

Web-ohjelmiston sovittaminen mobiilialustalle - Case: Viope

Juuso Soikkeli

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

5.11.2014



Tekijä tai tekijät Juuso Soikkeli	Ryhmätunnus tai aloitusvuosi 2011
Raportin nimi Web-ohjelmiston sovittaminen mobiilialustalle - Case: Viope	Sivu- ja liitesivumäärä 18
Opettajat tai ohjaajat Jukka Juslin	
<p>Projektin tavoitteena oli luoda kaksi eri mobiiliversiosta Viope-oppimisjärjestelmästä. Mobiiliapplikaatioiden kohderyhmänä ovat Viopea käyttävät opiskelijat. Opettajien työkaluja mobiiliapplikaatioihin ei projektin puitteissa rakennettu.</p> <p>Projektin alkaessa Android-applikaatio oli jo lähes beta-julkaisuvalmis, joten projektissa toteutettiin käytännössä vain Windows 8.1-applikaation kehitys ja julkaisu.</p> <p>Projektin tuloksena syntyi valmis ja julkaistu Windows 8.1-versio sekä beta-vaiheen Android-versio, jota ei vielä tässä vaiheessa julkaistu käyttäjille. Lisäksi projektin tuloksena saatiin huomattava määrä tietoa eri mobiilialaustoiden toiminnallisuuksista, jota voidaan hyödyntää tulevaisuudessa uusia mobiilijärjestelmiä kehitettäessä.</p>	
Asiasanat Windows 8.1, Android, mobiili, C#, Java, XAML	

Degree programme in Information Technology

<p>Authors</p> <p>Juuso Soikkeli</p>	<p>Group or year of entry</p> <p>2011</p>
<p>The title of thesis</p> <p>Adapting a Web Application to Mobile Platforms – Case: Viope</p>	<p>Number of report pages and attachment pages</p> <p>18</p>
<p>Advisor(s)</p> <p>Jukka Juslin</p>	
<p>The goal of the project was to implement two separate mobile versions of the Viope Learning Platform. The target audience for these mobile applications is students using Viope. Features for teachers were not implemented in the mobile applications within the project's time frame.</p> <p>At the beginning of the project, the Android version was close to being ready and released as a beta-version. Therefore the project mainly included the development and publishing of the Windows 8.1-version.</p> <p>The project resulted in completing and publishing the Windows 8.1 application, as well as an unreleased beta-version of the Android application. As an additional result of the project, a considerable amount of information regarding the functionalities of different mobile platforms was gathered, that may be useful in the future for developing new mobile applications.</p>	
<p>Key words</p> <p>Windows 8.1, Android, mobile, C#, Java, XAML</p>	

Sisällys

1	Johdanto	1
1.1	Viope Oy	1
1.2	Tavoitteet.....	1
1.3	Opinnäytetyön rajaus	2
1.4	Käsitteet.....	2
2	Tietoperusta	4
2.1	Android.....	4
2.2	Windows 8.1	5
2.3	MVVM.....	5
3	Viope-oppimisjärjestelmän sovittaminen mobiiliympäristöön	7
3.1	Sovelluksen rakenne.....	7
3.2	Kommunikointi palvelimen kanssa.....	7
3.3	Käsinkirjoituksen tunnistus.....	9
3.4	Kauppa.....	9
3.5	Applikaation UX-parannusprosessi	11
4	Windows 8.1-applikaation julkaisuprosessi	12
5	Web-ohjelmiston sovittaminen mobiiliympäristöön – Android vai Windows 8.1? ..	13
5.1	Android.....	13
5.2	Windows 8.1	13
5.3	Johtopäätökset.....	14
6	Yhteenveto	16
6.1	Tulokset	16
6.2	Jatkokehitys	17
6.3	Oma oppiminen	17
	Lähteet.....	18

1 Johdanto

Viope-oppimisjärjestelmä on internet-selaimessa toimiva ohjelmisto, jonka avulla opiskelijat voivat suorittaa ohjelmoinnin sekä matematiikan kursseja. Kurssit koostuvat teorian materiaalista, monivalintatehtävistä, avoimista tehtävistä, sekä ohjelmointi- tai matematiikan tehtävistä. Sovellus on alun perin suunniteltu käytettäväksi tavallisella PC-tietokoneella, esimerkiksi luokkahuoneessa. Nykyajan mobiilikeskeisessä maailmassa tämä ei kuitenkaan riitä, vaan käyttäjät haluavat ehdottomasti käyttää sovellusta myös mobiililaitteillaan.

Sivustoa on pyritty saamaan mobiilialustoille kelpaavaksi niinsanotun responsiivisen suunnittelun avulla. Tämä ei kuitenkaan tuota optimaalista käyttäjäkokemusta, muun muassa mobiiliprosessorien luontaisten ongelmien JavaScript-koodin kääntämisessä.

Tähän ongelmaan ratkaisuksi Viope Oy tilasi minulta mobiiliversiot oppimisjärjestelmästä. Mobiiliversioiden tulisi sisältää mahdollisuuden rekisteröitymiseen, sisäänkirjautumiseen, profiilin muokkaukseen, kurssin valintaan, teoriakappaleiden lukemiseen, monivalintatehtäviin, matematiikan tehtäviin käsinkirjoituksen tunnistuksella, keskustelualueiden lukemiseen sekä viesteihin vastaamiseen, tutor-kysymysten lukemiseen sekä omien kysymysten esittämiseen, sekä oman opiskelun seurannan mahdollisuuden erilaisten kaavioiden avulla. Mobiiliversiot toteutetaan aluksi Android- ja Windows 8.1-alustoille.

1.1 Viope Oy

Viope Oy on suomalainen teknologia-alan yritys, joka on perustettu vuonna 2001. Viope Oy:n kehittämä ja myymä tuote on digitaalinen, internet-selaimessa toimiva oppimisjärjestelmä jonka avulla voidaan luoda ja järjestää matematiikan sekä ohjelmoinnin kursseja.

1.2 Tavoitteet

Projektin tavoitteena on saattaa molemmat versiot mobiilisovelluksesta beta-julkaisuvalmiiksi yllämainitut ominaisuudet sisältäen, ja julkaista ne loppukäyttäjille.

Tämän jälkeen sovelluksiin tehdään vielä parannuksia sekä korjauksia käyttäjäpalautteen perusteella. Työn aloittamisen hetkellä Android-versio on jo lähes beta-julkaisuvalmis, joten käytännössä projekti pitää sisällään Android-version viimeistelyn, sekä Windows 8.1-version kehitystyön.

1.3 Opinnäytetyön rajaus

Viope-oppimisjärjestelmä sisältää kattavat työkalut opettajille, muunmuassa kurssien luomiseen sekä opiskelijoiden etenemisen seurantaan. Näitä ominaisuuksia ei kuitenkaan toteuteta tässä vaiheessa mobiiliversioihin, vaan mobiiliversiot jätetään puhtaasti opiskelijoiden työkaluksi.

1.4 Käsitteet

Beta-versio	Versio, joka on julkaistu käyttäjille, mutta ei vielä täysin valmis. Saattaa sisältää ohjelmistovirheitä tai ei-lopullisia käyttöliittymäelementtejä. Beta-versio ei ole kattavasti testattu.
Käyttäjä	Henkilö, joka on rekisteröitynyt Viope-oppimisjärjestelmän käyttäjäksi joko oppilaitoksensa kautta, internet-sivustolla http://www.viope.com tai suoraan applikaation kautta.
UX	User Experience (suom. Käyttäjäkokemus). UX-termillä kuvataan käyttäjän navigointia sovelluksessa, tiedon saatavuutta sekä käyttöliittymän ulkoasua.
Google Play	Googlen ylläpitämä sovelluskauppa, jonka avulla kehittäjät voivat jaella Android-sovelluksiaan.
Windows Store	Microsoftin ylläpitämä sovelluskauppa, jonka avulla kehittäjät voivat jaella Windows 8- sekä Windows 8.1-sovelluksiaan.

.NET	Microsoftin kehittämä ohjelmointikirjasto. Käytetään yleisimmin Microsoftin alustoilla toimivien ohjelmistojen kehitykseen.
HTTP	Hypertext Transfer Protocol. HTTP-protokollaa käytetään tiedonsiirtoon sovellusten ja palvelimien välillä.
DLL	Dynamic Link Library on Microsoftin kehittämä koodikirjastoformaatti, jonka avulla voidaan tuoda sovellukseen ulkopuolisten ohjelmistojen toiminnallisuuksia.

2 Tietoperusta

Tässä kappaleessa esitellään Android- ja Windows 8.1-käyttöjärjestelmät, sekä kehitystyön kannalta olennaiset teknologiat.

2.1 Android

Android on Googlen kehittämä käyttöjärjestelmä. Android hallitsi syyskuussa 2014 noin 85% älypuhelinien kokonaismarkkinoista, joten potentiaalista käyttäjäkuntaa sovellukselle on runsaasti. (Ong 2014) Android on avoimen lähdekoodin käyttöjärjestelmä, joten käyttöjärjestelmään voi halutessaan tehdä muutoksia täysin vapaasti.

Android-sovellukset ohjelmoidaan Java-ohjelmointikieltä käyttäen. Ulkoasut toteutetaan XML-merkintäkielellä, jota Android tulkistaa oman skeemansa avulla. Android-ohjelmistoja voidaan kehittää joko Googlen oman Android Studio-ohjelmiston avulla, tai Eclipse-liitännäisen avulla. Android-kehitys ei ole käyttöjärjestelmäriippuvaista, sillä Eclipse toimii kaikilla yleisimmillä käyttöjärjestelmillä (Windows, Mac, Linux).

Android-ohjelmistojen julkaisua ei ole rajoitettu millään tavalla. Ohjelmistopakettia voi halutessaan jaella omalta palvelimeltaan. Google kuitenkin ylläpitää omaa sovelluskauppaansa nimeltä Google Play, johon sovellus kannattaa julkaista näkyvyyden ja helpon saatavuuden vuoksi. Google ei edellytä minkäänlaista ennakkotarkastusprosessia Google Playhin julkaistaessa, mutta sovellus saatetaan poistaa kaupasta jälkikäteen, mikäli se rikkoo kaupan sääntöjä. Google Playhin julkaiseminen edellyttää kehittäjäksi rekisteröitymistä, sekä 25 dollarin kertamaksun. (Google 2014)

2.2 Windows 8.1

Windows 8.1 on Microsoftin kehittämä käyttöjärjestelmä, joka toimii niin mobiililaitteissa kuin tavallisissa PC-tietokoneissakin. Käyttöjärjestelmästä on kaksi eri versiota: tavallinen x86-arkkitehtuurin PC-käyttöjärjestelmä, sekä karsitumpi RT joka tukee ARM-prosessoriarkkitehtuuria. RT-käyttöjärjestelmää käytetään enimmäkseen halvemmissä tablet-laitteissa ARM-arkkitehtuurin edullisuuden vuoksi. Kehittämäni sovellus toimii käyttöjärjestelmän molemmilla versioilla.

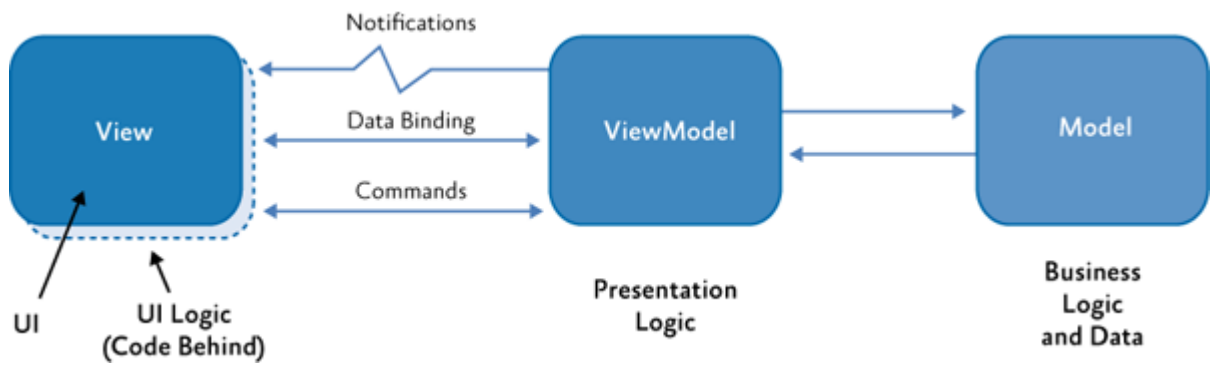
Windows 8.1-sovelluksia voidaan ohjelmoida useita eri kieliä käyttäen. Näistä ylivoimaisesti suosituin on C#. Muut vaihtoehdot ovat Visual Basic, C++ sekä JavaScript. Sovelluksen käyttöliittymä toteutetaan Microsoftin kehittämän XAML-merkintäkielen avulla.

Windows 8.1-sovellukset kehitetään Microsoftin kehittämän Visual Studio-kehitysohjelmiston avulla. Visual Studiosta on olemassa ilmaisversio Visual Studio Express, jonka avulla voidaan kehittää ainoastaan mobiiliohjelmistoja.

Windows 8.1-sovelluksia voidaan jaella ainoastaan Microsoftin ylläpitämässä Windows Storessa. Sovelluksen tulee läpäistä Microsoftin sertifiointiprosessi, jotta se hyväksytään myytäväksi Windows Storeen. Sertifiointiprosessissa sovellus tarkistetaan yleisimpien ohjelmointivirheiden sekä haittaohjelmien varalta. (Microsoft 2014)

2.3 MVVM

MVVM on Microsoftin kehittämä sovellusarkkitehtuurimalli, joka perustuu MVC- ja MVP sovellusarkkitehtuurimalleihin. MVVM on lyhenne termistä Model View ViewModel. MVVM:n päätarkoituksena on käyttöliittymäkoodin sekä bisneslogiikkakoodin vahva erottaminen toisistaan. MVVM:ssä käyttöliittymäkoodi kommunikoi back end-koodin kanssa data binding-teknologian avulla, jolloin back end-koodissa olevan muuttujan arvon vaihtuessa päivittyy arvo samalla hetkellä myös käyttöliittymään. Sama toimii myös toisin päin. (Osmani 2012)



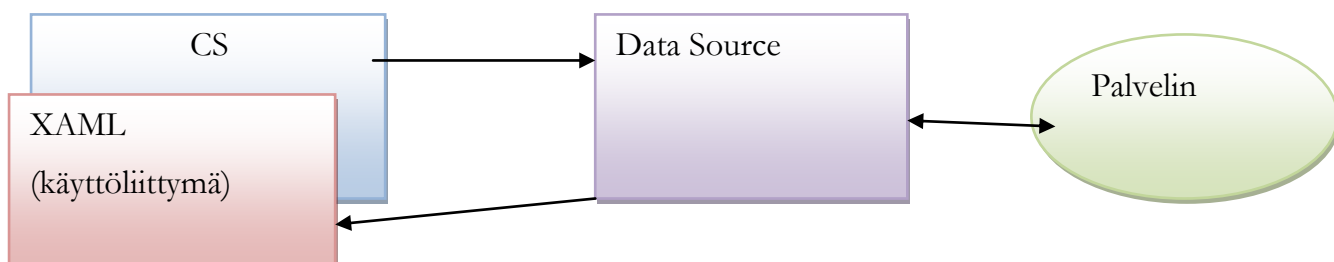
Kuvio 1 MVVM arkkitehtuurin kuvaus (Microsoft)

3 Viope-oppimisjärjestelmän sovittaminen mobiiliympäristöön

Tämän opinnäytetyön lopputuotena syntyy kaksi eri mobiiliapplikaatiota Viope-oppimisjärjestelmästä. Molemmat versiot ovat rakenteeltaan melko samanlaisia, joten keskityn tämän työn tarkoituksissa suurimmaksi osaksi Windows 8.1 –versioon.

3.1 Sovelluksen rakenne

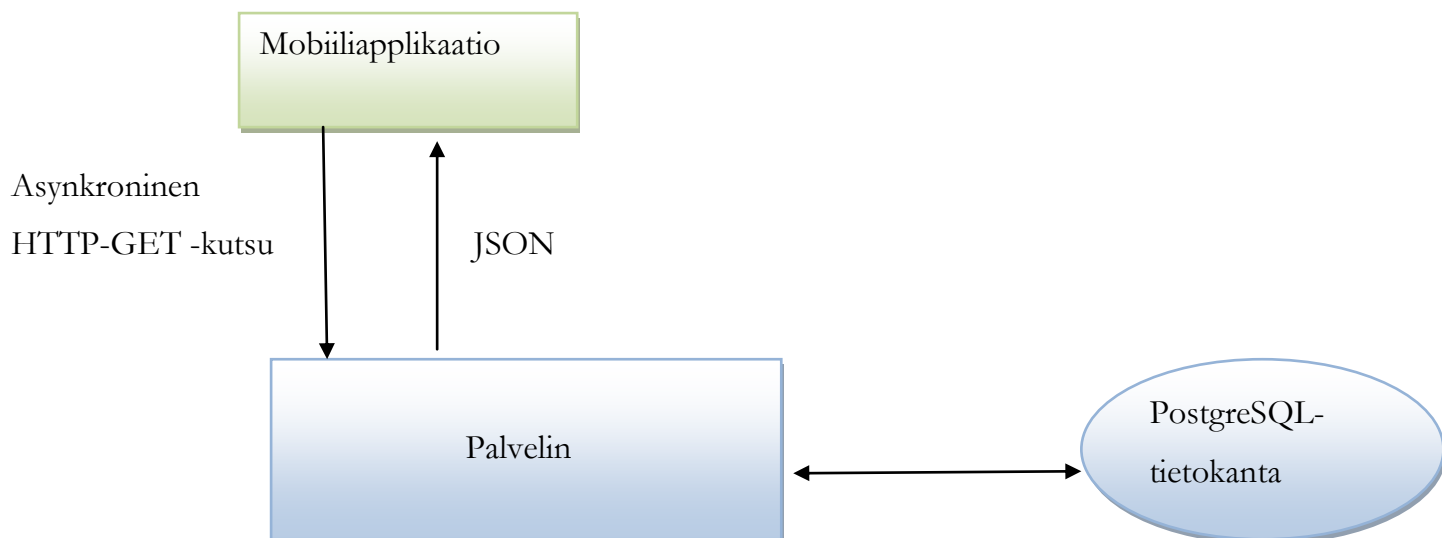
Windows 8.1-versio mobiiliapplikaatiosta on toteutettu kaikkia Microsoftin virallisia suosituksia noudattaen. Applikaatio on ohjelmoitu käyttäen C#- ja XAML-kieliä. Applikaatio noudattaa MVVM-rakennetta, joten käyttöliittymä ja loogiset operaatiot ovat vahvasti eritelty toisistaan. Applikaation jokainen sivu koostuu XAML-tiedostosta, jossa on määritelty sivun ulkoasu, CS-tiedostosta joka sisältää sivun vaatimat loogiset operaatiot ja sivujen väliset navigointimetodit, sekä toisesta CS-tiedostosta, jossa suoritetaan kaikki sivun vaatimat palvelinkutsut ja tietokokonaisuuksien rakentamisen, jotka voidaan data binding-tekniikan avulla esittää käyttäjälle XAML-tiedoston välityksellä.



Kuvio 2 Sovelluksen rakenne

3.2 Kommunikointi palvelimen kanssa

Mobiiliapplikaatiot saavat käyttämänsä tiedot dynaamisesti Viopen palvelimelta. Tiedot haetaan uudestaan jokaisella latauskerralla. Applikaatioon ei siis tallenneta käyttäjistä mitään tietoja. Applikaatiot kommunikoivat palvelimen kanssa JSON-rajapintaa käyttäen. Palvelin on asennettu Debian Linux-ympäristön päälle, ja se käyttää Perl- ja PostgreSQL-teknologioita sisällön tuottamiseen sekä tallentamiseen.



Kuvio 3 Tyypillinen esimerkki kommunikoinnista palvelimen kanssa

Kaikki HTTP-kutsut tehdään asynkronisesti, jolloin applikaation käyttöliittymäsi pysyy yhä aktiivisena. Tämä mahdollistaa esimerkiksi latausikonin näyttämisen ruudulla tietojen lataamisen ajaksi. Kaikki kerralla haettavat tietokokonaisuudet ovat kooltaan muutamia kilobittejä, joten latausajat pysyvät melko lyhyinä jopa mobiiliyhteyksilläkin.

```

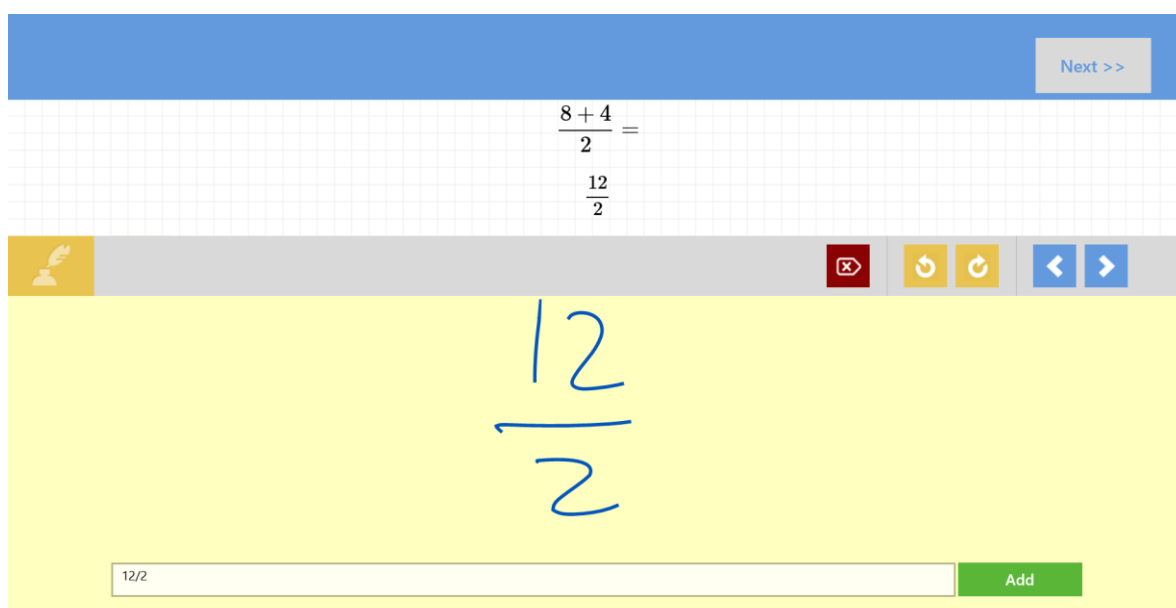
public async Task<ObservableCollection<StoreCourse>> GetData(){
    Windows.Storage.ApplicationDataContainer settings = Windows.Storage.ApplicationData.Current.RoamingSettings;
    string fetchUrl = (string)settings.Values["FetchUrl"];
    string cookie = (string)settings.Values["Cookie"];

    ObservableCollection<StoreCourse> courses = new ObservableCollection<StoreCourse>();
    Uri url = new Uri( );
    JsonArray root;
    using (var handler = new HttpClientHandler() { UseCookies = false })
    using (var client = new HttpClient(handler) { BaseAddress = url })
    {
        //cookieContainer.Add(url, new Cookie("Cookie", cookie));
        client.DefaultRequestHeaders.Add("Cookie", cookie);
        client.DefaultRequestHeaders.Add("X-Requested-With", "XMLHttpRequest");
        client.DefaultRequestHeaders.Add("user-agent", "Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)");
        HttpResponseMessage response = new HttpResponseMessage();
        try
        {
            response = await client.GetAsync(url);
        }
        catch (Exception)
        {
            return null;
        }
        var jsonString = await response.Content.ReadAsStringAsync();
        root = JsonValue.Parse(jsonString).GetArray();
    }
}
  
```

Kuvio 4 Esimerkki tyypillisestä asynkronisesta HTTP-kutsusta

3.3 Käsinkirjoituksen tunnistus

Yksi applikaation tärkeimmistä ominaisuuksista on matematiikan tehtävissä käytettävä käsinkirjoituksen tunnistusjärjestelmä. Käyttäjä voi kirjoittaa matemaattisia yhtälöitä piirustusalueelle joko sormellaan, tai jonkinlaista apuvälinettä, kuten esimerkiksi Microsoft Surfacen mukana toimitettavaa kynää käyttäen. Käyttäjän piirtämät viivat muutetaan X- ja Y-koordinaateiksi jotka lähetetään dll-kirjaston kautta toimivalle käsinkirjoitustunnistusliitännäiselle. Käsinkirjoitustunnistusliitännäinen analysoi koordinaatteja ja palauttaa applikaatiolle parhaan arvauksensa viivojen merkityksestä. Käsinkirjoituksen tunnistus toimii erittäin hyvin, ja useimmiten arvaa myös epäselvemmälläkin käsialalla kirjoitetut yhtälöt. Käsinkirjoituksen tunnistus parantaa siis käyttäjäkokemusta huomattavasti, sillä käyttäjät voivat kirjoittaa yhtälöitä totuttuun tapaan omalla käsialallaan saaden silti automaattisen tarkastuksen tuomat hyödyt.

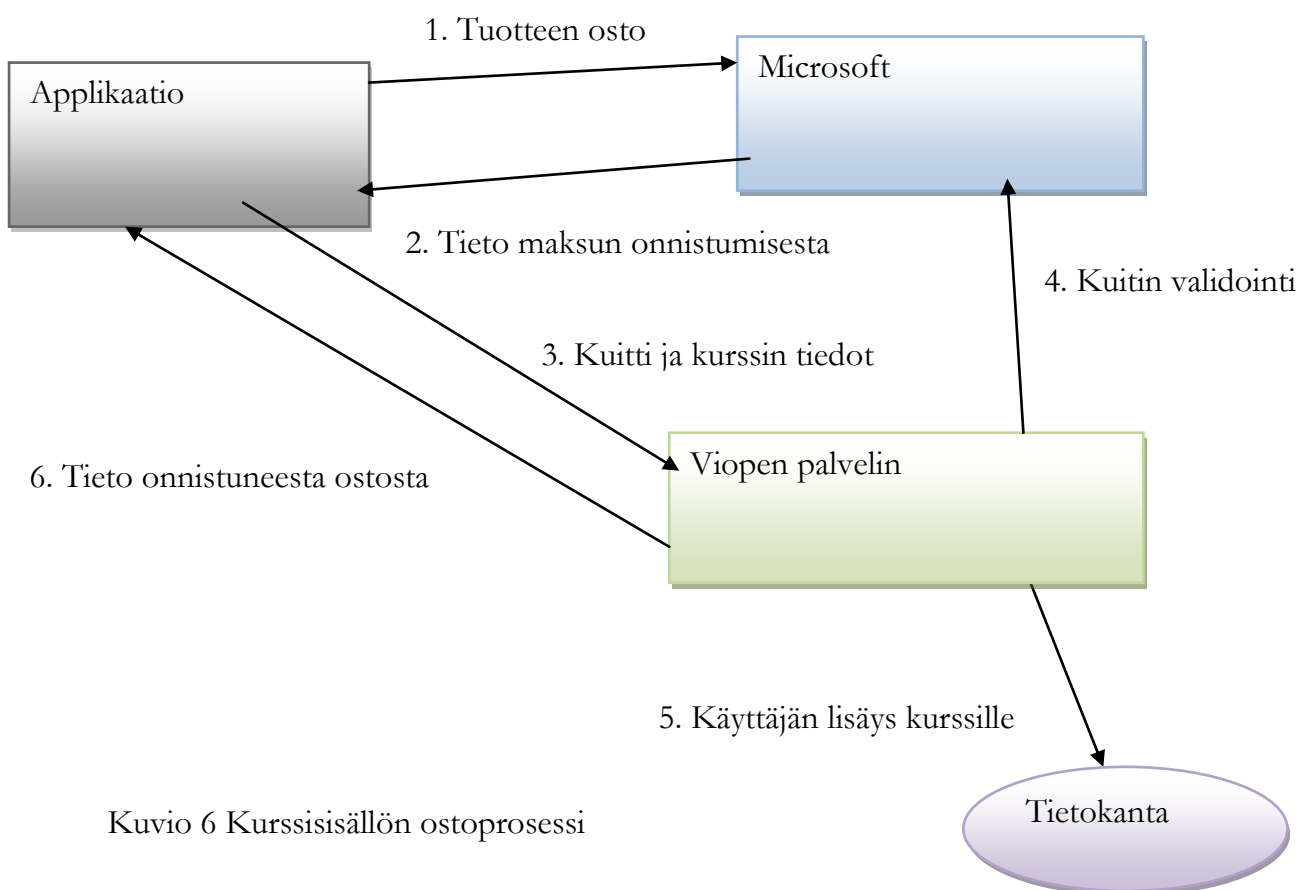


Kuvio 5 Käsinkirjoituksen tunnistus

3.4 Kauppa

Käyttäjien on mahdollista ostaa Windows 8.1-applikaation kautta uusia kursseja. Tätä varten applikaatioon on kehitetty oma kauppapaikka, joka listaa saatavilla olevat kurssit näyttäen niiden sisältämät aihepiirit sekä hinnan. Maksunvälittäjänä kauppa käyttää Microsoftin maksunvälityspalvelua. Tämä kuitenkin tuo osaltaan ongelmia, sillä

tuotteiden aktivoinnin tulee tapahtua Viopen palvelimen kautta. Suunnittelun kautta päädyin kaupan toteutuksessa seuraavaan prosessiin: Käyttäjä ostaa tuotteen, jolloin välitämme tiedon ostettavasta kurssista ja sen hinnasta Microsoftille. Käyttäjä kirjautuu Microsoft-tililleen ja maksaa kurssin hinnan, jolloin Microsoftin palvelin ilmoittaa applikaatiolle maksun tulleen perille. Tämän jälkeen applikaatio lähettää Viopen palvelimelle käyttäjän tunnistenumeron, ostetun kurssin tunnistenumeron, sekä XML-formaatissa olevan kuitin ostoksesta, joka sisältää uniikin kryptatun avaimen. Kuitin validiteetti varmennetaan Viopen palvelimella, ja mikäli se todetaan aidoksi, lisätään käyttäjä kurssille tunnistenumeroiden perusteella. Tästä tapahtumasta lähetetään JSON-formaatissa oleva varmenne takaisin applikaatiolle, jonka jälkeen applikaatio raportoi Microsoftin palvelimelle, että käyttäjälle on toimitettu hänen ostamansa tuote. Onnistuneen ostotapahtuman jälkeen käyttäjä ohjataan takaisin applikaation etusivulle, josta hän pääsee välittömästi käsiksi ostamaansa kurssisisältöön.



Kuvio 6 Kurssisisällön ostoprosessi

3.5 Applikaation UX-parannusprosessi

Projektin loppupuolella havaitsimme Windows 8.1-applikaatiossa olevan parantamisen varaa käyttäjäkokemuksen sekä käyttöliittymän asettelun saralla. Päätimme konsultoida ulkopuolista yritystä avuksemme ongelman ratkaisemiseksi. Valitsimme yhteistyökumppaniksemme suomalaisen Luxus Oy:n.

Luxus Oy toimitti parannusehdotuksia suurimpiin kompastuskiviin hyvin nopealla aikataululla, joiden pohjalta teimme tarvittavat muutokset applikaatioon. Ehdotettuja muutoksia olivat kirjautumis- ja rekisteröitymissivun ulkoasun uudelleensuunnittelu, marginaalien ja fonttien optimointi, kurssivalintasivun turhien elementtien poistaminen, ylävalikon uudelleensuunnittelu, toimintopainikkeiden lisääminen alavalikkoon sekä tehtävälistaussivun uudelleensuunnittelu.

Kun muutokset olivat toteutettu, lähetimme uuden version vielä kertaalleen Luxuksen tarkastettavaksi. Tällä kertaa applikaatiosta ei löytynyt kriittisiä käytettävyyso ongelmia.

Koimme UX-parannusprosessin tuovan huomattavan kohennuksen applikaation käyttäjäkokemukseen ja selkeyttävän informaation saatavuutta, joka on todella tärkeää applikaatiossa, jossa käsitellään hyvinkin suuria tietomääriä.

4 Windows 8.1-applikaation julkaisuprosessi

Käytettävyysparannuskierrosten jälkeen päätimme edetä applikaation julkaisuun. Julkaisua varten tuli hankkia yrityksille tarkoitettu 75 euron hintainen Windows-kehittäjälisenssi.

Ennen sertifiointia aloittamista applikaatiosta tuli toimittaa kuvankaappauksia, applikaation kuvaus sekä tärkeimmät ominaisuudet kaikilla tuetuilla kielillä (tässä tapauksessa englanti ja suomi), promootiokuvia viidellä eri resoluutiolla, sekä tietysti itse applikaation asennuspaketti. Tämän lisäksi applikaation sisäisistä ostoksista tuli toimittaa kuvaus sekä hintatiedot.

Applikaation asennuspakettia luodessa Visual Studion avustuksella oli mahdollista suorittaa koneellinen esi-sertifointi asennuspaketille, joka tarkasti yleisimmät mahdolliset ongelmat applikaation rakenteessa sekä ulkoasussa. Tämä prosessi on suositeltavaa käydä läpi, sillä mikäli automaattinen tarkistus ei läpäise testejä, ei se myöskään todennäköisesti läpäise varsinaista sertifiointia.

Applikaatio lähetettiin Microsoftin sertifiointiin maanantaina 27.10.2014.

Microsoftin julkaisuprosessin ongelmien vuoksi applikaatio saatiin julkaistua Windows Store-kauppapaikalle vasta 17.11.2014.

5 Web-ohjelmiston sovittaminen mobiiliympäristöön – Android vai Windows 8.1?

Tämän kappaleen tarkoituksena on verrata Android- ja Windows 8.1-käyttöjärjestelmiä siltä näkökannalta, kummalle alustalle web-sovelluksen sovittaminen on kehittäjän kannalta miellyttävämpää. Vaikka molemmat teknologiat ovat suunniteltu toimimaan mobiilialustoilla, on niissä silti huomattavia eroja. Tämän vuoksi on vaikea antaa yksiselitteistä vastausta alustan paremmuudesta, joten vertailu perustuu vain omaan mielipiteeseen.

5.1 Android

Androidin massiivinen käyttäjäkunta tekee siitä todella houkuttavan alustan kehittäjille, sillä potentiaalista käyttäjäkuntaa on vaikea jättää huomioitta. Alustan avoimuus helpottaa ja nopeuttaa sovelluksen julkaisuprosessia, sillä minkään näköistä ennakkotarkistusta ei tarvitse suorittaa ennen kauppapaikalle pääsyä. Tämä on osaltaan myös huono asia, sillä sovellusten määrän vuoksi kaupassa on todella vaikea saada näkyvyyttä.

Tekniseltä näkökannalta Android-sovelluskehitystä vaikeuttaa alustan Java-riippuvuus. Java-kielen vanhanaikaisuus ilmenee muun muassa asynkronisten HTTP-kutsujen todella kömpelöstä toteutuksesta, joka on valitettavasti todella suuri osa applikaation rakennetta web-ohjelmistoa sovittaessa. Android yrittää tarjota tähän ratkaisua omalla AsyncTask-luokallaan, mutta se ei silti yllä lähellekkään .NET-kielien helppokäyttöisyyttä tällä saralla.

Android-applikaatiot toimivat vain Android-alustalla, eivätkä ne ole helposti siirrettävissä muille alustoille.

5.2 Windows 8.1

Windows 8.1-sovellukset toimivat Microsoftin .NET-alustan päällä, jonka ansiosta niitä voidaan kehittää mielensä mukaan käyttäen joko C#, C++, Visual Basic tai JavaScript-

ohjelmointikieliä. .NET alustaan on myös tiukasti integroitu asynkronisuus yksinkertaisilla await-komennoilla, jonka ansiosta se sopii optimaalisesti web-ohjelmiston sovittamiseen. Windows 8.1-kehitystyökalut tarjoavat myös huomattavasti paremmat työkalut applikaation ulkoasun ja käytettävyyden hiomiseen, esimerkiksi edistyneen animaatiokirjastonsa ansiosta. Microsoftin XAML-ulkoasunkuvauskieli on äärimmäisen joustava ja mahdollistaa kokonaan omien ulkoasukomponenttien luomisen suhteellisen vähäisellä vaivalla.

Windows 8.1-sovellukset toimivat sellaisenaan pöytätietokoneilla ja tableteilla. Puhelinversiota varten tulee kehittää pienelle näytölle optimoitu käyttöliittymä, mutta taustalla toimivan koodilogiikan voi säilyttää ennallaan. Windows 8.1 tunnistaa automaattisesti käyttäjän syöttömetodin (hiiri, kosketus) ja optimoi joitain ulkoasuelementtejä sen mukaisesti, kuten esimerkiksi applikaation reunoilla esiintyvät vierityspalkit.

Windows 8.1 julkaisuprosessia hidastaa Microsoftin asettamat melko tiukat vaatimukset applikaation ulkoasua sekä toiminnallisuuksia koskien. Nämä ominaisuudet tarkistetaan Microsoftin toimesta sertifiointiprosessissa, jonka kaikki julkaistavat applikaatiot tulee läpäistä. Sertifiointiprosessin ansiosta Windows Storessa myytävien applikaatioiden voidaan olettaa täyttävän hyväksyttävät laatustandardit, toisin kuin Google Play-kaupassa. Applikaatioiden kokonaisuus on myös vähäisempi Windows Storessa, jonka ansiosta näkyvyyden saaminen omalle applikaatiolleen on todennäköisempää.

5.3 Johtopäätökset

Kaiken kaikkiaan mielestäni applikaation kehittäminen oli huomattavasti miellyttävämpää Windows 8.1-alustalle. Kehitystyökalut ovat edistyneemmät, jonka ansiosta pienet ja epäolennaiset asiat voitiin jättää helpommin työkalujen vastuulle, ja keskittyä applikaation olennaisiin osiin. Tämän, sekä XAML-kielen monimuotoisuuden ansiosta Windows 8.1-applikaatioon oli merkittävästi helpompi luoda käyttäjäkokemukseltaan miellyttävä ja selkeä käyttöliittymä.

Tästä huolimatta myös Android on hyvinkin varteenotettava alusta. Kehitys ei missään nimessä ole liian monimutkaista, ja Androidin ylivoimaisesti isoin käyttäjäkunta tekee siitä miltei pakollisen mikäli haluaa maksimoida applikaation potentiaalisen yleisön. Android-versiota voidaan pitää myös myös kustannustehokkaampana vaihtoehtona avoimen julkaisuprosessinsa ansiosta

6 Yhteenveto

Projektin tavoitteena oli molempien applikaatioiden beta-julkaisu. Tähän tavoitteeseen ei kuitenkaan rajallisen aikataulun vuoksi ehditty Windows 8.1-applikaation kehitysprosessin pitkittymisen vuoksi johtuen viime hetken päätöksestä parantaa applikaation käytettävyyttä.

6.1 Tulokset

Projektin lopputuloksena syntyi julkaisematon beta-versio Android-applikaatiosta sekä valmis ja julkaistu Windows 8.1-applikaatio.

Windows 8.1-applikaation julkaistu versio pitää sisällään seuraavat ominaisuudet:

- Rekisteröityminen
- Sisäänkirjautuminen
- Profiilin muokkaus
- Kurssien listaus
- Organisaation vaihtaminen
- Kurssikauppa
- Kurssin tehtävien listaus
- Teoriakappaleet
- Monivalintatehtävät
- Matematiikan tehtävät
- Käsinkirjoituksen tunnistus matematiikan tehtävissä
- Keskustelualue
- Tutor-kysymysten esittäminen sekä lukeminen
- Oman edistymisen seuranta

Toteutettu ominaisuuslista vastaa sisällöltään suunniteltua kokonaisuutta, lukuun ottamatta kurssikauppaa, joka päätettiin lisätä applikaatioon projektin loppuvaiheissa.

6.2 Jatkokehitys

Molempia applikaatioita tullaan jatkokehittämään projektin jälkeen. Android-versio vaatii muutaman ominaisuuden parantamista, sekä käytettävyyden hiomista ennen julkaisua. Windows 8.1-versioon on jo suunniteltu jatkokehityssuunnitelma, joka tulee vielä muovautumaan käyttäjäpalautteen perusteella.

Viope-oppimisjärjestelmä on edelleen hyvin aktiivisen kehityksen alla, joten tulevien muutosten tulee näkyä myös järjestelmän mobiiliversioissa. Tästä syystä molempia versioita tullaan aktiivisesti kehittämään pitkälläkin aikavälillä.

6.3 Oma oppiminen

Käytettävistä teknologioista minulla oli jo valmiiksi melko kattava osaaminen – en olisi muuten ryhtynyt näin isoon projektiin yksin. Opin silti melko paljon uutta liittyen spesifisesti Windows 8.1-tablettityökaluihin, sillä osaamiseni liittyi suurimmalta osin alustan puhelinversioon. Sain myös tärkeää tietoa liittyen Windows Storen julkaisuprosessiin.

Lähteet

Google 2014 Developer Registration Luettavissa:

<https://support.google.com/googleplay/android-developer/answer/113468?hl=en>.

Luettu 24.9.2014

Microsoft 2014 App certification requirements for the Windows Store Luettavissa:

<http://msdn.microsoft.com/en-us/library/windows/apps/hh694083.aspx>. Luettu

24.9.2014

Ong, J. 2014 Report: Android reached record 85% smartphone market share in Q2

2014, Xiaomi now fifth-largest vendor Luettavissa:

<http://thenextweb.com/google/2014/07/31/android-reached-record-85-smartphone-market-share-q2-2014-report/>. Luettu 24.9.2014

Osmani, A. 2012 Understandin MVVM – A Guide For JavaScript Developers Luetta-

vissa: <http://addyosmani.com/blog/understanding-mvvm-a-guide-for-javascript-developers/>. Luettu 24.9.2014