajapinnat hoitaa tuon kiertotien teon kun jokin selain ei tue ominaisuutta ja siten kehittäjällä on vähemmän työtä.

Ilman rajapintoja työmäärä olisi huomattavasti suurempaa, koska jouduttaisiin kirjoittamaan koodia uudestaan eri selaimille, jotta kaikki toimisi kuten pitää. Rajapintojen tarpeellisuus kasvaa vain lisää aina, kun tulee uusia sovelluksia ja sovellus versioita. Usein vanhat versiot jäävät monille laiteille ainoaksi mahdollisuudeksi päivitysten loppumisen takia. Kehittäjät joutuvat sitten kehittämään niillekin sovelluksia, koska joskus suurikin osa käyttäjistä voi olla vanhan version varassa (kuva 1).



Kuva 1. Erittely kaavio (Android 2012, hakupäivä 8.11.2012)

2.2 Rajapinnat

Selaimien erilaisuus ja puutteet tiedostetaan yleisesti, joten myös rajapintojen määrä on melko suuri. Valitsen esimerkiksi kaksi kohtuullisen suosittua web-rajapintaa, jotka on tehty mobiiliselaimia varten. Ensimmäinen niistä on Sencha Touch, jossa käytetään Javascriptiä sivun ulkoasun ja toiminnallisuuden tekoon. Sencha Touchia voi käyttää myös natiivisovelluksen tekoon, mutta nyt keskityn vain sen ulkoasun teko- ja toimintojen teko-osaan. Kerron lisää Sencha Touchin mahdollisuuksista myöhemmin sovelluskehys osiossa. Toiseksi valitsin jQuery Mobilen, joka on hyvin yleinen ja joka on myös käytössä esimerkkisovelluksessa. JQuery Mobilea käytetään, kuten normaalissa html- ja Javascript -kehityksessä eli työskentely tapahtuu lähinnä html-tiedostossa.

Sencha Touch on hyvin suosittu varsinkin nopeuden takia. iPhone-kehityksessä Sencha Touch varsinkin on suosittu, koska alusta on nopea ja ei siinä ole niin suuria nopeuseroja laitteiden välillä kuin Androidilla. Koska Android ja iPhone käyttävät molemmat WebKittiin pohjautuvaa selainta, on yhteensopivan rajapinnan teko Androidille ja iPhoneille helpompaa kuin ei webkittiin perustuville selaimille (Ableson 2009, Hakupäivä 8.11.2012). WebKit on internetselaimille tarkoitettu avoimen lähdekoodin ydin (WebKit 2012. Hakupäivä 25.11.2012).

Kehitys Sencha Touchissa tapahtuu Javascript-tiedostossa. Html-tiedosto sisältää vain linkin Javascript-tiedostoon, jossa sivun toiminnallisuus ja sisältö ovat. Kehityskieli ei ole perus-javascriptiä vaan Sencha Touchin omaa rakennetta ja komentoja. Tämän takia

tässä on hiukan isompi opettelukynnys kuin esimerkiksi jQuery Mobilessa. Ei jQuery Mobilen koodikaan ole rakenteiltaan ja komennoiltaan täysin samanlaista kuin perus-Javascript, mutta se on kuitenkin hyvin samankaltaista.

jQuery Mobile on myöskin suosittu rajapinta. Yksi suosion syistä on jQuery, jota käytetään jQuery Mobilessa. jQuery on hyvin suosittu ja yleisessä käytössä oleva Javascript-rajapinta tietokoneiden selaimia varten. Tästä syystä kehittäjillä on helppo siirtyä käyttämään jQuery Mobilea, koska aiempi tietokoneiden selaimia varten tehdyistä sivuista saatu kokemus, auttaa omaksumaan jQuery Mobilen käytön nopeammin.

jQuery Mobilessa kehitys tapahtuu lähes samaan tapaan kuin perus html- ja Javascript-kehityksessä. Tarvitaan vain jQuery Mobilen Javascript- ja css-tiedostot, sekä jQuery Javascript-tiedostot, että voidaan aloittaa jQuery Mobilen käyttö. Tiedostot linkitetään html-tiedoston alkuun, jonka jälkeen voidaan aloittaa kehitys. Taulukossa näkyy html sivun koodi, josta aloitetaan sivun teko (Taulukko 1). jQuery Mobile kuitenkin antaa tuen useampien alustojen selaimille, joten ei joudu itse korjaamaan selainten eroavaisuuksien aiheuttamia ongelmia (jQuery Mobile 2012c, Hakupäivä 8.11.2012). Toki mikään rajapintakaan ei ole täydellinen, mutta silti kuitenkin oikea suunta, että kehittäminen helpottuisi, ja ennen kuin jonkunlaiset pitävät standardit saadaan aikaiseksi. Nytkin on kyllä standardeja, mutta pienetkin erot selainten koodin suorituksessa rikkoo saman komennon toimivuuden selaimien välillä.

Taulukko 1. Peruspohja jQuery Mobile html-sivusta (jQuery Mobile 2012b, Hakupäivä 8.11.2012)

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.css" />
  <script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
  <script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.js"></script>
</head>
<body>

<div data-role="page">

  <div data-role="header">
    <h1>My Title</h1>
  </div><!-- /header -->

  <div data-role="content">
    <p>Hello world</p>
  </div><!-- /content -->

</div><!-- /page -->

</body>
</html>
```

2.3 Tuki eri alustoille

Olennaista rajapinnoissa on se, mitä alustoja ne tukevat. Valinta menee tietysti kohderyhmän mukaan, mutta suosituimmat alustat kannattaa ottaa huomioon siinäkin tapauksessa, että pääalusta olisi jokin muu. Esimerkeiksi valitsemani jQuery Mobile ja Sencha Touch tukevat molemmat tämän hetken suosituimpia alustoja, Androidia ja iPhonea. Tosin kaikki tämän hetken järkevät rajapinnat tukevat Androidia ja iPhonea. Mahdollinen nouseva alusta, Windows Phone, kannattaa myös huomioida ja seurata sen kehitystä.

jQuery Mobilen tukemat tärkeimmät alustat mobiilikehitystä ajatellen (jQuery Mobile 2012c, Hakupäivä 8.11.2012):

- Android 2.1-2.3
- Android 3.2 (Honeycomb)
- Android 4.0 (ICS)
- Android 4.1 (Jelly Bean)

- Apple iOS 3.2-6.0
- Windows Phone 7-7.5.

Tuki löytyy myös monille pienemmille alustoille sekä tietokoneen internetselaimille, mutta otin esille vain oleellisimmat ja suosituimmat alustat tällä hetkellä.

Sencha Touchin tukemiin alustoihin kuuluu Android 2.3+ ja Apple iOS 4+ (Sencha 2012a, Hakupäivä 8.11.2012). Tuki löytyi muutamille muillekin alustoille, mutta keskityn niihin kahteen, koska ne ovat tällä hetkellä suosituimpia alustoja.

Edellä olevasta voi tehdä päätelmän, että jQuery Mobilessa on tuki useammille alustoille, mutta se ei lopulta ole kovin merkittävää, kun ottaa huomioon sen, että tuki kahdelle tärkeimmälle löytyy. Windows Phonon suosio on vielä pieni arvoitus, mutta tulevat kuukaudet osoittavat, miten Windows Phonelle käy. Sencha Touchiin ollaan tekemässä kuitenkin myös tukea Windows Phone 8 -alustalle (Sencha 2012b, Hakupäivä 8.11.2012). Huomion arvoista on kuitenkin se, että Sencha Touch keskittyy selvästi pienempään määrään tuettuja alustoja kuin jQuery Mobile. Joten, jos haluaa tuen useammille ja vanhemmille laitteille, kannattaa harkita enemmän jQueryä kuin Sencha Touchia. Mutta jos tukea ei tarvitse vanhoille alustoille, voi miettiä onko Sencha Touch parempi vaihtoehto, koska sirpaloitumista eri alustojen tukemisen takia ei ole niin paljon kuin jQuery Mobilessä. Mahdollisesti kehitys on myös nopeampaa kun keskitytään pienempään määrään alustoja.

3 SOVELLUSKEHYSTEN TARKASTELU

3.1 Toiminta

Sovelluskehyksiä tarvitaan, jotta saadaan yhteys alustan toiminnallisuuksiin käyttämättä natiivikoodia. Sovelluskehys hoitaa kommunikoinnin alustan toimintojen kanssa, esimerkiksi vaikka kännykässä olevan kameran käyttämisen kirjoittamatta yhtään natiivikoodia (PhoneGap 2012, Hakupäivä 8.11.2012). Sovelluskehukset ovat oleellisia juuri siinä, kun kehityksessä halutaan käyttää selainpohjaista kehitystä ja kuitenkin tehdä natiivi-sovellus, jonka voi asentaa mobiililaitteeseen.

Sovelluskehyksiä käytetään yleensä yhdessä jonkin web-rajapinnan kanssa, jotta tuki eri selaimien kanssa on mahdollisimman hyvä. Web-pohjainen sovelluskehitys on tuttua web-kehittäjille, joten sovelluskehysten käyttäminen yhdessä web-rajapinnan kanssa mahdollistaa nopean kehityksen eri laitteille. Muuten pitäisi opetella joka mobiilialustan koodausta erikseen ja myös tehdä omat erilliset koodit kaikille alustoille. Helppous ja nopeus on saanut kehittäjät kiinnostumaan sovelluskehysten käyttämisestä, tietysti aiemman osaamisen hyödyntäminen on aina kannattavaa, jos se on mahdollista ilman, että käyttäjäkokemus kärsii.

3.2 Phonegap

Phonegap on yksi suosituimmista sovelluskehyksistä. Alustan kehitys on ollut nopeaa ja uusia versioita tulee usein. Mikä on hyvä asia, koska alusta on keskeneräinen ja uutta tuettavaa tulee koko ajan lisää. Sanon keskeneräinen, koska tuki ei ole vielä täysin samanlaista kaikille alustoille. Osalle alustoista ei välttämättä ole kaikkia samoja toiminnallisuuksia kuin toisille on. Kuitenkin tämä on vielä kehittyvä ja kasvava suuntaus, joten tuen uskoisi paranevan ajan mittaan. Adobe osti PhoneGapin takana olleen Nitobi-yhtiön, joten luottamusta web-pohjaiseen kehitykseen löytyy isoiltakin yhtiöiltä (Grannell 2011, Hakupäivä 8.11.2012).

Phonegap hoitaa kännykän toiminnallisuudet html-tiedoston avulla ilman, että kehittäjän tarvitsee käyttää itse natiivikoodia. Kehittäjä siis tekee oman osuutensa käyttämällä html:ää, css:ää ja Javascriptiä. Html-sivulle vain linkitetään PhoneGap

Javascript-tiedosto, jonka kautta käskyt natiivitoiminnoille lähtevät.(PhoneGap 2012, Hakupäivä 8.11.2012)

PhoneGapin internetsivuilta löytyvä dokumentointi on hyvin tehty ohjelmointirajapintojen osalta ja uusia alustoja lisätään, kun niitä tulee. Täysi tuki PhoneGapin ohjelmointirajapinnoille löytyy vain iPhone, Android ja Windows Phone alustoista (PhoneGap 2012, Hakupäivä 8.11.2012). Eri versiot PhoneGapista on listattuna documentti-sivulla ja jokaiselle versiolle on omat documentoinnit tallessa. Siirryttäessä uuteen versioon kannattaa aina tarkistaa asennus ohjeet PhoneGapin sivuilta, jotta mahdolliset muutokset asennuksessa eivät aiheuta turhaan ylimääräisiä ongelmia. PhoneGapin sivuilta löytyy myös Tools-sivu, johon on listattu erilaisia PhoneGapille tehtyjä tai sen kanssa yhteen sopivia lisäosia.

3.3 Sencha Touch

Otin Sencha Touchin toiseksi sovelluskehysesimerkiksi sen suosion takia. Aiemmin kerroin Sencha Touchista web-rajapintakäytössä, jossa sitä voikin käyttää esimerkiksi PhoneGapin kanssa, mutta Sencha Touchilla voi myös pakata ohjelmia Sencha Packagerin avulla (Sencha 2012c, Hakupäivä 8.11.2012). Senchalla on myös ohjelmointirajapinta Androidin ja iPhonen natiivitoimintoihin, kuten esimerkiksi kameraan (Sencha 2012d, Hakupäivä 8.11.2012).

Sencha nimen alta löytyy myös paljon muita kehitysvälineitä, joista osa on täysin maksullisia. Kuten esimerkiksi Sencha Architect, joka on visuaalinen kehitystyökalu, jonka tarkoitus on helpottaa suunnittelijan työskentelyä yhdessä kehittäjän kanssa (Sencha 2012e, Hakupäivä 8.11.2012).

Kehitystyökaluna Senchaa on kehitetty paljon, mutta siirtyminen Senchan käyttäjäksi esimerkiksi web-kehittäjäpohjalta vaatii paljon uuden opettelua ja sisäistämistä. Monet valitsevatkin mielellään tutun ja turvallisen jQueryn. Myös alustan käyttöönotto vaikuttaa hiukan monimutkaisemmalta kuin vaikka verrattuna PhoneGapin käyttöönottoon, joka on ohjeistettu selkeästi. Lopulta kyse on kuitenkin kehittäjän tarpeesta ja osaamisesta, kumman valitsee vai valitseeko kumpaakaan. Valmista osaamista kannattaa aina kuitenkin hyödyntää jos se vaan tuottaa parhaan tuloksen.

4 SOVELLUS VERSIOT

4.1 Esittely

Tarkoitus on kokeilla miten siirrettävyys eri alustoille onnistuu käytännössä. Pohjaksi teen Android version ja siirrän sen Windows Phonelle. Teen tarvittavia korjauksia ja arvioin miten paljon ohjelmistokehitys helpottaa sovelluksen tekoa useille alustoille.

Sovellukseksi teen pienen demon, jolla testaan siirrettävyyden helppoutta. Nimeksi valitsin My Demo, jonka ajattelin olevan riittävän selkeä. Sovellus itsessään on pieni muistio, joka tallentaa näkyville koordinaatit, sekä kommenttitekstin. Tieto ei jää tietokantaan, joten sovelluksen sulkemisen jälkeen tiedot häviää.

Työvälineinä käytän Eclipseä ja Notepad++:aa Android -version teossa. Windows Phonelle sovellusta siirrettäessä ja muokattaessa käytän Visual Studio 2010 Express for Windows Phone. Testauksen suoritan emulaattorissa Windows Phone -versiossa.

4.2 Android

Aloitin välineiden asennuksen menemällä PhoneGapin -sivuille. Valitsin uusimman version, joka oli 2.2.0. Hakeuduin Getting Started with Android -sivulle ja aloin hakemaan tarvittavia kehitystyökaluja. Hakeuduin ensin Eclipsen -sivuille hakemaan kehitysalustaa, mutta huomasin että Android SDK ADT Bundle for Windows on tullut Androidin sivuille, joten otin sen. Android SDK ADT Bundle for Windows sisältää Eclipsen ja ADT-pluginin, jotka olisi muuten pitänyt hakea erikseen (Android 2012b, Hakupäivä 23.11.2012).

PhoneGap 2.2.0 asennusohjeet erosivat paljon vanhempien versioiden ohjeista. Ohjeita oli vaikea seurata ilman aiempaa kokemusta. Lopulta jouduin toteamaan, että ohjeet ovat liian puutteelliset. Yksi komentona käytetty sana oli kirjoitettu väärin, sekä ohjeiden 4. Setup New Project -vaiheessa puuttui ohje Windows -alustaa varten (PhoneGap 2012c, Hakupäivä 23.11.2012). Päädyin sitten edelliseen versioon, jonka asennus oli helppoa ja myös aiempi kokemus auttoi asentamista. Käytän siis sovellusdemon teossa PhoneGapin versiota 2.1.0.

Lopuksi lisään vain jQuery Mobile 1.2.0 tiedostot ja jQuery 1.8.2 Javascript -tiedoston projektin www-kansioon Eclipsessä. Nyt työkalut on valmiina ja voin aloittaa sovelluksen teon.

Ensin hain jonkinlaista tekstin lisäys ominaisuutta sovellukseen. Koska käytössä on jQuery Mobile, etsin sitä niiden sivuilta. Käytin taulukossa näkyvää koodia, jota hieman muokkasin (Taulukko 2). Hain lisäksi koodin painiketta varten, joka on seuraavanlainen `<input type="submit" value="Submit Button" />` (jQuery Mobile 2012e, Hakupäivä 25.11.2012). Koin yllättävän paljon vaikeuksia tuon kanssa ja kokeilin monia erilaisia tapoja saada kirjoitettu teksti näkyviin napin painamisen jälkeen. Lopulta hyvin yksikertainen jQuery koodi tuli vastaan, jolla oli helppoa käsitellä tekstikentän tekstiä. Omaksi koodin pätkäksi muodostui `$("#basic").val()`, johon hain mallia jQuery:n sivulta (jQuery 2012, Hakupäivä 25.11.2012).

Taulukko 2. Tekstikenttä (jQuery Mobile 2012d, Hakupäivä 25.11.2012)

```
<label for="basic">Text Input:</label>
<input type="text" name="name" id="basic" data-mini="true" />
```

Sovelluksessa sijainnin näyttämiseen käytän PhoneGapin Geolocation ohjelmistorajapintaa. Omaksi koodiksi muodostui taulukossa näkyvä koodi, johon hain mallia PhoneGapin sivulta (Taulukko 3).

Taulukko 3. Sijainnin haku (PhoneGap 2012d, Hakupäivä 25.11.2012)

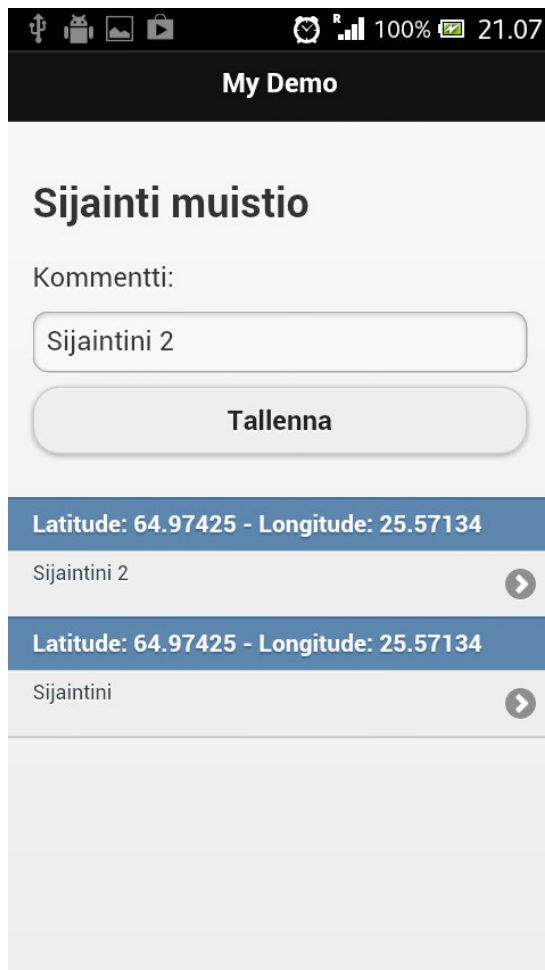
```
var onSuccess = function(position){
  var resultLati = (position.coords.latitude).toFixed(5);
  var resultLongi = (position.coords.longitude).toFixed(5);
  var onError = function(){
    alert("Sijaintia ei pystytty määrittämään");
  }
  navigator.geolocation.getCurrentPosition(onSuccess, onError);
```

Tekstin ja koordinaattien näyttämiseen käytin jQuery Mobilen sivulta löytyvää listview ominaisuutta (Taulukko 4). Tallennuksessa käytin jQuery:n `.prepend` -käskeyä, joka lisää tiedon määriteltyyn elementtiin ensimmäiseksi. Kun talletin muistioon, ei jQuery Mobilen tyylimäärittelyt tulleet mukaan, vaan talletus näkyi tyyllittelemättömänä listana. Se korjautui lisäämällä `.listview("refresh")` komennon loppuun.

Taulukko 4. Lista (jQuery Mobile 2012f, Hakupäivä 25.11.2012)

```
<ul data-role="listview">
  <li><a href=""></a></li>
</ul>
```

Tässä vaiheessa sovellus on valmiina siirrettäväksi Windows Phone 7 -alustalle. Ohjelman näkymä valmiissa Android-versiossa (Kuva 2).



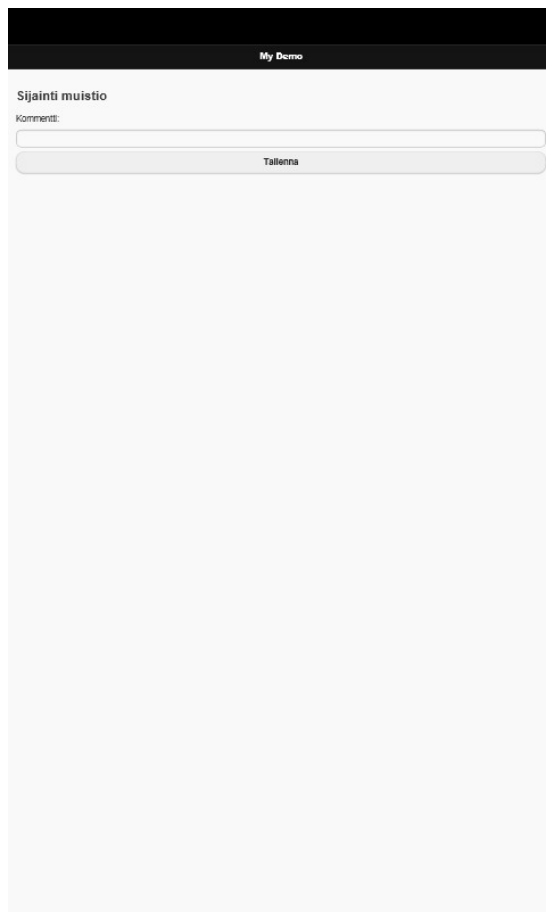
Kuva 2. Sovelluksen Android-versio.

4.3 Windows Phone

Ensin asensin Microsoft Visual Studio 2010 Express for Windows Phone -kehitysalustan, jonka jälkeen asensin PhoneGap 2.1.0 version. PhoneGapin Windows Phone -versio löytyi samasta paketista kuin Android -versiokin. Visual Studio 2010 Express for Windows Phone -sovelluksen kautta tein asennuspaketin, jolla asennus oli helppoa.

Seuraavaksi kopioin Android -projektin www-kansion ja siirsin tarvittavat tiedostot Windows Phone -projektin kansioon. Jotta sain uudet tiedostot näkyviin Windows Phonen -sovellustyökalussa, minun piti klikata Show All Files kuvaketta Solution Explorer -valikossa. Sen jälkeen uudet tiedostot piti liittää projektiin painamalla hiiren oikeanpuoleista nappia ja valitsemalla Include In Project. Kuvatiedostoista piti muuttaa Build Action valinta Resourcesta Contentiksi. Ilman tuota muutosta kuvatiedostot eivät näy sovelluksessa.

Ensimmäinen käynnistys emulaattoriin suoraan kopioituilla Androidin www-kansion tiedostoilla ei toiminut täysin. Näkymän skaalaus oli pielessä, joten sovellusnäkyvä oli liian pieni (Kuva 3). Asia korjautui kopioimalla meta-koodi PhoneGapin Windows Phone -version index.html -tiedostosta (Taulukko 5). Tuolla pienellä korjauksella sovellus tuli toimivaksi Windows Phonessa.

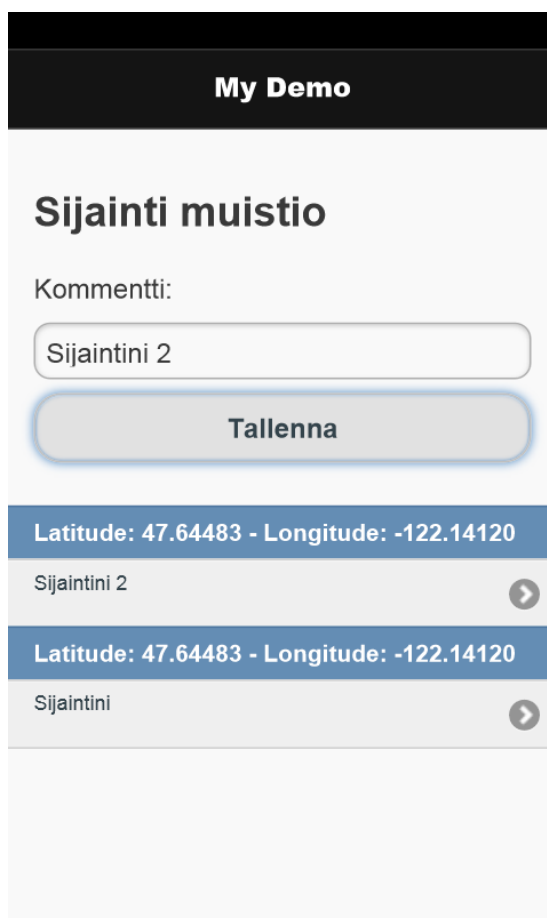


Kuva 3. Windows Phone -versio ilman muokkauksia.

Taulukko 5. Sovellusnäkömön skaalaus Windows Phonessa.

```
<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height, target-densitydpi=device-dpi" />
```

Sovellus oli aika yksinkertainen ja pieni, joten pelkkä skaalauksen korjaus riitti muokkaukseksi Windows Phone -versiossa. Sovelluksen toiminnallisuudet toimivat samoin kuin Android -versiossa. Kuvassa sovelluksen päänäkymä, joka näyttää samalta kuin Android -versiossa (Kuva 4). Sovelluksen siirrettävyys eri alustalle pienellä sovelluksella onnistui siis kohtuu helposti.



Kuva 4. Sovelluksen Windows Phone -versio.

5 POHDINTA

Aihe olisi aika laaja jos keskittyisi moniin eri web-rajapintoihin ja sovellukehyksiin. Kuitenkin itselle rajausta muutamien rajapintoihin oli tarpeellista, jotta työmäärä ei kasva liian suureksi. Itsellä aikataulun seuraaminen oli hiukan vaikeaa ja jouduin tekemään aika tiiviseen tahtiin työtä lopulta.

Huomasin työtä tehdessä miten nopeasti kaikki muuttu sovellusalalla. Uusia versioita tulee usein ja ohjeet muuttuu samalla. Itselle oli yllättävää miten paljon asennusvaihekin saattoi muuttua yhtäkkiä uudessa versiossa. Tässä huomasin miten sovelluskehitys alalla täytyy kokoajan ylläpitää omaa osaamistaan. Tosin vaikuttaa myöskin valitseeko työkaluiksi uuden kehittyvän alustan, vai jonkin vanhemman, joka ei ole enää niin suuressa muutoksessa.

Sovellusta tehdessä huomasin taas miten yksinkertaisessakin asiassa joutuu testaamaan ja hakemaan tietoa, ennenkuin tajuaa miten yksinkertaista se olikaan lopulta. Sovelluksen valmiiksi saannin jälkeen siirto toiselle alustalle meni helposti tosi pienellä muutoksella. Tuossa web-rajapinta ja sovellukehys hoitivat tehtävänsä hyvin. Voin todeta, että pienimuotoisessa sovelluksessa jQuery Mobile ja PhoneGap hoitivat tehtävänsä kuten pitikin. Laajemman sovelluksen ollessa kyseessä tulos voi kuitenkin olla erilainen. Kannattaakin aina arvioida kehitysvälineet sovelluksen laajuuden mukaan. Kokeilla vaikka nopeassa demossa tärkeimmät ominaisuudet ja välineiden käytettävyys.

LÄHTEET

- Ableson, Frank 2009. Android and iPhone browser wars, Part 1: WebKit to the rescue. Hakupäivä 8.11.2012.
<<http://www.ibm.com/developerworks/opensource/library/os-androidiphone1/>>
- Android 2012a. Platform Versions. Hakupäivä 8.11.2012.
<<http://developer.android.com/about/dashboards/index.html>>
- Android 2012b. Get the Android SDK. Hakupäivä 23.11.2012
<<http://developer.android.com/sdk/index.html>>
- Grannell, Craig 2011. Adobe buys PhoneGap creator. Hakupäivä 8.11.2012.
<<http://www.netmagazine.com/news/adobe-buys-phonegap-creator-111440>>
- jQuery 2012. .val(). Hakupäivä 25.11.2012.
<<http://api.jquery.com/val/>>
- jQuery Mobile 2012a. Fixed toolbars. Hakupäivä 8.11.2012.
<<http://jquerymobile.com/demos/1.2.0/#/demos/1.2.0/docs/toolbars/bars-fixed.html>>
- jQuery Mobile 2012b. Getting Started with jQuery Mobile. Hakupäivä 8.11.2012.
<<http://jquerymobile.com/demos/1.2.0/#/demos/1.2.0/docs/about/getting-started.html>>
- jQuery Mobile 2012c. Mobile Graded Browser Support. Hakupäivä 8.11.2012.
<<http://jquerymobile.com/gbs/>>
- jQuery Mobile 2012d. Mini sized elements. Hakupäivä 25.11.2012.
<<http://jquerymobile.com/demos/1.2.0/docs/forms/docs-forms.html>>
- jQuery Mobile 2012e. Button basics. Hakupäivä 25.11.2012.
<<http://jquerymobile.com/test/docs/buttons/buttons-types.html>>
- jQuery Mobile 2012f. Listviews. Hakupäivä 25.11.2012.
<<http://jquerymobile.com/test/docs/lists/docs-lists.html>>
- PhoneGap 2012a. Camera. Hakupäivä 8.11.2012.
<http://docs.phonegap.com/en/2.2.0/cordova_camera_camera.md.html#Camera>
- PhoneGap 2012b. Supported Features. Hakupäivä 8.11.2012.
<<http://phonegap.com/about/feature>>
- PhoneGap 2012c. Getting Started with Android 2.2.0. Hakupäivä 23.11.2012.
<http://docs.phonegap.com/en/2.2.0/guide_getting-started_android_index.md.html#Getting%20Started%20with%20Android>
- PhoneGap 2012d. Geolocation. Hakupäivä 25.11.2012.
<http://docs.phonegap.com/en/2.1.0/cordova_geolocation_geolocation.md.html#Geolocation>
- Sencha 2012a. Sencha Touch 2 Device Support. Hakupäivä 8.11.2012.
<<http://www.sencha.com/products/touch/features/>>
- Sencha 2012b. Sencha Touch with Windows Phone 8. Hakupäivä 8.11.2012.
<<http://www.sencha.com/blog/sencha-touch-with-windows-phone-8>>
- Sencha 2012c. Packager. Hakupäivä 8.11.2012.
<http://docs.sencha.com/touch/2-1/#!/guide/native_packaging>
- Sencha 2012d. Camera. Hakupäivä 8.11.2012.
<<http://docs.sencha.com/touch/2-1/#!/api/Ext.device.Camera>>
- Sencha 2012e. Sencha Architect. Hakupäivä 8.11.2012.
<<http://www.sencha.com/products/architect/>>
- WebKit 2012. The WebKit Open Source Project. Hakupäivä 25.11.2012.
<<http://www.webkit.org/>>