

Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Hyvinvointiteknologia

2010

Kimmo Koski

Rikkaiden puhelinsovellusten toteuttaminen Windows Phone 7 -käyttöjärjestelmälle



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan ko. | Hyvinvointiteknologia

Syyskuu 2010 | Sivumäärä 50

Ohjaaja: TKT Janne Roslöf

Kimmo Koski

Rikkaiden puhelinsovellusten toteuttaminen Windows Phone 7 -käyttöjärjestelmälle

Opinnäytetyö on osa esiselvitystä, joka tehdään tukemaan alkavan projektin kehitystyötä. Projektin tarkoituksena on luoda Internet-sovellus, jonne käyttäjät voivat lisätä ja jakaa viittomakielen termistöä videoina. Tässä työssä tehty puhelinsovellus tehdään tehostamaan Internet-sovelluksen käyttöä.

Puhelinsovellus toteutetaan käyttäen Microsoftin kehittämää Windows Phone 7 -puhelinkäyttöjärjestelmää. Käyttöjärjestelmä käyttää Microsoft Silverlight -teknologiaa, jonka avulla voidaan luoda puhelimeen rikkaita käyttöliittymiä. Rikkaalla käyttöliittymällä tarkoitetaan sovellusta, johon voidaan lisätä grafiikkaa, animaatioita ja mediaa. Tarkoituksena on luoda käyttöliittymä, joka kasvattaa sovelluksen käyttökokemusta ja käytettävyyttä.

Työssä kehitetään puhelinsovellus, jonka avulla voidaan käyttää samoja ominaisuuksia kuin Internet-sovelluksella. Työssä selvitetään, miten Windows Phone 7 -puhelinsovelluksen käyttöliittymä tulisi toteuttaa Microsoftin julkaisemien määritelmien mukaan.

Puhelinsovellus yhdistetään Internet-sovelluksen käyttämään tietokantaan Open Data Protocol -yhteydellä, jonka avulla puhelimesta voidaan käyttää samaa sisältöä kuin Internet-sovelluksessa. Työssä esitetään vaiheittain, miten OData-verkkopalvelu luodaan ja millä tavalla sitä voidaan käyttää puhelinsovelluksessa.

Puhelinsovelluksen esiselvitystyö on tehty tukemaan toteutuksen tuotantoversion suunnittelua ja toteuttamista.

ASIASANAT: Windows Phone 7, puhelinsovellus, Open Data Protocol

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Health Informatics

September 2010 | Number of pages 50

Instructor: D.Sc. Janne Roslöf

Kimmo Koski

Designing a Rich Phone Application for Windows Phone 7 Operating System

This thesis is a part of a feasibility study made to support the development of a starting product development project. The project aims to create a web application where users can share sign language terms in the form of video. The phone application created in this thesis was made to boost the usage of the Internet application.

The application was developed using Microsoft Windows Phone 7 phone operating system using Microsoft Silverlight technology, which allows developing rich user interfaces. A rich user interface means an application where you can use graphics, animations and media. The purpose is to create a user interface that increases the user experience and application usability.

The goal of this thesis was to create a phone application allowing users to have the same features in their phone as in the Internet application. The thesis examined how the user interface of Windows Phone 7 application should be developed according to Microsoft's specifications.

The phone application is connected to the same database as the Internet application is using. Open Data Protocol allows the phone to use the same content as the Internet application. The thesis documents step by step how to create an OData web service and how it can be used by the phone application.

The feasibility study of the phone application was made to support the design of the production version of the implementation.

KEYWORDS: Windows Phone 7, Phone application, Open Data Protocol

SISÄLTÖ

SYMBOLIT JA LYHENTEET

1 JOHDANTO	1
2 TOIMEKSIANTO	2
2.1 Ongelma	3
2.2 Ratkaisu	3
2.3 Nykytilan kartoitus	5
2.4 Projektikilpailut	8
3 PUHELINSOVELLUKSET	10
3.1 Haasteet	10
3.2 Microsoft - Windows Phone 7	11
3.3 Vaihtoehtoiset puhelinalustat	12
4 WINDOWS PHONE 7	14
4.1 Yleiskatsaus	14
4.2 Arkkitehtuuri	16
4.3 Käyttöliittymä	21
4.3.1 Start-aloitussivu	22
4.3.2 Sovellusten käyttöliittymä	23
5 OPEN DATA PROTOCOL	27
5.1 Protokollan toiminta	27
5.2 URI:n merkintätavat	29
6 SIGNBOOK-PUHELINSOVELLUS	32
6.1 Käyttöliittymä	32
6.2 Toiminnot	35
6.2.1 Termien haku	35
6.2.2 Tietuesivu	36
6.2.3 Chat	38
6.2.4 Oma profiili -sivu	39
6.2.5 Suosikit	39
6.2.6 Offline-viittomat	40
6.3 OData	40
6.3.1 OData-verkkopalvelun luominen	40
6.3.2 OData:n käyttäminen	42

7	JATKOKEHITYS	48
8	YHTEENVETO	49
	LÄHTEET	50

SYMBOLIT JA LYHENTEET

.NET	On Microsoftin kehittämä komponenttikirjasto, jota käyttävät kaikki VisualStudio.NET-ympäristössä kehitetyt ohjelmistot.
API	(Application Programming Interface) on käyttöliittymä, jonka kautta eri ohjelmat voivat vaihtaa tietoja keskenään.
EDM	(Entity Data Model) on tietomalli, joka pitää sisällään tietokantakuvauksen halutusta tietokannasta.
JIT	(Just In Time) on kääntäjä, jolla nimensä mukaan suoritetaan tarvittava työ vasta, kun se on tarpeellista, joka kasvattaa suorituskykyä.
JSON	(JavaScript Object Notation) on tiedonsiirtomuoto, jota voidaan käyttää esimerkiksi Java-pohjaisissa sovelluksissa.
OData	(Open Data Protocol) on protokolla, jonka avulla voidaan jakaa tietoa järjestelmän ulkopuolisille laitteille.
URI	(Uniform Resource Identifier) on merkkijono, jolla voidaan kertoa tietyn tiedon kuten verkkosivuston osoite.
XAML	(eXtensible Application Markup Language) on kuvauskieli, jonka avulla kaikki Silverlight-sovellusten käyttöliittymät luodaan.
XML	(eXtensible Markup Language) on standardoitu merkintäkieli, jonka avulla voidaan kuvata suuriakin määriä tietoa.

1 JOHDANTO

Tässä opinnäytetyössä tehtiin esiselvitystä Diakonia-ammattikorkeakoulun ja Turun ammattikorkeakoulun yhteistyöhankkeeseen nimeltään Signbook. Hankkeen tarkoituksena on luoda Internet-sovellus, jonka avulla voidaan jakaa viittomakielen termistöä videoina muille sivuston käyttäjille. Työtä voidaan käyttää apuna kehitystyössä, kun varsinainen projekti alkaa.

Tässä työssä toteutetaan Windows Phone 7 -puhelinkäyttöjärjestelmälle rikkaan käyttöliittymän omaava sovellus. Puhelinsovelluksen ulkoasu luodaan Microsoftin käyttöliittymämäärittelyjen mukaisesti. Sovellukset luodaan käyttämällä Silverlight-teknologiaa, joka takaa rikkaan käyttökokemuksen näyttävillä vektorigrifiikoilla ja animaatioilla. Käyttöliittymän rakenteessa keskitytään kahteen tärkeimpään kontrolliin, joiden sisään sovelluksen toiminta rakennetaan.

Yleisenä ongelmana puhelinsovelluksissa on ollut informaation hakeminen pääjärjestelmien tietokannoista. Open Data Protocol tarjoaa tähän ongelmaan ratkaisun tarjoamalla syötteen Windows Azure -pilvipalvelimella sijaitsevasta Azure SQL -relaatiotietokannasta. Käyttämällä OData Client Library liitännäistä voidaan puhelinsovelluksesta käsitellä OData-syötettä ja suorittaa kyselyitä, lisäyksiä ja muita operaatioita.

Raportissa käydään läpi käytettyjen teknologioiden teoriaa, jonka jälkeen tutustutaan niiden perustoimintaan käytännössä.

2 TOIMEKSIANTO

Toimeksiantona on esiselvitystyö, joka kuuluu Diakonia-ammattikorkeakoulun ja Turun ammattikorkeakoulun yhteishankkeeseen. Hankkeen tarkoituksena on luoda Internet-sovellus, jolla viittomakielen käyttäjät ja siitä kiinnostuneet voivat jakaa viittomakielen termejä videoina. Sovellus tulee olemaan laaja, minkä vuoksi siihen tehdään tarvittavia esiselvityksiä mahdollisten ongelmakohtien kartoittamiseksi, selvittämiseksi ja ratkaisemiseksi.

Esiselvitys aloitettiin kahden opiskelijan tekemällä suunnittelutyöllä. Suunnittelussa käytiin läpi asiat, jotka ovat sovellukselle välttämättömiä, ja toiminnot, jotka olisivat hyvä lisä perustoiminnoille. Opiskelijaryhmä lähti viemään esiselvitystä eteenpäin prototyypitysasteelle, jonka avulla toimeksiantaja saa paljon paremman kokonaiskuvan projektista ja sen eri osista. Turun ammattikorkeakoulusta kerättyyn opiskelijaryhmään kuuluivat lisäksi Petteri Lehtonen, Tomi Härkönen sekä Johannes Maliranta.

Petteri Lehtonen on tehnyt samaan esiselvitysprojektiin liittyvän opinnäytetyön nimeltään Rikkaiden internet-sovellusten toteuttaminen Microsoft-teknologioilla. Lehtonen perehtyi opinnäytetyössään rikkaiden Internet-sovellusten suunnitteluun ja kehittämiseen Silverlightin avulla. Työssä syvennyttiin tutkimaan sitä miten Silverlight-sovellus voidaan yhdistää tietokantaan käyttäen RIA Services -tekniikkaa. [1]

Yhteistyöllä toimeksiantajan kanssa on ollut keskeinen rooli projektin etenemiselle, koska opiskelijaryhmällä ei ole omia kokemuksia tai tietoutta viittomakielistä eikä sen opiskelusta. Diakonia-ammattikorkeakoulun henkilökunta on tarjonnut paljon perustietoutta viittomakielistä opiskelijaryhmälle.

2.1 Ongelma

Viittomakielen opiskelu on aikuisille haastavaa, koska viitottua kieltä vastaanotetaan näköaistin avulla kuuloaistin sijaan. Näköaistin käyttäminen on usein haastavaa viittomakielen oppimisen alkuvaiheessa, koska opiskelijat eivät ole tottuneet käyttämään silmiä kielikanavana. Tilannetta vaikeuttaa myös se, että viittomakieli on kolmiulotteinen ja käsien sekä vartalon liikkeiden lisäksi merkittäviä ovat myös ei-manuaaliset seikat, kuten kasvojen ilmaat tai huulet. Tällä hetkellä opiskelijat käyttävät apuna mm. vanhoja VHS-kasetteja, piirtämistä ja viittoman kuvaamista kirjoittamalla. Paras tapa on kuitenkin nähdä viittoma kokonaisuudessaan joko suoraan viittovalta henkilöltä tai vaihtoehtoisesti media-tallenteelta.

Verkosta löytyy viittomakielen sanakirjoja, mutta ne ovat pääsääntöisesti suljettuja eikä niissä ole sosiaalisen median elementtejä eikä mahdollisuutta lisätä tai arvioida sanakirjan sisältöä. Erikoissanastoa, kuten tiettyyn harrastukseen tai ammattiin liittyvää, ei ole yleisesti saatavilla lainkaan.

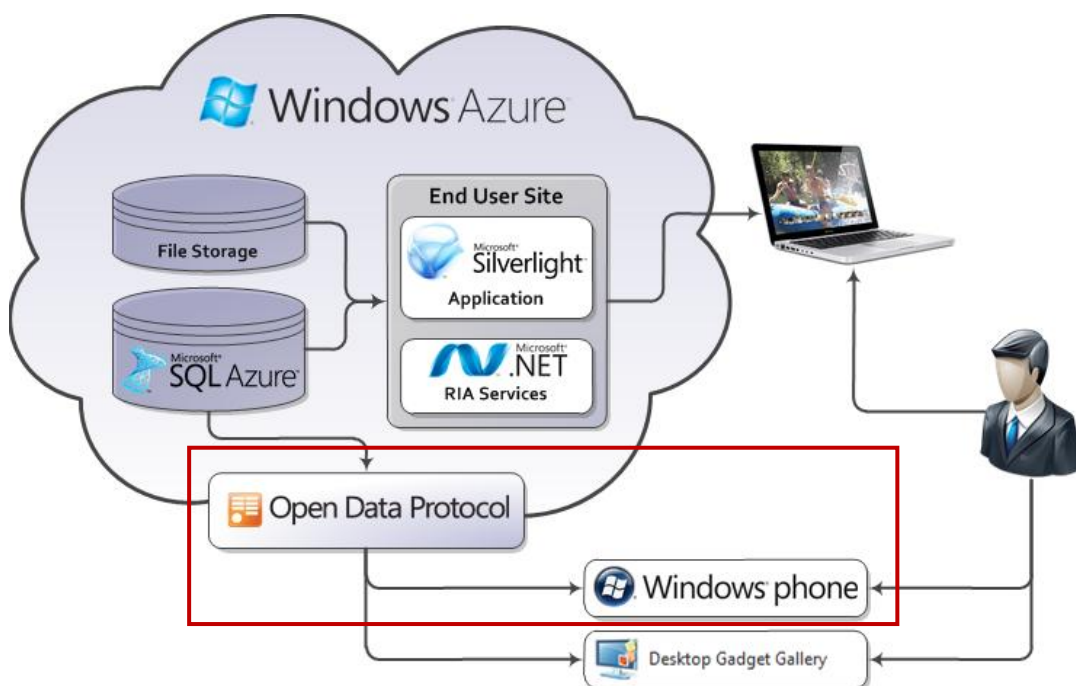
Diakonia-ammattikorkeakoulun viittomakielen opiskelijat ovat toivoneet ratkaisua näihin ongelmiin. Opiskelijat haluavat keskitetyn paikan, jonne he voisivat kerätä viittomakielen termistöä ja jakaa esimerkiksi työelämäharjoittelussa oppimia viittomia koko luokan kesken. Tämä tehostaisi heidän suomalaisen viittomakielen opiskeluaan sekä sanaston hallintaa.

2.2 Ratkaisu

Tarkoituksena on luoda avoin alusta viittomakielen termistölle, jonne käyttäjät voivat lisätä termistöä videoina. Sovelluksesta ei tehdä sanakirjaa, jossa yhtä sanaa vastaa yksi viittoma, vaan mahdollistaa käyttäjien lisätä synonyymeja ja sosiaalisia, alueellisia sekä ikäluokkien murreilmauksia. Synonyymien lisääminen tulee vaatimaan sisällön laadun arviointia ja kommentointia, mistä syystä sivustoon tullaan lisäämään sosiaalisia elementtejä, joiden avulla käyttäjät ovat vuorovaikutuksessa toistensa kanssa ja arvioivat sivuston sisältöä yhdessä.

Sosiaalisuuden kautta äidinkieleltään viittomakieliset, viittomakieltä opiskelevat sekä sen kanssa työskentelevät pystyvät helpommin kommunikoimaan keskenään viittomakielestä. Näin koko alan yhteinen kielikeskustelu esimerkiksi tietyn termin soveltuvuudesta tiettyyn käyttöympäristöön lisääntyy. Lisäksi sovellus voi olla avuksi jokaiselle, joka kohtaa esimerkiksi työympäristössään viittomakielisen ihmisen ja tarvitsisi muutaman avainviittoman, jonka avulla saa asian hoidettua.

Esiselvitystyötä tehtäessä sovellukselle annettiin nimeksi Signbook, joka ei välttämättä tule olemaan lopullisen sovelluksen nimi. Sovelluksen arkkitehtuuri tulee olemaan hyvin laaja, mutta silti toiminnaltaan yksinkertainen. Tässä työssä tullaan käymään läpi kuvassa 2.1 punaisen alueen sisäpuolelta löytyviä arkkitehtuurin osia, eli Open Data Protocol -yhteyttä ja Windows Phone 7 -sovelluksen kehittämistä ja sen yhdistämistä OData-syötteen tarjoamaan sisältöön.



Kuva 2.1: Signbook arkkitehtuurin tekninen kuvaus

Signbook-sovellusta tullaan ajamaan Windows Azure -pilvipalvelin ympäristössä. Azure tarjoaa nopean ja skaalautuvan ympäristön, jonka avulla voidaan esimerkiksi suorituskykyä nostaa, jos käyttäjämäärät kasvavat nopeasti.

Azuressa voidaan ajaa itse sovellusta ja sen taustalla käytettävää tietokantaa SQL Azuressa. Lisäksi pilvipalvelimeen pystytään tallentamaan kaikkien viittomien videot, joiden avulla videoiden lataaminen sovellukseen toimii mahdollisimman sulavasti.

Signbookin selainsovellus on tehty kokonaan Microsoft Silverlight 4 -teknologialla. Silverlight tarjoaa mahdollisuuden luoda rikkaita käyttöliittymiä, jotka ovat käyttökokemukseltaan paljon perinteisiä verkkosivuja edellä. Silverlight vaatii käyttäjän tietokoneelta ainoastaan selainliitännäisen, joka on jo kaikissa markkinoiden johtavissa Internet-selaimissa. Silverlight-sovellus käyttää tietojen hakemiseen WCF RIA Servicesia, jonka avulla saadaan luotua yhteys Silverlight-sovelluksen ja SQL Azure -tietokannan välille. RIA Services tarjoaa nopeita ja yksinkertaistettuja tapoja tiedon hakemiseen tietokannasta ja päivittämiseen tietokantaan.

Windows Azure tulee sisältämään myös Open Data Protocol (OData) -syötteen, joka muuntaa SQL Azuressa olevan tietokannan Web API:ksi, joka tarjoaa Signbookin sisällön syötteenä ulospäin. Syötettä voidaan sittemmin käyttää Azure-ympäristön ulkopuolella olevissa laitteissa tai sovelluksissa.

Azure-ympäristön ulkopuolelle on toteutettu Windows Phone 7 -sovellus, joka hyödyntää OData-syötettä hakiessaan samat tiedot puhelimeen, jotka löytyvät myös selainsovelluksesta. Lisäksi ulkopuolelle on kehitetty Windows Gadget, jonka avulla voidaan tarkastella uusimpia viittomatermejä suoraan työpöydältä.

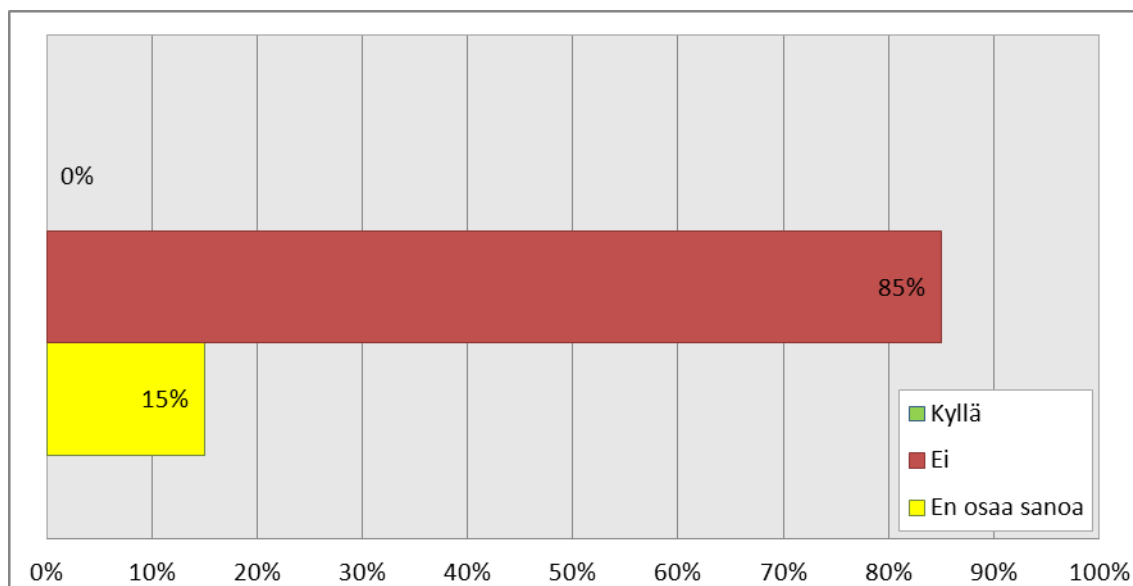
2.3 Nykytilan kartoitus

Nykytilan kartoittamisella haetaan tietoutta loppukäyttäjiltä tulevan sovelluksen käytöstä ja tämän hetkistä ongelmista. Kartoitusta varten vierailtiin Diakonia-ammattikorkeakoulussa, jossa pidettiin esitys Signbook-sovelluksesta viittomakielen opiskelijoille ja henkilökunnalle. Tavoitteena oli demonstroida läsnäolijoille sitä, mitä tulevalla sivustolla voidaan tehdä. Vierailun tarkoituksena oli kartoittaa viittomakielen opiskelun nykyisiä käytäntöjä ja kuulla opiskelijoiden

ja henkilökunnan mielipiteitä siitä, tulisiko Signbook tukemaan ja parantamaan opiskelun laatua.

Vierailua varten laadittiin viittomakielen opiskelijoille kysely, johon he vastasivat Signbook-sovelluksen esityksen ja yleisen keskustelun jälkeen. Kyselyyn vastasi 47 opiskelijaa, ja sen pohjalta saatiin suhteellisen vankka tietämys tämän hetkisestä tilanteesta. Viittomakielen opiskelumenetelmät ovat lähes samanlaiset ympäri maailmaa, joten kyselyn tuloksia voidaan myös käyttää arvioidessa olemassa olevien haasteiden laajuutta Suomen ulkopuolella.

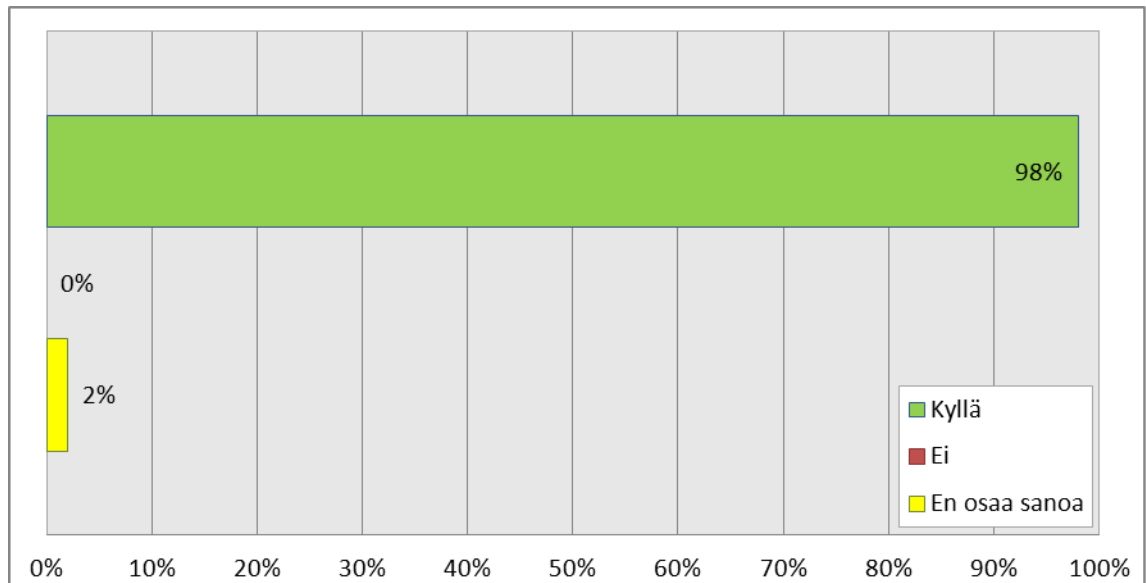
Ensimmäiseksi opiskelijoilta kysyttiin, ovatko he tyytyväisiä nykyisiin opetusmenetelmiin, joilla viittomakieltä opetetaan. Kuvassa 2.2 oleva pylväsdiagrammi näyttää kysymyksen tuloksen, ja siitä voidaan havaita, että jopa 85 % opiskelijoista on tyytymättömiä nykyisiin menetelmiin. Lisäksi huomioitavissa on se, ettei kukaan kyselyyn vastanneista vastannut myönteisesti esitettyyn kysymykseen. Tuloksesta voidaan päätellä, että opiskelijat haluaisivat parannusta nykyisiin opiskelumetodeihin.



Kuva 2.2: Oletko tyytyväinen nykyisiin viittomakielen opiskelumetodeihin?

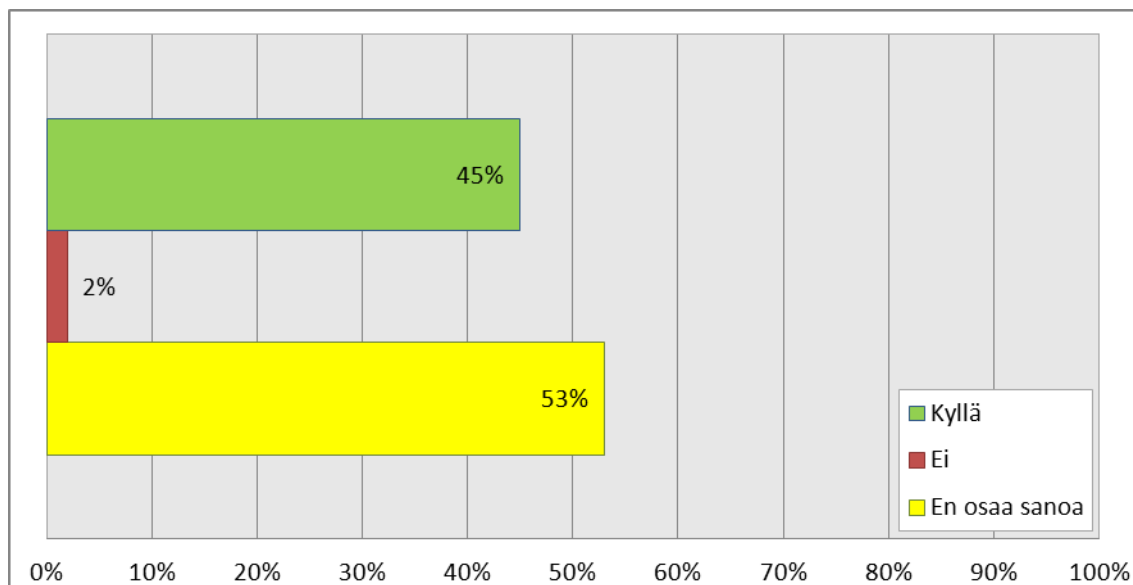
Opiskelijoilta kysyttiin, auttaisiko Signbook viittomakielen opiskelua ja käyttäisivätkö he sitä, jos se olisi käytettävissä. Kuvassa 2.3 oleva diagrammi näyttää, että lähes kaikki opiskelijat vastasivat myönteisesti. Oli helppo päätellä,

että oltiin oikealla polulla sovelluksen prototyypin kehittämisessä. Hyvänä lisänä oli hieno nähdä, ettei kukaan opiskelijoista vastannut kysymykseen kieltävästi.



Kuva 2.3: Auttaisiko Signbook viittomakielen opiskelua ja käyttäisitkö sitä, jos se olisi saatavilla?

Signbook-sovellukseen liittyen kysyttiin vielä, lisääisivätkö opiskelijat itse sisältöä Signbookkiin. Kuvassa 2.4 oleva pylväsdiagrammi osoittaa, että lähes puolet opiskelijoista olisi valmiita itse lisäämään sisältöä sivustolle, mikä vaikuttaa jo tässä vaiheessa todella hyvältä prosentilta, koska sivuston sisällön kasvuun tarvitaan paljon aktiivisia käyttäjiä. Kysymys kuitenkin osoitti, että oli puolet opiskelijoista oli epävarmoja sisällön lisäämisestä. Opiskelijoilla saattaa olla suuri kynnyks lisätä termistöä omalla nimellään sekä omilla kasvoilla. Opiskelijoilla on mahdollisesti pelko siitä, ettei viittoman suoritus tapa olekaan oikein ja opiskelijat pelkäävät sen tulevan heille taakaksi.



Kuva 2.4: Lisäisitkö sisältöä Signbookkiin?

Lopuksi opiskelijoilta kysyttiin kokonaisarviota siitä, kuinka helpolta Signbookin käyttö näytti arvoasteikolla 1–5. Vastauksista lasketuksi keskiarvoksi saatiin 4.3, joka tässä vaiheessa projektia vaikuttaa todella hyvältä. Parannettavaa tietysti aina on, mutta on hyvä huomata, että opiskelijat pitivät tämänhetkistä sovellusta selkeänä ja helppokäyttöisenä. Viittomakielen tulkkiopiskelijat eivät yleensä ole kovin hyviä tietoteknisiltä taidoiltaan, joten näin korkeaa prosenttia voidaan pitää hyvin merkittävänä indikaattorina sovelluksen helppokäyttöisyydestä.

2.4 Projektikilpailut

Tekninen esiselvitystyö alkoi kirjallisella esiselvityksellä minun ja Petteri Lehtosen toimesta. Esiselvityksellä kartoitettiin sivuston toiminnallisuuksia ja vaatimuksia. Esiselvityksen edetessä päätettiin lähteä viemään Signbook-sovellusta prototyypitysvaiheeseen, jolla osallistuimme myös kahteen projektikilpailuun.

Ensimmäinen opiskelijaprojektikilpailu oli ICT Showroom, joka on Turun ammattikorkeakoulun, Turun yliopiston ja Åbo Akademin välinen kilpailu. Kilpailu pidettiin ICT-talossa Turussa 11. maaliskuuta 2010. Kilpailussa oli mukana yli 30 projektityötä, joista tuomaristo valitsi Signbook-projektin kilpailun voittajaksi.

Signbook-sovelluksella osallistuttiin myös Microsoft Imagine Cup 2010 -kilpailuun, joka on maailman suurin IT-alan opiskelijaprojektikilpailu. Signbook-projekti pääsi jatkoon kansallisessa loppukilpailussa neljän parhaan joukkoon. Kilpailun toisen vaiheen esitykset pidettiin Suomen finaalissa 27. huhtikuuta 2010 Helsingissä Microsoftin tiloissa. Suomen Imagine Cup tuomaristo valitsi neljästä parhaasta ryhmästä Signbookin edustamaan Suomea maailmanlaajuiseen finaaliin, joka vuonna 2010 pidettiin Puolan pääkaupungissa Varsovassa (Kuva 2.5).



Kuva 2.5: Microsoft Imagine Cup 2010 -logo

Imagine Cup 2010 -finaali pidettiin 3.–9. heinäkuuta 2010. Software Design -sarjan loppukilpailuun tuli joukkueita 68 eri maasta ympäri maailmaa, joista jokainen oli ensin voittanut oman maansa finaalin. Kilpailu eteni kolmessa vaiheessa:

- ensimmäinen karsinta 12 ryhmään
- toinen karsinta 6 ryhmään
- viimeinen esitys, joista valitaan 3 parasta.

Kilpailun karsinta oli tiukka, koska ryhmiä oli niin paljon ja koska kaikki projektityöt olivat hyvin korkealaatuisia. Karsintojen ja hyvin menneiden esitysten jälkeen pääsimme lopulta maailman kuuden parhaan projektiryhmän joukkoon, mutta emme yltäneet palkinnoille asti. Kansainvälinen projektikilpailumenestys takaa osaltaan sen, että Signbook-sovellusta pidetään vahvana työkaluna tämän ongelman ratkaisemiseen.

3 PUHELINSOVELLUKSET

Älypuhelimet ovat yleistyneet viimeisinä vuosina huomattavasti niiden hinnan laskun myötä. Hintojen laskun vuoksi myös puhelinsovellusten kehittäminen on tullut osaksi yritysten toimintaa. Puhelinsovelluksen avulla voidaan tuoda esimerkiksi yrityksen verkkosivujen tärkeimmät toiminnot puhelimeen, mikä voi kasvattaa tai tehostaa palvelujen käyttöä. Sosiaaliset mediat ovat hyvä esimerkki siitä, miten puhelimet tehostavat niiden käyttöä. Uusimmille puhelimille on kaikkiin tehty omia sovelluksia, joiden avulla pystytään käyttämään sosiaalisten medioiden kuten Facebookin ja Twitterin perusominaisuuksia. Nykyään älypuhelimien käyttöjärjestelmät pystyvät lähes samaan, mihin tietokoneiden käyttöjärjestelmät. Siitä syystä on mahdollista luoda tehokas puhelinsovellus myös Signbookille, joka tarvitsee videoiden toistoa ja yhteyttä samoihin tietoihin, joita verkkosivusto käyttää.

Signbook-projektin ensimmäinen puhelinsovellus päätettiin kehittää Windows Phone 7 -alustalle. Windows Phone 7 -käyttöjärjestelmään päädyttiin, koska se on helpoin ympäristö luoda puhelinsovellus sen jälkeen kun Signbookin verkkosivusto oli kehitetty käyttäen Silverlightia. Puhelinsovellus toimii täysin Silverlightilla kuten verkkosivustokin, joten puhelinsovelluksen käyttöliittymän luonti tapahtuu samalla tavalla. Windows Phonen taustakoodina toimii myös C# ja sillä pystytään käyttämään joitain samoja luokkakirjastoja, joita käytetään tehdessä Silverlight-sovellusta selaimen.

3.1 Haasteet

Suurimpana haasteena puhelinsovellusten luomisessa on käyttäjäkunnan tavoittaminen, koska puhelinkäyttöjärjestelmiä on useita ja ne kaikki toimivat erilaisilla alustoilla. Koska puhelinkäyttöjärjestelmät toimivat erilaisilla alustoilla, joudutaan jokaiseen käyttöjärjestelmään tekemään sovellus erikseen. Ennen puhelinsovelluksen yhdistäminen jonkun järjestelmän tietokantaan on ollut paljon vaikeampaa kuin nykyään. Nyt käytössä on erilaisia tapoja, jotka on tehty helpottamaan puhelinsovellusten ja suurempien järjestelmien välistä tiedonsiirtoa. Open Data Protocol (OData) on yksi mahdollisuus päästä käsiksi

Microsoft SQL ja SQL Azure -tietokantoihin. OData:n avulla myös kolmannen osapuolen kehittäjät voivat luoda omia sovelluksia eri puhelinalustoille käyttäen julkista OData-syötettä. Tämän avulla voidaan antaa palvelun käyttäjille mahdollisuus luoda sovelluksia, jotka voivat hyvinkin erota siitä mitä palvelun omistajan näkemys sovelluksesta on. Tämä käytäntö on tullut laajalti esille sosiaalisen median kuten Twitterin ja Facebookin kolmannen osapuolen kehittämissä puhelinsovelluksissa, joita on lukuisia määriä eri puhelinkäyttöjärjestelmissä.

3.2 Microsoft - Windows Phone 7

Windows Phone 7 on Microsoftin 2010-vuoden alussa julkaisema uusi puhelinkäyttöjärjestelmä. Microsoft alkoi kehittää seuraajaa Windows Mobile -käyttöjärjestelmälle jo vuonna 2004, mutta projekti lopetettiin kokonaan vuonna 2008 erinäisten viivästysten vuoksi. Microsoftin puhelintiimi alkoi tyhjältä pöydältä kehittää täysin erilaista käyttöjärjestelmää, mitä edeltäjä Windows Mobile oli. Microsoft hyödynsi Silverlight-teknologiaa myös puhelinkäyttöjärjestelmässä, jonka avulla käyttöjärjestelmässä pystytään esimerkiksi käyttämään rikkaista Internet-sovelluksista tuttuja animaatioita. Silverlight-teknologian ansiosta puhelimen käyttöliittymän käyttökokemus on erilainen kilpailijoihin nähden. Ensimmäisten Windows Phone -puhelinten myynti alkaa Euroopassa tämänhetkisten tietojen mukaan lokakuussa 2010. [2]

Suurimmat arvostelusivustot ovat nähneet uudessa Windows Phone -käyttöjärjestelmässä paljon potentiaalia. Microsoftilla on comScoren tekemän tutkimuksen mukaan parhaillaan 13 %:n markkinaosuus Yhdysvaltojen älypuhelinmarkkinoista, mutta sen uskotaan kasvavan uusien Windows Phone 7 -puhelinten myötä. [3]

3.3 Vaihtoehtoiset puhelinalustat

Työssä valittiin alustaksi uusi Windows Phone 7 -käyttöjärjestelmä. Älypuhelinmarkkinoilla on silti paljon muita suuria puhelinalustoja, joita on mahdollista hyödyntää samalla tavalla kuin Microsoftin kehittämää käyttöjärjestelmää. Open Data Protocol -verkkopalvelun avulla sama informaatio voidaan hakea myös muihin älypuhelmiin, kuten Googlen Android ja Applen iOS. Työtä varten selvitettiin yleistietoa markkinoiden vahvimmistä älypuhelinkäyttöjärjestelmistä.

Google – Android

Android on käyttöjärjestelmä, joka on tehty kannettaville laitteille, kuten puhelimiin ja paneelitietokoneisiin. Käyttöjärjestelmä on Googlen kehittämä ja se perustuu Linux-ytimeen. Android oli alkujaan pienemmän yrityksen kehittämä, mutta Google osti sen itselleen nähdessään potentiaalisen käyttöjärjestelmän tuleviin kannettaviin laitteisiin.

Ensimmäinen Android 1.5 -alusta julkaistiin helmikuussa 2009, joka on koodinimeltään "Cupcake". Ensimmäisen julkaisun jälkeen on tullut jo kolme uutta versiota, joista viimeisin versio Android 2.2 "Froyo". Froyo on julkaistu vasta muutamille laitteille, mutta se tulee yleistymään todella nopeasti vuoden 2010 loppuun mennessä. Android-alustan yksi vahvuuksista on sen avoimuus. Avoimuudella Google on saanut taakseen suuren sovelluskehittäjäjoukon. Android on ollut saatavilla vasta puolitoista vuotta, mutta kehittäjät ovat jo tehneet yli 70 000 ohjelmaa saataville Android Markettiin, josta kaikki Androidin käyttäjät voivat ladata niitä laitteisiinsa.

Android-käyttöjärjestelmä koostuu Java-sovelluksista, joita ajetaan Java-pohjaisella oliokielellä. Javan taustalla on suuri määrä erilaisiin tarkoituksiin tarkoitettuja kirjastoja, joita voidaan hyödyntää sovelluskehityksessä. Tätä kaikkea ajetaan Dalvik-virtuaalikoneella, joka hyödyntää Just In Time (JIT) -kääntäjää. Tietojen tallentaminen on ratkaistu käyttämällä SQLite-relaatiotietokantaa. [4]

Android puhelimien suosio on kasvanut räjähtävää vauhtia ja kasvaa edelleen. comScore-sivuston tilastojen mukaan Android on jo ottanut 13 % Yhdysvaltojen älypuhelin markkinoista. [3]

Apple – iOS

iOS on Applen kehittämä puhelinkäyttöjärjestelmä, joka kehitettiin alun perin iPhone puhelimelle. Myöhemmin Apple on julkaissut iPod Touch- ja iPad-laitteet, jotka käyttävät samaa käyttöjärjestelmää.

Ensimmäisen iOS-käyttöjärjestelmän Apple julkaisi iPhone puhelimen kanssa tammikuussa 2007. Käyttöjärjestelmää kutsuttiin ennen iPhone OS:ksi, mutta myöhemmin Apple julkaisi muita laitteita samalle alustalle, jonka jälkeen nimeksi tuli iOS. iOS-käyttöjärjestelmällä on tunnetusti eniten sovelluskehittäjiä ympäri maailman. iPhone puhelimeen on kehitetty yli 225 000 sovellusta, jotka on saatavilla Applen omasta sovelluskaupasta nimeltä App Store. iOS on Mac OS X:stä johdettu käyttöjärjestelmä ja käyttää samaa Darwin-alustaa, joten käyttöjärjestelmä on luonteeltaan Unixin tapainen. iOS-käyttöjärjestelmä koostuu neljästä osasta, jotka ovat ydinkerros, palvelukerros, Cocoa-kosketuskerros ja mediakerros. [5]

comScore-sivuston tilastojen mukaan Apple on edellä Googlen Android -käyttöjärjestelmää vajaan 25 %:n markkinaosuudellaan. Ainoastaan RIM on Applen edellä 42 %:n markkinaosuudella. RIM-käyttöjärjestelmää käytetään BlackBerry puhelimissa, jotka ovat suuressa suosiossa Yhdysvalloissa, mutta Euroopassa muut käyttöjärjestelmät hallitsevat markkinoita. [3]

4 WINDOWS PHONE 7

4.1 Yleiskatsaus

Windows Phone 7 -sovellusalusta antaa mahdollisuuden luoda rikkaita sovelluksia, joita voidaan käyttää puhelimella. Sovelluksia voidaan luoda jo aikaisemmin käytössä olleilla teknologioilla kuten Silverlightilla ja XNA Frameworkilla. Kehittäminen Windows Phone -alustaan on hyvin helppoa, jos on aikaisemmin tehnyt esimerkiksi rikkaita Internet-sovelluksia Silverlightin avulla, koska kehitystyökalut ovat samat. Tämä tehostaa ohjelmistokehittäjien työtä todella paljon, koska ei jouduta opettelemaan uusia työkaluja sovellusten tekemiseen.

Windows Phone 7 -sovellusalusta tarjoaa kaksi eri ohjelmistoalustaa, joilla voi kehittää sovelluksia.

- Silverlight, joka muodostuu Extensible Application Markup Language (XAML) -pohjaisesta kielestä, jolla luodaan sovelluksen käyttöliittymä. Toiminnallisuus luodaan tapahtuma-ajoisella taustakoodilla, jossa voidaan käyttää esimerkiksi C# tai VB.NET -kieltä.
- XNA Framework on tarkoitettu pelikehitykseen, jonka avulla voidaan luoda näyttäviä pelejä.

Windows Phone -sovellusalusta auttaa kehittäjiä luomaan sovelluksia monella tavalla. Alustaan on suunniteltu helposti käytettävät ohjelmointirajapinnat, joiden avulla pystyy esimerkiksi hakemaan GPS-koordinaatit kehitettyyn puhelinsovellukseen. Jokaiselle sovellukselle on oma eristetty hiekkalaatikko, jossa sovellusta ajetaan ja jonne sovellukset voivat tallettaa pysyviä ja väliaikaisia tietoja. Ajonaikaisia palveluita voidaan käyttää yhdistäessä sovellusta muihin palveluihin, kuten ilmoituspalveluun. Lisäksi kehittäjille annetaan mahdollisuus jakaa tai myydä omia sovelluksiaan Windows Phone Marketplace -sovelluskaupan kautta loppukäyttäjille.

Sovellusalustan suunnittelussa on ollut useita tavoitteita. Alustalle haluttiin mahdollistaa rikkaiden ja monipuolisten sovellusten luominen. Haluttiin tukea käyttäjien henkilökohtaisten tietojen tiedonsiirtoa useiden eri laitteiden välillä, sekä luoda vahva kehittäjäportaali ja sovelluskauppa, jotka takaavat sovelluksen koko elinkaaren hallinnan. Lisäksi tätä kaikkea tukemaan haluttiin toteuttaa tehokkaat ja tuottavat kehitystyökalut.

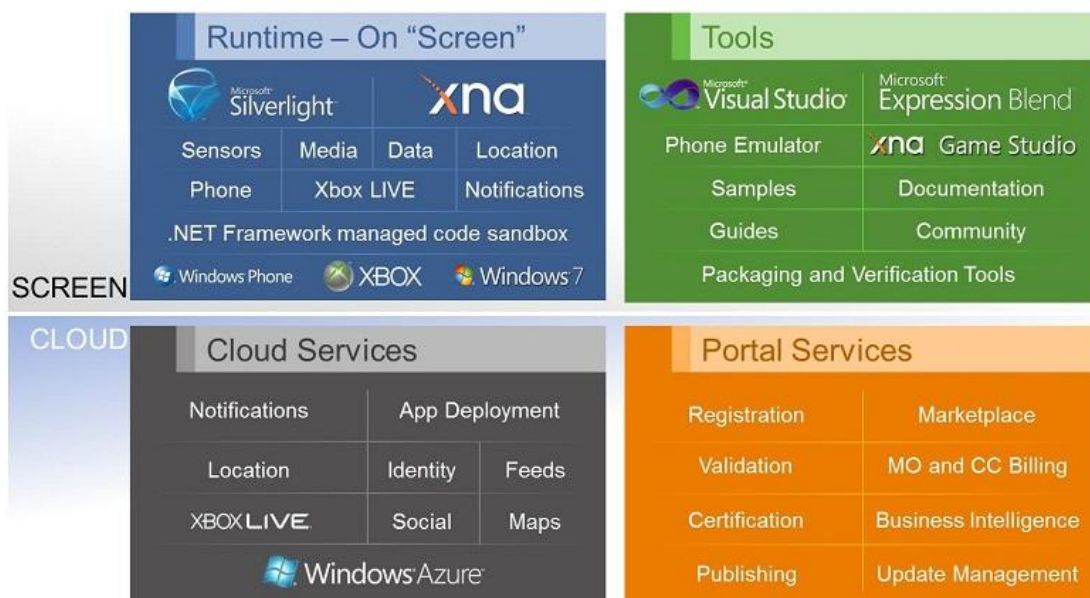
Windows Phone 7 -sovellusten kehittämiseen tarvittavat työkalut on mahdollista ladata ilmaiseksi Microsoftin sivuilta osoitteesta: <http://go.microsoft.com/?linkid=9713250>. Ladattava paketti sisältää seuraavat kehitystyökalut:

- Visual Studio 2010 Express for Windows Phone
- Windows Phone Emulator
- Silverlight for Windows Phone
- Microsoft Expression Blend for Windows Phone
- XNA Game Studio 4.0.

Tässä paketissa ovat kokonaisuudessaan kaikki työkalut, joita voidaan tarvita sovellusta kehittäessä. [6 s.14–15]

4.2 Arkkitehtuuri

Windows Phone 7 -sovellusalustan arkkitehtuuri koostuu neljästä pääkomponentista (Kuva 4.1).



Kuva 4.1: Windows Phone -alustan pääkomponentit [4]

Runtime-On "Screen"

Tällä komponentilla tarkoitetaan kaikkea sitä, mitä voidaan tehdä tämän käyttöjärjestelmän omaavassa puhelimessa. Silverlight ja XNA Framework takaavat Windows Phone 7 -sovelluksen ominaisuuksien kanssa kattavan ympäristön, jolle voidaan rakentaa turvallisia ja graafisesti rikkaita sovelluksia. Näiden kahden alustan, puhelimen yksittäisten komponenttien ja yleisen luokkakirjaston avulla sovelluskehittäjät pystyvät käyttämään suurta määrää erilaisia komponentteja, joilla voidaan kehittää suuriakin sovelluksia puhelinympäristöön. Windows Phone 7 -alustalle kehitetyt sovellukset toimivat kaikissa saman käyttöjärjestelmän puhelimissa, koska Microsoft on määrittänyt minimi laitteistovaatimukset puhelinvalmistajille. Minimivaatimuksilla säästetään useiden puhelinriippuvaisten asetusten määrittämiseltä, joka auttaa paljon puhelinsovelluksen kehittäjää.

Silverlight on hyvä teknologia, kun halutaan toteuttaa rikkaiden Internet-sovellusten tyyliä käyttöliittymiä, koska se tarjoaa mahdollisuuden käyttää grafiikkaa, animaatioita ja mediaa kuten videoita. Puhelinsovellusta kehitetään lähes samalla tavalla kuin Internet-sivustoa. Puhelinsovelluksen käyttöliittymä voi sisältää useita eri sivuja aivan kuten Internet-sivustokin. Sivujen avulla kehittäjiä on helppo laajentaa sovellustaan koskematta olemassa olevaan ohjelmaan. Lisäksi Microsoft suosittelee sovelluksiin käytettäväksi Windows Phone -kontrolleja, jotka kuuluvat käyttöjärjestelmän normaaliin käyttöliittymään. Tällä taataan se, että puhelinsovelluksen käyttökokemus ei muutu käyttöjärjestelmän ja sovellusten välillä, vaikka sovellus olisi kolmannen osapuolen kehittämä.

XNA Framework on kokoelma ohjelmistoja, palveluita ja resursseja, jotka mahdollistavat pelikehittämisen Microsoftin pelialustoille. Microsoftin tarjoamalla teknologialla ammattikehittäjät pystyvät nopeasti toteuttamaan pelejä Windows Phone, Xbox 360, Zune HD ja Windows 7 -ympäristöihin. XNA Framework tarjoaa suuren määrän ohjelmistorajapintoja pelinkehitykseen. Ohjelmistorajapinnat sisältävät 2D sprite -pohjaiset rajapinnat, jotka tukevat kääntämistä, skaalaamista, venyttämistä ja suodatusta. 3D-rajapinnat tukevat geometriaa, tekstuureita, valaistusta ja varjostusta.

Puhelimessa on myös erilaisia antureita, jotka palauttavat tietoja, joita voidaan hyödyntää puhelinsovelluksen kehityksessä. Esimerkiksi kiihtyvyyssanturilla saadaan tieto kun puhelinta käännellään kädessä.

Isolated Storage on Silverlightin tarjoama ns. hiekkalaatikko, jonne sovelluskehittäjät voivat tallentaa sovelluksen tarvitsemia tai sovelluksen käytön yhteydessä tarvittavaa tietoa. Kaikilla sovelluksilla on oma hiekkalaatikko, josta ei ole suoraa pääsyä taustalla pyörivän käyttöjärjestelmän tiedostojärjestelmään. Tämä estää sovelluksen tietojen pääsemisen väriin käsiin tai tuhoutumisen jonkun toisen sovelluksen toimesta.

Microsoft on myös kehittänyt yksinkertaisen ohjelmistorajapinnan, josta sovellukset voivat kysyä käyttäjän fyysistä sijaintia. Sovellukset voivat hankkia tiedon:

- nykyisestä sijainnista
- sijainnin muutoksesta
- suunnasta ja nopeudesta
- etäisyydestä kahden paikan väliltä.

Työkalut

Visual Studio 2010 on uusin Microsoftin kehittämä työkalu, jonka avulla pystytään kehittämään kaikkia .NET Frameworkkiin kuuluvia kieliä. Ohjelmistokehityksen tuo mukanaan paljon ominaisuuksia, jotka auttavat suunnittelijoita työssään. Graafisen käyttöliittymän kautta pystyy tekemään paljon perusasioita sovelluksiin ilman, että joutuu kaiken kirjoittamaan koodilla. Visual Studiossa ohjelmoijia helpottaa myös Intellisense, jonka avulla koodieditori ennustaa mahdollisia vaihtoehtoja koodia kirjoittaessa. Visual Studio sisältää myös graafisen editorin, jonka avulla on helppo luoda graafisia käyttöliittymiä.

Expression Blend on kehitystyökalu, jonka avulla kehittäjät pystyvät helposti luomaan rikkaita Internet-sovelluksia, mutta nyt myös Windows Phone 7 -sovelluksia. Blend on paras työkalu, kun halutaan luoda näyttäviä Silverlight-sovelluksia, koska siinä on paremmat ominaisuudet XAML-kuvauskielen hallintaan kuin Visual Studiossa. Blendin avulla pystyy helposti muuttamaan kaikkien kontrollien attribuutteja graafisesta käyttöliittymästä koskematta XAML-koodiin. Silverlight on tunnettu myös animaatioista, joiden avulla sivustoihin ja muihin sovelluksiin saadaan näyttävyyttä ja käyttökokemusta parannettua. Blendin avulla on helppo luoda animaatioita Silverlightiin, koska siinä on graafinen työkalu animaatioiden aikajanan muokkaamiseen elementti elementiltä ja animaation toiminnan pystyy näkemään kääntämättä ohjelmaa. Expression Blend on hyvä työkalu, kun tehdään Silverlight-sovellusten graafista kehitystä, mutta ei taustakoodin

kirjoittamiseen. Kun sovellukseen tehdään toiminnallisuutta taustakoodia apuna käyttäen, se kannattaa tehdä Visual Studiolla, koska se omaa paremman koodieditorin ja Intellisensen.

Windows Phone -emulaattori on Visual Studioon ja Expression Blendiin integroitu Windows Phone 7 -puhelinkäyttöjärjestelmä. Emulaattorin avulla voidaan kehitetty puhelinsovellus käynnistää suoraan sovelluskehittimistä puhelimeen, jolloin sovelluksen testaaminen ja virheiden korjaus on paljon helpompaa ja tehokkaampaa, kuin siirtää sovellus joka kerta fyysiseen puhelimeen testaamista varten. Emulaattorilla ei tosin ole kaikkia ominaisuuksia kuin oikealla puhelimella. Esimerkiksi sillä ei pystytä testaamaan puhelimen antureita käyttäviä sovelluksia.

Pilvipalvelut

Pilvipalvelujen avulla voidaan käsitellä suuria määriä dataa ja lähettää puhelimelle ainoastaan tarvittavat tiedot, minkä avulla voidaan parantaa puhelimien suorituskykyä ja akun kestoa. Puhelinsovelluksella voidaan siirtää tietoja eri pilvipalveluiden välillä. Yhteydet myös kolmannen osapuolen tekemiin verkkopalveluihin on tuettu. Pilvipalvelurajapintoina toimivat esimerkiksi seuraavat:

- **Ilmoitukset** – Windows Phone 7 -sovellusalusta tarjoaa rajapinnan pilvipohjaiselle ilmoituspalvelulle, jonka kautta pystytään lähettämään ilmoituksia sovelluksille kun jotain tärkeää tapahtuu. Pilvipalvelun käyttäminen ilmoitusten lähettämiseen tuo suuren hyödyn puhelimeen, koska puhelinsovelluksen ei tarvitse tarkistaa tilaansa jatkuvasti vaan ilmoitukset tulevat puhelimen ulkopuolelta.
- **Sijainti pilvipalvelu** – Tämä palvelu toimii yhdessä puhelimesta löytyvän sijainti ohjelmistorajapinnan kanssa. Palvelu voi käyttää langatonta yhteyttä, verkkomastoja ja GPS-tietoja yksittäisen laitteen paikantamiseen.
- **Syötteet, Sosiaaliset mediat ja Kartta palvelut** – Pilvessä on lukuisa määrä verkkopalveluita, joiden kautta käyttäjä voi käyttää sosiaalisen

median sivustoja, vastaanottaa syötteitä sivustoista ja käyttää karttaa navigointiin. Sovelluskehittäjät pystyvät viemään pilvipalvelujen käyttöä pidemmälle hyväksikäyttämällä näitä palveluita omissa sovelluksissaan.

Portaalipalvelut

Windows Phone Marketplace -sovelluskaupan kautta sovelluskehittäjät pystyvät helposti jakamaan sovellustaan kaikille Windows Phone 7 -käyttäjille. Portaalissa olevat työkalut auttavat sovelluskehittäjää julkaisemaan oman sovelluksensa. Sovellus julkaistaan .xap-tiedostona, joka sisältää pakattuna kaikki sovelluksen tiedostot, jotka tarvitaan sen suorittamiseen puhelimessa.

Kehittäjät pystyvät seuraamaan oman julkaisunsa tilaa portaalista. Kun sovellus on hyväksytty, siitä lähetetään viesti sovelluskehittäjälle, minkä jälkeen kehittäjä voi julkaista sovelluksensa sovelluskauppaan. Lisäksi kehittäjät voivat hinnoitella oman sovelluksensa tai julkaista sen ilmaisena ja kerätä rahaa mainostuloilla. Sovelluskauppa tukee sovellusten maksamista suoraan puhelinoperaattorin kautta tai luottokortilla, joka takaa vaivattoman tavan ladata myös maksullisia sovelluksia.

Sovelluksen kehittäjä pääsee myös näkemään tilastoja julkaisemastaan ohjelmasta. Tilastoista näkee esimerkiksi kuinka paljon sovellusta on käytetty, raportteja myynnistä ja käyttäjien palautteet. [6 s.16–20]

4.3 Käyttöliittymä

Windows Phone 7 -käyttöliittymän ulkoasu perustuu alkujaan Metro nimiseen designiin, joka on saanut inspiraationsa lentokenttien ja metroasemien kylteistä (Kuva 4.2).



Kuva 4.2: Esimerkkejä käyttöliittymän suunnittelun inspiraatioin lähteistä [7]

Metro-tyylinen design on luotu seuraten seuraavaa viittä periaatetta:

- Puhdas, kevyt, avoin ja nopea, joka tekee siitä visuaalisesti erottuvan. Käyttöliittymä sisältää runsaasti yksiväristä tilaa, joka vähentää sekavuutta ja nostaa typografian keskeiseksi suunnittelun osaksi.
- Käyttöliittymä keskittyy sisältöön, josta käyttäjä välittää eniten. Tällä tavoin tuotteeseen on kaikkien helppo päästä sisälle.
- Integroidulla laitteistolla ja ohjelmistolla käyttöliittymä takaa saumattoman käyttökokemuksen, josta yhden napin painalluksella on pääsy hakuun, aloitussivulle ja edelliselle sivulle.
- Windows Phone 7:n kosketus ja eleet kapasitiivisella näytöllä ovat yhdenmukaisia Windows 7:n kanssa ja se sisältää laitteistokiihdytettyjä animaatioita ja siirtymiä parantamaan käyttäjäkokemuksia.

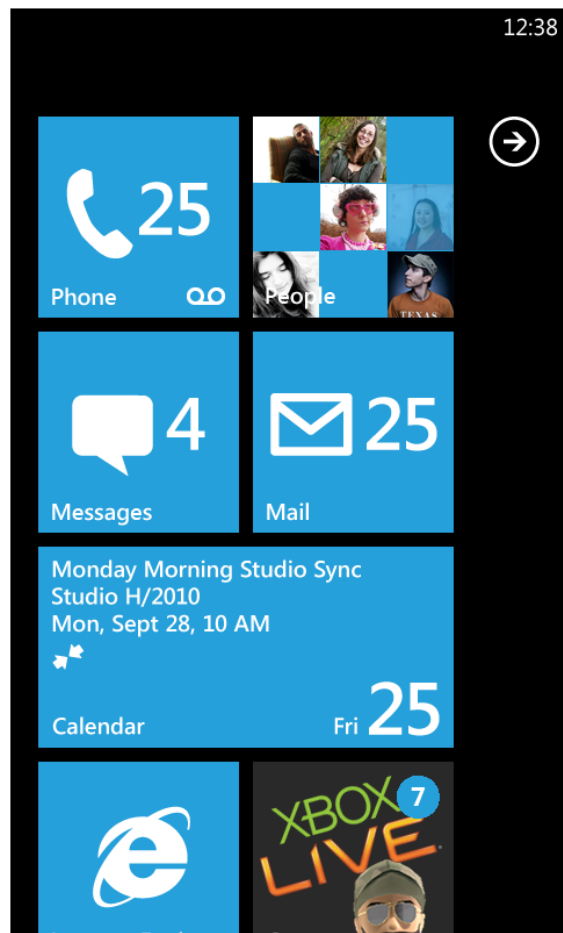
- Käyttöliittymä on mahdollista muokata henkilökohtaisten tarpeiden mukaan ja sitä täydentävät automaattisesti päivittyvät tiedot, jotka ovat käyttäjälle tärkeimpiä. Lisäksi täysin integroitu Zune-mediasoitin tuo mukanaan kuva-, video- ja musiikkiominaisuudet.

Microsoft suosittelee, että Windows Phone 7 -kehittäjät hyväksyisivät Metro-design tyylin omiin sovelluksiinsa. Tätä varten Microsoft on julkaissut käyttöliittymän mallintamiseen dokumentin, joka tarjoaa tiedot, perusteet ja ohjeet Metro-designin noudattamiseen. Vaikka vaatimukset ja toteutukset vaihtelevat sovelluksesta toiseen, Metro-tyylisiä elementtejä käyttäen voidaan luoda paljon johdonmukaisempi käyttökokemus käyttäjälle. Käyttöjärjestelmän Metro-tyyliä mukailevat sovellukset helpottavat sovellusten käyttöä, koska ne toimivat samalla tavalla kuin itse puhelin. Windows Phone 7 -kehitystyökalut sisältävät kokoelman Metro-tyylisiä kontrolleja, joiden avulla on helppo noudattaa puhelimen tyyliä myös omissa sovelluksissa. [7 s.21–22]

4.3.1 Start-aloitussivu

Windows Phone 7 -käyttöjärjestelmän aloitussivu koostuu erikokoisista tiilistä. Tiilet ovat interaktiivisia kuvakkeita, jotka muuttavat muotoaan sovelluksen vaatiessa. Tiilet voivat esimerkiksi ilmaista, jos käyttäjä on saanut uuden viestin. Käyttäjä pystyy muokkaamaan aloitussivua haluamalla tavalla esimerkiksi laittamalla siihen eri sovelluksia tai suoraan jonkun ystävän kontaktin.

Painamalla yhtä tiiltä aloitussivulta päästään haluttuun ohjelmaan, joka avautuu sivun päälle. Järjestelmään on tehty myös yksinkertainen listaus kaikista asennetuista sovelluksista, joihin pääsee siirtymällä yhden ruudun oikealle aloitusruudusta. Listauksen vuoksi käyttäjien ei tarvitse kuin lisätä käytetyimmät sovellukset ja kontaktit tiiliksi, jotka nopeuttavat ja tehostavat puhelimen käyttöä (Kuva 4.3).



Kuva 4.3: Windows Phone 7 -aloitussivu [7]

4.3.2 Sovellusten käyttöliittymä

Kun Windows Phone 7 -käyttöjärjestelmään luodaan sovellusta, Microsoft suosittelee kahta erilaista käyttöliittymämallia: Panorama ja Pivot.

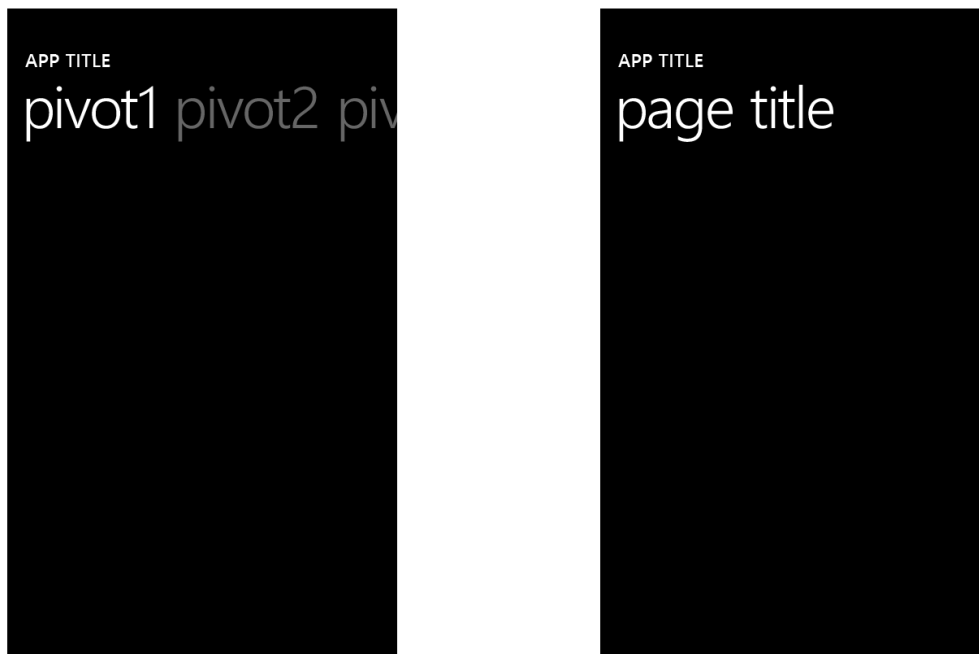
Panorama-käyttöliittymä on erilainen normaaleihin puhelinsovelluksiin verrattuna, koska se voi sisältää useita sovelluksen sivuja vierekkäin samanaikaisesti. Sivuja pystytään vaihtamaan sormella liikuttamalla vasemmalle tai oikealle (Kuva 4.4).



Kuva 4.4: Sovelluksen käyttöliittymä käyttäen Panorama-kontrollia [7]

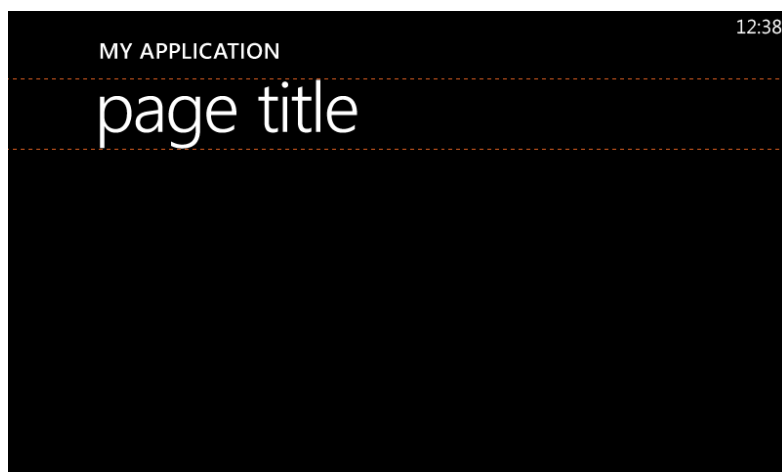
Panorama-kontrolli koostuu eri kerroksisista osista, jotka toimivat itsenäisen logiikan mukaan. Käyttöliittymän taustakuva, otsikko ja sivut liikkuvat eri kerroksilla, mikä luo näyttävän käyttökokemuksen kun kontrollia liikutetaan sivusuunnassa. Panorama-kontrollin käytetyin elementti on tiili, joka tekee käyttökokemuksesta täydellisen. Tiilet ovat interaktiivisia neliöitä tai suorakulmioita, ja ne voivat olla linkitettyinä ulkopuolisiin tapahtumiin, joiden kautta niiden ulkomuotoa voidaan muuttaa reaaliaikaisesti aivan kuten aloitussivullakin.

Toinen vaihtoehto käyttöliittymän luontiin on Pivot. Pivot-kontrolli on paljon yksinkertaisempi kuin Panorama, ja se muistuttaakin enemmän perinteisiä puhelinsovelluksia. Pivot-käyttöliittymässäkin on useita sivuja, mutta ne toimivat vain pystysuunnassa. Pivot-käyttöliittymän navigointi tapahtuu yläreunaan tulevalla valikolla, jossa ovat sivujen otsikot. Valikkoa pystyy liikuttamaan sormella ja valitsemaan haluamansa sivun sovelluksesta. Sivuja pystyy myös vaihtamaan sormella reunasta reunaan liikuttamalla (Kuva 4.5).



Kuva 4.5: Sovelluksen käyttöliittymä käyttäen Pivot-kontrollia

Pivot-käyttöliittymä sopii hyvin pieniin sovelluksiin, joissa esimerkiksi listataan paljon tietoa. Useissa tapauksissa Pivot-näkymää käytetään yhdessä sovelluksen kanssa, jonka päänäkymä on luotu Panorama-kontrollilla. Esimerkiksi jos sovelluksessa on sivu jolle tulee suuri määrä erilaista tietoa, pystytään Pivot-kontrollilla yksinkertaisesti jakamaan tieto eri sivuille, jolloin tiedon esittäminen on paljon luontevampaa. Pivot-käyttöliittymä voidaan laittaa myös toimimaan vaakatasossa toisin kuin Panorama-näkymä, joka tuo käytettävyyttä esimerkiksi puhelimiin joissa on fyysinen näppäimistö (Kuva 4.6). [7 s.22–25]



Kuva 4.6: Pivot-kontrolli käännettynä vaakatasoon

5 OPEN DATA PROTOCOL

5.1 Protokollan toiminta

Open Data Protocol (OData) mahdollistaa HTTP-pohjaisten datapalveluiden luomisen. Datapalvelun avulla voidaan kuvata abstrakti tietomalli tietokannasta syötteeksi, jota voidaan lukea ja muokata sovelluksista yksinkertaisilla HTTP-kutsuilla. OData on tarkoitettu mahdollistamaan tietojen esittäminen eri lähteistä kuten relaatiotietokannoista, tiedostojärjestelmistä, sisällön hallintajärjestelmistä ja perinteisiltä verkkosivuilta.

OData perustuu XML-rakenteiseen hierarkiaan. OData tuottama syöte koostuu kokoelmista, johon kuuluu erinäinen määrä merkintöjä. Jokainen merkintä rakentuu listasta ominaisuuksia, joihin voidaan lisäksi määritellä erilaisia tietotyyppisiä esimerkiksi päivämäärä, kokonaisluku tai merkkijono. Merkinnät voivat olla osa suurempaa hierarkiaa, ja ne voivat olla suhteessa toisiin merkintöihin ja kokoelmiin linkkien kautta. Yksinkertaiset datapalvelut pitävät sisällään ainoastaan yhden kokoelman. Monimutkaisemmat datapalvelut voivat sisältää useampia kokoelmia, minkä takia on hyvä julkaista lista kaikista korkeimman kerroksen kokoelmista, jotta asiakkaat pystyvät näkemään kaikki datapalvelun suhteet. Seuraavassa datapalvelu syötteessä näkyvät Signbook-relaatiotietokannan korkeimman tason kokoelmat, jotka vastaavat relaatiotietokannassa tauluja:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<service xml:base="http://localhost:2222/SignbookDataService.svc/"
xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:app="http://www.w3.org/2007/app"
xmlns="http://www.w3.org/2007/app">
  <workspace>
    <atom:title>Default</atom:title>
    <collection href="Comments">
      <atom:title>Comments</atom:title>
    </collection>
    <collection href="Signs">
      <atom:title>Signs</atom:title>
    </collection>
    <collection href="request">
      <atom:title>request</atom:title>
    </collection>
  </workspace>
</service>
```

```

</collection>
<collection href="User">
  <atom:title>User</atom:title>
</collection>
<collection href="User_friend">
  <atom:title>User_friend</atom:title>
</collection>
</workspace>
</service>

```

Syötteiden lisäksi OData-palvelun avulla voidaan käyttää palveluoperaatioita. Palveluoperaatiot ovat yksinkertaisia palvelufunktioita, joiden avulla voidaan palauttaa monimutkaisiakin tuloksia syötteestä tietokanta kyselyiden tapaan. Perus HTTP-pyyntöillä pystytään luomaan kyselyjä, päivityksiä, lisäyksiä ja poistoja OData-syötteen kautta.

Kehittäjien työn helpottamiseksi OData-palveluun on lisätty metadata-asiakirjamuoto, jonka avulla kehittäjä voi katsoa täydellisen kuvauksen Entity Data Modelista (EDM). EDM kuvaa koko tietomallin, joka on käytössä OData:n kautta. Metadata-asiakirja muotoinen syöte saadaan näkyviin kirjoittamalla osoitteeseen "\$metadata". Seuraava XML on pieni ote aikaisemmasta tietomallin hierarkiasta. Metadata-asiakirjasta nähdään relaatiotietokannan request taulun sisältä löytyvät kentät, niiden tyypit ja ominaisuudet: [8 s.27–28]

```

<EntityType Name="request">
  <Key>
    <PropertyRef Name="request_id" />
  </Key>
  <Property Name="request_id" Type="Edm.Int32" Nullable="false"
  p8:StoreGeneratedPattern="Identity"
  xmlns:p8="http://schemas.microsoft.com/ado/2009/02/edm/annotation" />
  <Property Name="user_email" Type="Edm.String" Nullable="false"
  MaxLength="50" Unicode="true" FixedLength="false" />
  <Property Name="title" Type="Edm.String" Nullable="false" MaxLength="50"
  Unicode="true" FixedLength="false" />
  <Property Name="description" Type="Edm.String" Nullable="false"
  MaxLength="Max" Unicode="true" FixedLength="false" />
  <Property Name="datetime" Type="Edm.DateTime" Nullable="false" />
  <Property Name="valid" Type="Edm.Boolean" Nullable="true" />
  <NavigationProperty Name="User"
  Relationship="SignDBModel.FK_request_User" FromRole="request"
  ToRole="User" />

```

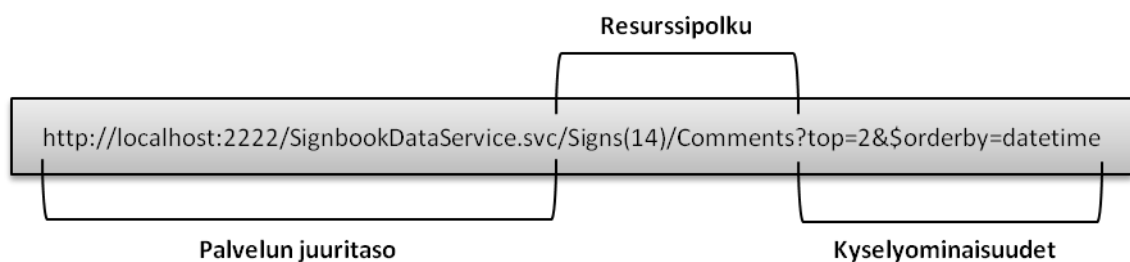
</EntityType>

5.2 URI:n merkintätavat

ODatan kanssa voidaan käyttää Uniform Resource Identifiersia (URI) apuna, kun syötteestä halutaan hakea vain tiettyjä tietoja. Yksinkertaisilla HTTP-viesteillä voidaan esimerkiksi määrittää, mistä relaatiotietokannan taulusta halutaan tietoja hakea. Lisäksi hakuun voidaan lisätä esimerkiksi järjestely tai muita normaaleja kyselyominaisuuksia, jotka ovat tuttuja SQL-kielestä. OData-palvelun käyttämä URI koostuu kolmesta osasta:

- palvelun juuritasosta
- resurssipolusta
- kyselyominaisuuksista.

Kuvassa 5.1 on esimerkki URI:stä, jolla haetaan syöte Signbookin relaatiotietokannasta. Esimerkissä haetaan tietokannan Sign-taulussa olevan viittoman kaksi uusinta kommenttia ja järjestellään ne päivämäärän mukaan. Kuvaan on eritelty kaikki kolme osaa. Palvelun juuritaso on OData-palvelun osoite. Juuritason jälkeen URI:hin määritellään resurssipolku Sign-taulun viittomaan, jonka avainkentässä tietokannassa on numero 14. Viittoman määrittelyn jälkeen voidaan resurssipolkuun määritellä tietokannan taulun relaatio toiseen tauluun, tässä tapauksessa kyseisen viittoman kommentteihin. Lopuksi syötteen esitystapaa voidaan muokata kyselyominaisuuksilla. Kyselyominaisuudet sisältävät perus SQL-kielestä tuttuja tapoja esimerkiksi järjestellä ja suodattaa syötteen tuloksia.



Kuva 5.1: Esimerkki Open Data Protocol -syötteen URI:stä

OData-verkkopalvelun juuritasolla tarkoitetaan tarkkaa osoitetta palvelimella sijaitsevaan syötteeseen. OData:n käyttämiseen ei välttämättä tarvita muuta kuin juuritason osoite, jos käytössä on esimerkiksi OData Client Library, jonka avulla pystytään hallitsemaan koko OData-syötettä ilman käyttämättä resurssipolkua tai kyselyominaisuuksia.

Resurssipolku mahdollistaa navigoinnin syötteen sisällä. Polkuun pystyy määrittelemään esimerkiksi tietokannan taulun, jonka jälkeen syöte näyttää ainoastaan kyseisen taulun sisällön. Resurssipolun avulla voidaan myös hyödyntää tietomallissa olevia suhteita taulujen välillä. Polkuun voidaan esimerkiksi määritellä näytettäväksi juuritaulun yhteen kenttään liittyvät kentät toisesta taulusta suhteen avulla.

Resurssipolkuun voidaan määrittää erilaisia sääntöjä, jotka muistuttavat paljon SQL-kyselyitä. Sääntöjen käyttäminen ei ole pakollista ODataa käyttäessä, mutta niiden avulla voidaan helposti hakea vain tarvittavat tiedot, joka nopeuttaa tiedonkeruuta huomattavasti. OData-syötteen URI:hin voidaan määrittää seuraavat kyselyominaisuudet:

- \$orderby – Pystytään järjestämään syötteen tulokset määritellyn kentän mukaisesti joko nousevasti tai laskevasti.
- \$top – Syötteestä voidaan valita tietty määrä sisältöä alusta eteenpäin. Tätä on hyvä hyödyntää \$orderby ominaisuuden kanssa, jolloin voidaan taulusta ottaa esimerkiksi 5 uusinta lisäystä.
- \$skip – Toimii samaan tyyliin kuin \$top ominaisuus, mutta tällä ominaisuudella ylitetään alusta eteenpäin määritelty määrä sisältöä.
- \$filter – Suodatuksen avulla voidaan hyödyntää erilaisia operaatioita hakiessa dataa kuten SQL kielessäkin. Käytettävät operaatiot selviävät taulukosta numero 5.2.

Taulukko 5.2: Suodatuksessa käytettävät operaatiot [8]

Operator	Description	Example
Logical Operators		
Eq	Equal	/Suppliers?\$filter=Address/City eq 'Redmond'
Ne	Not equal	/Suppliers?\$filter=Address/City ne 'London'
Gt	Greater than	/Products?\$filter=Price gt 20
Ge	Greater than or equal	/Products?\$filter=Price ge 10
Lt	Less than	/Products?\$filter=Price lt 20
Le	Less than or equal	/Products?\$filter=Price le 100
And	Logical and	/Products?\$filter=Price le 200 and Price gt 3.5
Or	Logical or	/Products?\$filter=Price le 3.5 or Price gt 200
Not	Logical negation	/Products?\$filter=not endswith(Description,'milk')
Arithmetic Operators		
Add	Addition	/Products?\$filter=Price add 5 gt 10
Sub	Subtraction	/Products?\$filter=Price sub 5 gt 10
Mul	Multiplication	/Products?\$filter=Price mul 2 gt 2000
Div	Division	/Products?\$filter=Price div 2 gt 4
Mod	Modulo	/Products?\$filter=Price mod 2 eq 0
Grouping Operators		
()	Precedence grouping	/Products?\$filter=(Price sub 5) gt 10

- \$expand – Tämä ominaisuus tulee hyödylliseksi kun kyselyssä täytyy käyttää taulujen välisiä suhteita apuna. Kun syötteeseen halutaan listata päätaulun sisältö mukaan lukien tiedot toisesta taulusta, jotka liittyvät päätaulun kenttiin on expand-ominaisuus paras ratkaisu. Muuten jouduttaisiin tekemään kaksi eriaikaista kyselyä, joilla aluksi haettaisiin päätaulun tiedot ja toisella taas päätauluun liittyvät tiedot.
- \$format – Voidaan muuttaa syötteen rakennetta toiseen. Käytettävissä on Atom Publishing Protocol (AtomPub) ja JSON-syötteet.
- \$select – Toimii samaan tapaan kuten SQL-kielessäkin. Tämän avulla voidaan valita yksi tai useampi haluttu kenttä, jonka arvo palautetaan syötteeseen tietokannasta.
- \$inlinecount – Laskentaominaisuus, jonka avulla voidaan laskea kuinka monta merkintää tietokannan taulussa on. [9 s.29–31]

6 SIGNBOOK-PUHELINSOVELLUS

Puhelinsovellus kehitettiin Signbook-sovellukseen, jotta sen sisältöön pystyisi pääsemään käsiksi muullakin tavalla kuin tietokoneella. Puhelinsovelluksen avulla viittomakielen kanssa työskentelevät pystyvät hakemaan viittomia milloin tahansa, jos omistavat älypuhelimien ja datayhteyden. Sovellusta pystyvät hyvin hyödyntämään esimerkiksi tulkit, jotka mahdollisesti tarvitsevat jotain tiettyä viittomaa työssään. He voivat tarkistaa viittoman vaivattomasti käyttämällä puhelinta.

6.1 Käyttöliittymä

Signbook-puhelinsovellukseen haluttiin tuoda mahdollisimman paljon samoja ominaisuuksia, joita löytyy verkkosivustolta. Koska tiedettiin, että sovellukseen tulee paljon ominaisuuksia, jotka mahdollisesti tarvitsevat oman sivun tiedon esittämiselle, tuntui parhaalta vaihtoehdolta käyttää Panorama-käyttöliittymää.

Signbook-sovelluksen Panorama-käyttöliittymä koostuu valikosta ja ystävälistauksesta. Valikko on luotu käyttäen puhelimen oman alkuvalikon tapaan tiiliä. Tiilivalikon lisäksi selaamalla Panorama näkymää oikealle löytyy käyttäjän ystävälistaus, josta voi aloittaa esimerkiksi keskustelun ystävän kanssa. Panorama-tyyliä käyttämällä pystytään yksinkertaisesti laajentamaan ohjelmaa esimerkiksi, jos halutaan lisätä tiiliä valikkoon tai halutaan luoda lisää sivuja ystävät sivun oikealle puolelle (Kuva 6.1).



Kuva 6.1: Signbook-sovelluksen Panorama-kontrolli

Microsoft ei ollut vielä julkaissut virallista Panorama-kontrollia puhelimeen esiselvitystyön kehitysvaiheessa, joten esiselvityksessä käytettiin kolmannen osapuolen kehittämää kontrollia, joka on toiminnoiltaan samanlainen kuin virallinen kontrolli tulee todennäköisesti olemaan. Epävirallinen Panorama-kontrolli on julkaistu CodePlex-sivustolla. [10]

Kontrollin käyttäminen omassa sovelluksessa on todella yksinkertaista. Aluksi ladataan kontrollin luokkakirjastot CodePlex-sivustolta ja lisätään projekti oman puhelinsovellusprojektin sisään. Lisäyksen jälkeen sovelluksen aloitussivun XAML-tiedostoon lisätään seuraava viittaus kontrollin luokkakirjastoon:

```
xmlns:phone="clr-
namespace:Phone.Controls.Samples;assembly=Phone.Controls.Sa
mples"
```

Viittauksen jälkeen Visual Studio tunnistaa ulkopuolisen kontrollin projektista ja sitä voidaan käyttää sovelluksessa. Panorama-kontrolliin voidaan määrittää myös parametrit "Title" ja "BackgroundImage", joiden avulla saadaan lisättyä Panorama-tyyliin kuuluva otsikko ja vaihtaa taustakuvaksi sovellukseen sopivampi kuva. Panorama-kontrolli lisätään aloitussivulle seuraavalla tavalla:

```
<phone:PanoramaControl x:Name="panoramaMenu"
                        Title="signbook"
                        BackgroundImage="SignbookBG.png">
</phone:PanoramaControl>
```

Kun itse Panorama-kontrolli on lisätty, voidaan aloittaa sivujen lisääminen kontrollin sisään. Jokainen Panorama-kontrollin osa, vastaa yksittäistä sivua. Kontrolli lukkiutuu automaattisesti eri sivujen kohdalle, jolloin on mahdollista selata sivua myös alaspäin ilman, että sivu liikkuu sivusuunnassa. Tämä mahdollistaa pidemmänkin listan hyödyntämisen Panorama-sivulla, esimerkiksi ystävälistauksen. Panorama-sivujen vaihtaminen vaatii pidemmän sormiliikkeen näytön toisesta reunasta toiseen, jolloin sivu vaihtuu seuraavaan sulavan animaation avulla. Kuten itse Panorama-kontrollissa myös jokaiselle sivulle voi antaa muutamia tarpeellisia parametreja, kuten "Header", "Width" ja "AutoSnap", joiden avulla esimerkiksi sivun otsikon laittaminen käy todella yksinkertaisesti. Jos sivulla käytetään tiiliä eivätkä kaikki tiilet mahdu kerralla näyttöön, tulee silloin määrittää sivulle suurempi leveys ja ottaa automaattinen lukkiutuminen pois päältä, jolloin voidaan siirtää kontrollia sen verran, että näytön ulkopuolelle jääneet tiilet saadaan näkyviin tarvittaessa ilman, että sivu vaihtuu seuraavaan. Panorama-kontrolliin voidaan määrittää niin monta sivua kuin halutaan ja niitä on todella yksinkertaista lisätä myös jälkeinpäin, jos sovellusta halutaan laajentaa. Panorama-kontrollin sivu määritellään seuraavalla tavalla:

```
<phone:PanoramaItem Header="my book"
                    Width="580"
                    AutoSnap="False">

    <Grid x:Name="TileGrid">
```



```

<Grid.RowDefinitions>
  <RowDefinition Height="183"/>
  <RowDefinition Height="183"/>
</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Orientation="Horizontal">
  <!-- Ylärivin tiilet tässä (Esimerkki)-->
  <Image x:Name="TileSearch"
    Source="images/tiles/search.png"
    Width="173" Height="173"
    MouseLeftButtonDown="TileSearch_MouseLeftButtonDown">
</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal">
  <!-- Alarivn tiilet tässä -->
</StackPanel>
</Grid>

</phone:PanoramaItem>

```

6.2 Toiminnot

6.2.1 Termien haku

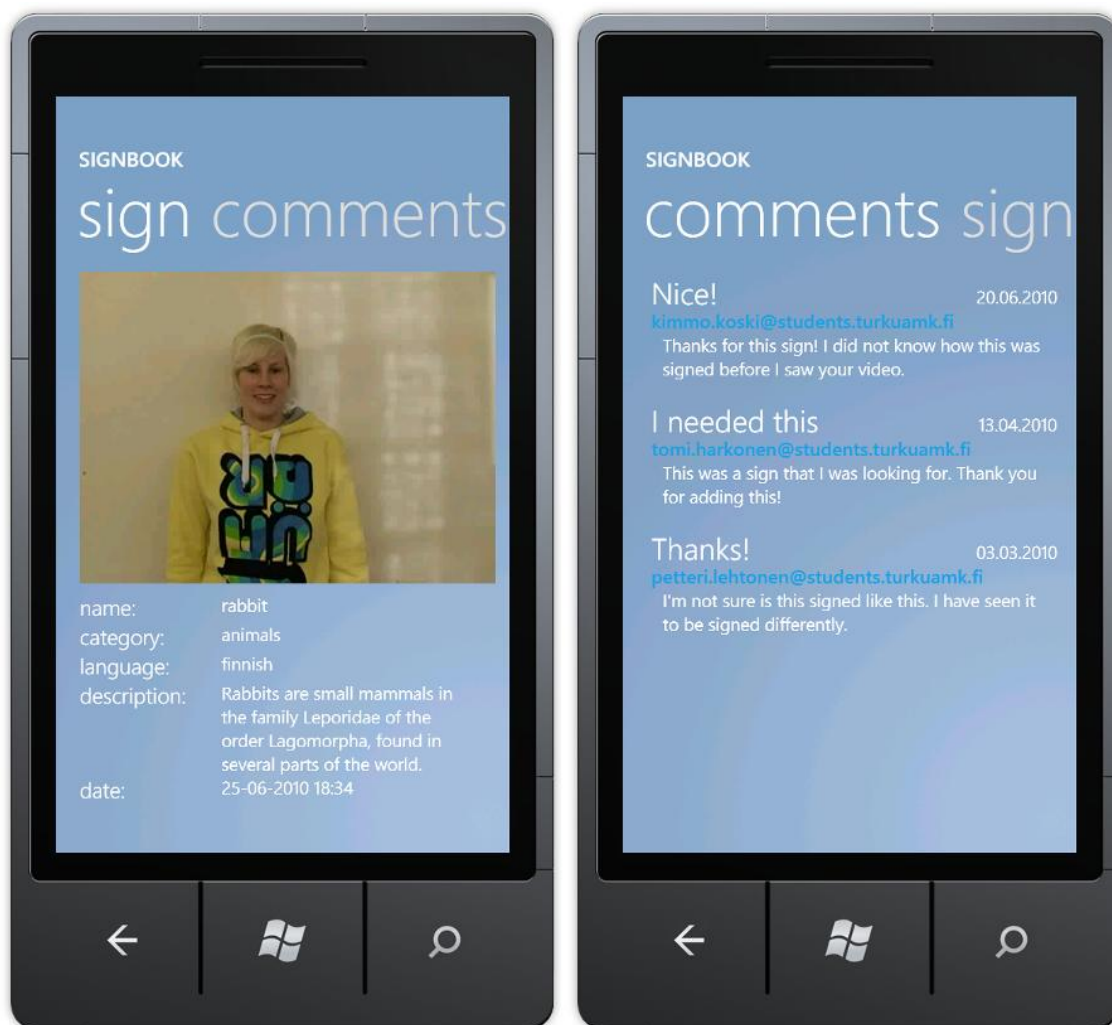
Termien hakeminen on todennäköisesti käytetyin ominaisuus. Siitä syystä hakusivusta tehtiin mahdollisimman yksinkertainen. Sivulle tullessaan käyttäjä näkee ainoastaan tekstikentän ja hakupainikkeen. Kun tekstikenttä aktivoidaan, tulee näppäimistö automaattisesti näkyviin ja käyttäjä voi kirjoittaa haluamansa hakusanan. Haku etsii kaikkia sanoja, joiden yhteydessä jossain kohdassa sanaa kyseinen termi sijaitsee. Hakutulokset muodostuvat listana hakukentän alapuolelle. Tuloksista voi nähdä viittoman suomenkielisen vastineen sekä mihin semanttiseen kategoriaan kyseinen termi kuuluu. Kun yhtä tuloksista painetaan, päästään kyseiseen tietueeseen (Kuva 6.2).



Kuva 6.2: Signbook-sovelluksen hakusivu

6.2.2 Tietuesivu

Viittomien katsomisen tulisi olla vaivatonta ja helppoa. Tiedossa on, että yhteen tietueeseen kuuluu viittomakielinen vastine videona, termin tiedot ja kommentit, joten on järkevää käyttää Pivot-tyylistä sivua. Pivot-tyylisellä sivulla voidaan jättää ensimmäiselle sivulle ainoastaan tärkeimmät asiat, kuten viittoman video ja tiedot. Muut tiedot voidaan laittaa omille sivuille, joita voidaan helposti selata Pivot-kontrolliin kuuluvalla valikolla (Kuva 6.3).



Kuva 6.3: Signbook-sovelluksen viittomasivut käyttäen Pivot-kontrollia

Pivot-kontrollin käyttäminen on yhtä yksinkertaista kuin Panorama-kontrollin. Pivot-kontrolli löytyy samasta CodePlexin julkaisemasta paketista. Kontrolli luo valikollisen sivun, joka sisältää useimpia puhelimen näytön kokoisia sivuja. Pivot-kontrolli luodaan seuraavalla tavalla:

```
<phone:PivotControl Name="pivotSign">
</phone:PivotControl>
```

Kun itse kontrolli on luotu, sen sisälle voidaan lisätä sivuja. Kuten Panorama-kontrollissa, myös Pivot-kontrollin sivuihin voidaan määrittellä parametreja, kuten "Title" ja "Header". Parametrien avulla määritellään sivun nimi, johon tässä tapauksessa määritettiin ohjelman nimi Signbook. Kontrolli

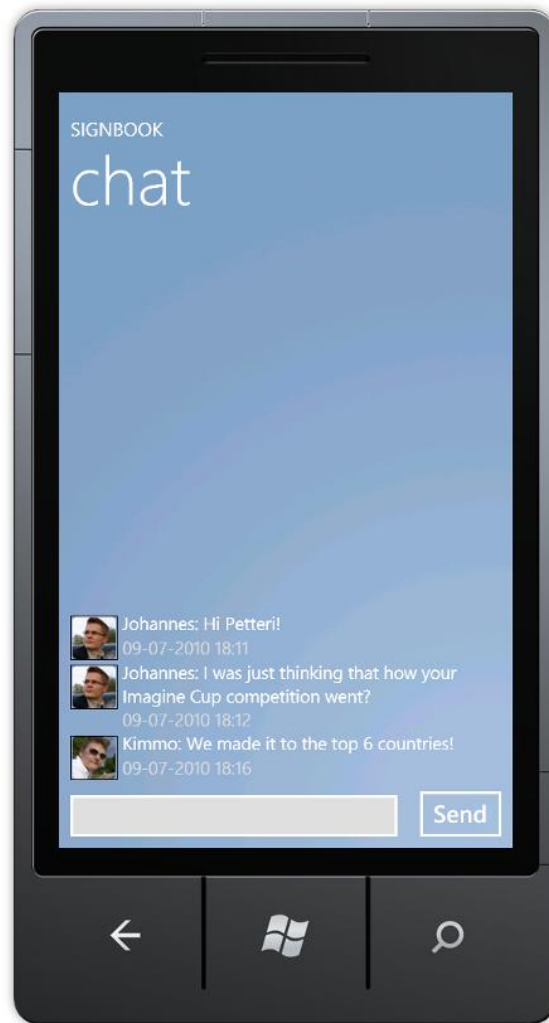
vaatii sivun otsikon määrittämisen jokaiseen sivuun, koska se on keskeisessä roolissa vaihdettaessa sivuja. Kontrolli luo valikon sivun yläreunaan, jonne sivujen otsikot menevät ja niitä painamalla voi selata kontrollia eteenpäin. Pivot-kontrollin sivu luodaan seuraavalla tavalla:

```
<phone:PivotItem Title="SIGNBOOK" Header="sign"
Background="Transparent">
  <StackPanel Orientation="Vertical">
    <MediaElement x:Name="SignVideoElement" Stretch="Fill"
AutoPlay="False" Width="440" Height="330"/>
    <!-- Videon alle tulee viittoman tiedot -->
  </StackPanel>
</phone:PivotItem>
```

Pivot-kontrollia on yksinkertaista käyttää ja sitä kannattaa käyttää aina, kun kyseiseen osaan ohjelmaa on tulossa paljon tietoa, jota ei pysty järkevästi esittämään yhdellä sivulla. Monet Windows Phone 7 -ohjelmat on tehty täysin käyttäen Pivot-kontrollia, jos esimerkiksi ohjelma ei ole niin laaja että siihen tarvittaisiin suurempia valikoita.

6.2.3 Chat

Signbookin sosiaalisen median elementit ovat tärkeitä, koska käyttäjät lisäävät ja arvioivat sivuston sisältöä. Sosiaalistuminen ja yhteisöllisyys koetaan tärkeänä sivuston toiminnan kannalta. Signbookkiin on tehty mahdollisuudet lisätä omaan profiiliin ystäviä, joiden kanssa voi keskustella viittomakielistä yleensä, termityöstä tai jostain tietystä käsitteestä. Tämä mahdollisuus luotiin myös puhelinsovellukseen, jonka avulla käyttäjä voi olla yhteydessä Signbook-ystäviin myös puhelimen kautta (Kuva 6.4).



Kuva 6.4: Signbook-sovelluksen keskustelusivu

6.2.4 Oma profiili -sivu

Oma profiili -sivulta käyttäjät pystyvät muuttamaan oman profiilin asetuksia. Asetuksiin kuuluu muun muassa kirjautumisasetukset ja käyttäjän lisätiedot, joita hän mahdollisesti haluaa jakaa muille Signbook-käyttäjille.

6.2.5 Suosikit

Koska Signbook-sovellusta käytetään usein opiskeluun tai nopeaan termin tarkistamiseen, haluttiin puhelinsovellukseen lisätä suosikkilistaus. Suosikkilistauksen avulla käyttäjä pystyy esimerkiksi lisäämään tietyn kategorian termit suosikeiksi, minkä jälkeen ne löytyvät helpommin

sovelluksesta. Listaus toimii myös hyvin opiskelun yhteydessä. Jos opiskelijalla on esimerkiksi opiskeltavana tietyn pää- tai alaluokan kuten tekniikan/tietotekniikan alan termistö, ne voidaan lisätä listaukseen ja pitää niitä siellä kunnes ne ovat opittu. Myös hankalimpien termien tallennus on hyvä tehdä suosikkilistaan, josta ne löytyvät nopeasti.

6.2.6 Offline-viittomat

Offline-latausominaisuus haluttiin tuoda myös puhelinsovellukseen, jolloin käyttäjät pääsevät käsiksi esimerkiksi suosikkilistalla oleviin termeihin ilman datayhteyttä, koska videoiden katsominen tarvitsee useasti vähintään 3G-yhteyden toimiakseen pätkimättä, eikä sitä ole aina saatavilla. Toiseksi, jos käyttäjällä ei ole kiinteää datayhteyttä puhelimessa voi hän ladata tarvittavat termit offline-tilaan kun käytössä on langaton yhteys.

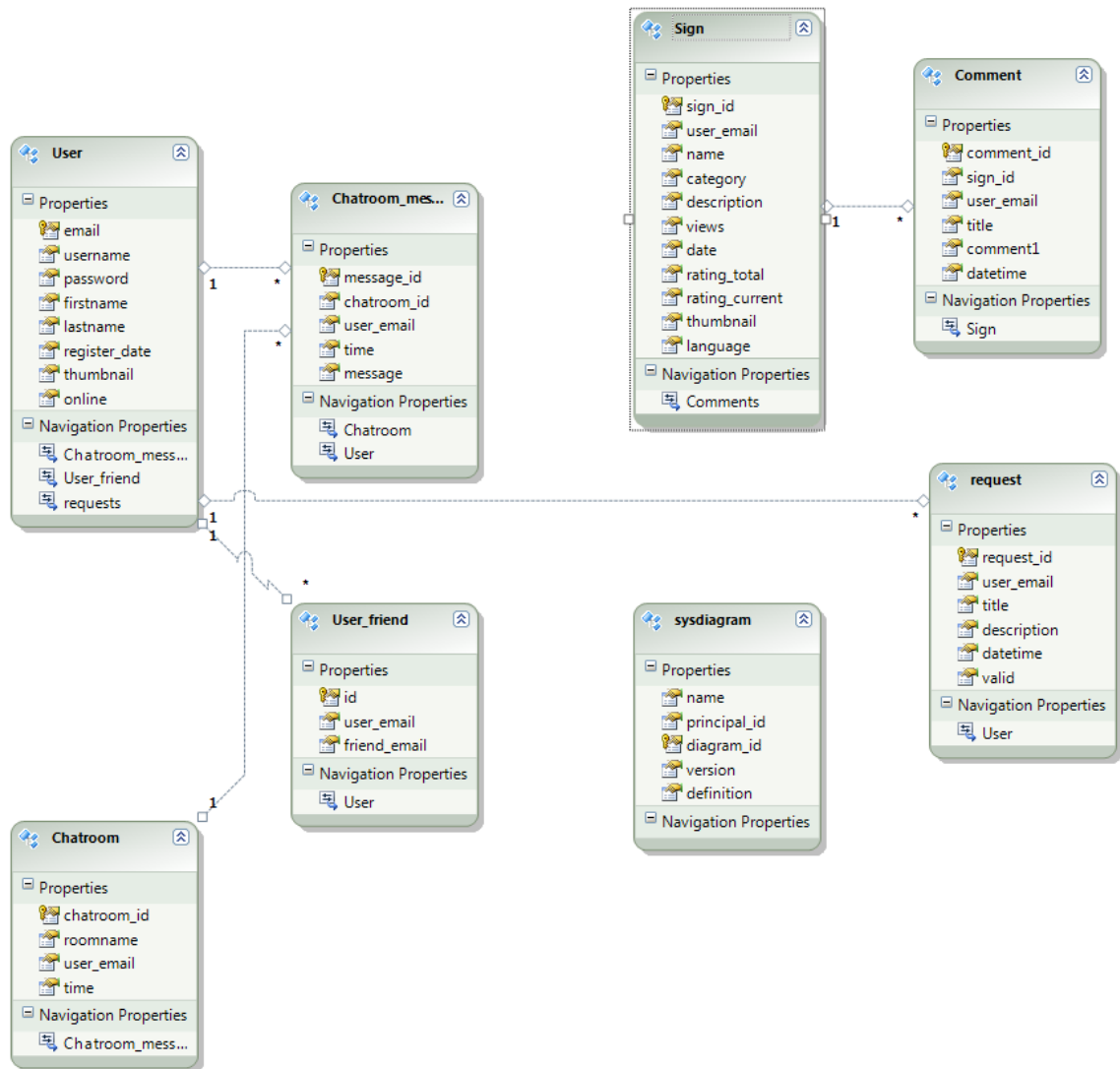
6.3 OData

6.3.1 OData-verkkopalvelun luominen

Kaikki sisältö puhelinsovellukseen tuodaan Signbookin Azure SQL -tietokannasta käyttäen Open Data Protocol -verkkopalvelua. OData:n avulla puhelimessa pystytään käyttämään samaa tietoa, joka on verkkosivustolla ja sovellus pysyy aina ajan tasalla. OData-palvelua pyöritetään myös pilvipalvelimella, jolloin se on aina käytössä eri kohteisiin.

Ennen kuin OData-palvelu voidaan ottaa käyttöön, tarvitaan ADO.NET Entity Data Model (EDM), joka on tietomalli Signbookin tietokannasta. Kun tietomallia luodaan, siihen määritellään yhteysasetukset haluttuun tietokantaan. Kun yhteys tietokantaan on saatu, pitää määritellä mitä tietokannan osia halutaan lisätä tietomalliin. Kun kyseessä on OData-palvelun käyttämä tietomalli, kannattaa valita koko tietokannan sisältö, koska sen käytettävyyttä voidaan rajoittaa erikseen OData-palvelun oikeuksilla.

Kun tietomalli on luotu, se avautuu kuvan 6.5 tapaisena graafisena kuvauksena, josta näkee kaikki käytössä olevat taulut ja niiden ominaisuudet ja suhteet toisiin tauluihin. Tietomallin avulla OData-verkkopalvelu pystyy hyödyntämään samoja ominaisuuksia kuin normaali relaatiotietokanta, kuten suhteita taulujen välillä.



Kuva 6.5: Graafinen näkymä Signbook-tietokannan tietomallista

Seuraavaksi projektiin lisätään Data Service. Kun Data Service lisätään projektiin, Visual Studio tekee automaattisesti valmiin rungon palvelun luomiseen. Palvelu kannattaa nimetä järkevällä tavalla, koska tiedoston nimi tulee olemaan OData-palvelun verkko-osoitteessa.

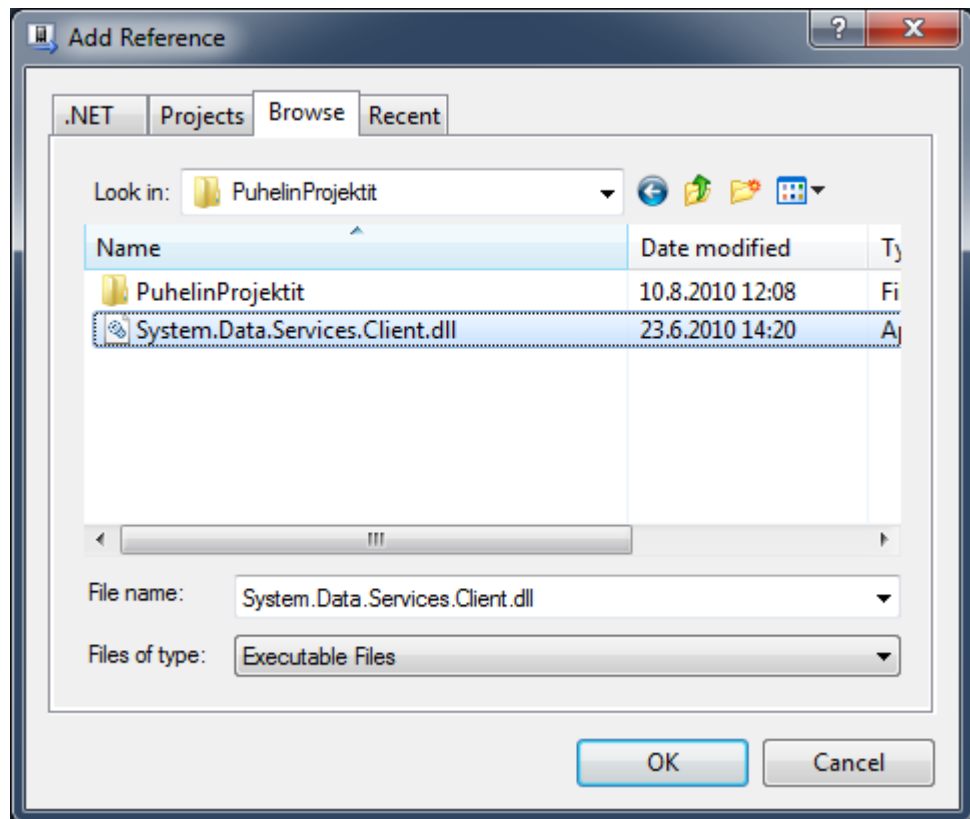
Aluksi palvelu tulee liittää aiemmin luotuun tietomalliin; tässä tapauksessa se on nimeltä `SignDBEntities`. Kun palvelu on liitetty oikeaan tietomalliin, voidaan aloittaa sääntöjen määrittelemisen. Sääntöjen avulla voidaan antaa tai rajoittaa oikeuksia palvelun käyttäjiltä. Esimerkkikoodissa Data Service on liitetty `SignDBEntities`-tietomalliin ja palvelun käyttöön on ainoastaan sallittu luku-oikeudet Sign-tietokantatauluun. Tällä tavoin OData-syötteessä ei näy kaikkia tietokannan osia, vaikka tietomalliin kaikki valittiinkin. Sääntöjä voi palveluun lisätä rajattomasti, mutta kannattaa ainoastaan tarjota syötteet niihin tietoihin, jotka ovat tarpeellisia jotta palvelu ei joudu lataamaan turhaa tietoa. Kun säännöt on määritelty, voidaan palvelu käynnistää pilvipalvelimelta. OData-syötettä voi kaikki käyttää ja sen avulla on mahdollista, että tulee myös muita kolmannen osapuolen sovelluksia Signbookista, jos suosio kasvaa.

```
namespace SignODATA
{
    // SignDBEntities on tietomallin nimi.
    public class SignbookDataService :
        DataService<SignDBEntities>
    {
        public static void
        InitializeService(DataServiceConfiguration config)
        {
            // Määritellään säännöt OData syötteen käyttöön.
            config.SetEntitySetAccessRule("Sign",
            EntitySetRights.AllRead);
        }
    }
}
```

6.3.2 ODatun käyttäminen

ODatan käytön helpottamiseksi Windows Phone 7 -puhelinsovelluksissa Microsoft on julkaissut OData Client Libraryn, joka on tehty käsittelemään ODatun-syötteitä. Client Libraryllä on monia samoja ominaisuuksia kuin ADO.NET Data Services Client:llä, jota käytetään Silverlight-sovelluksissa kun niissä halutaan käyttää tietokantoja. ODataa pystyy käyttämään myös ilman kyseistä kirjastoa, mutta silloin OData-syötettä joutuu käsittelemään XML:nä jolloin tiedonkeruu syötteestä on paljon työläämpää.

ODatan käyttäminen aloitetaan lisäämällä OData Client Library puhelinsovellusprojektin References-kansion sisään (Kuva 6.6).



Kuva 6.6: Lisätään Client Library -tiedostojärjestelmästä projektiin

Visual Studio ei vielä täysin tue OData:n käyttöä, eikä se automaattisesti osaa muodostaa luokkakirjasto OData-syötteen osoitteesta. Luokkakirjaston OData:n käyttämästä tietomallista pystyy luomaan .NET 4.0 Frameworkin mukana tulevalla DataSvcUtil.exe sovelluksen avulla. Tiedosto sijaitsee %windir%\Microsoft.NET\Framework\v4.030319 -kansiossa. Luokkien luominen tapahtuu seuraavalla komennolla komentorivillä:

```
DataSvcUtil.exe
/uri:http://localhost:3333/SignbookDataService.svc
/out:SignbookEntities.cs /Version:2.0
/DataServiceCollection
```

Komennossa URI osoittaa verkko-osoitteeseen, jossa OData on käynnissä. Out tarkoittaa tiedostonnimeä, jonka sovellus luo. Kun sovellus on ajettu,

tiedosto ilmestyy samaan kansioon, josta sovellus ajettiin. Tämän jälkeen `SignbookEntities.cs` -tiedosto pitää lisätä puhelinsovellus projektiin. Kun tiedosto löytyy projektista, osaa OData Client Library käyttää kyseisessä osoitteessa olevaa syötettä.

Esimerkkinä luodaan viittomien haku käyttäen OData-syötettä. OData:n käsittely tapahtuu kokonaan taustakoodissa, joka on toteutettu C#-ohjelmointikielellä. Aluksi hakusivulle pitää ilmoittaa missä tarvittavat kirjastot sijaitsevat:

```
using System.Data.Services.Client;
using SignDBModel;
using SignbookMobile.ViewModels;
```

ODatan käsittelyyn tarvitaan OData Client Library ja tietomallin luokkakirjasto. Lisäksi käytetään ViewModels-luokkia, joissa syötteestä haetut tiedot lisätään olioihin.

Kun käytössä on tarvittavat kirjastot, voidaan hakusivulle kertoa missä OData-syöte sijaitsee. Sivulle luodaan uusi `SignDBEntities` olio, jolle kerrotaan syötteen verkko-osoite seuraavalla tavalla:

```
SignDBEntities SignDB = new SignDBEntities(new
Uri("http://localhost:3333/SignbookDataService.svc/"));
```

Osoitteen jälkeen on mahdollista aloittaa kyselyn tekeminen syötteeseen. Tässä tapauksessa kysely aloitetaan kun käyttäjä on kirjoittanut hakusanan tekstikenttään ja painanut haku painiketta.

```
private void ButtonSearch_Click(object sender,
RoutedEventArgs e)
{
    var queryGetSearchedSigns = SignDB.Sign;

    queryGetSearchedSigns.BeginExecute(
        delegate(IAsyncResult asyncResult)
        {
            Dispatcher.BeginInvoke(
                () =>
                {
                    DataServiceQuery<Sign> query =
                        asyncResult.AsyncState as DataServiceQuery<Sign>;
```

```

        if (query != null)
        {
            var signs = query.EndExecute(asyncResult);
            this.DataContext = new
                SignSearchViewModel(signs, TextBoxSearch.Text);
        }
    });
},
queryGetSearchedSigns);
}

```

Aluksi luodaan kyselylle muuttuja, jolle kerrotaan mihin tauluun kysely halutaan muodostaa. Termejä haetaan suomenkielisellä vastineella, joten muuttujalle kerrotaan `SignDB.Sign`, josta `SignDB` on aluksi luotu tietomalliolio ja `Sign` on tietomallissa oleva tietokantataulu.

Kysely tehdään OData-syötteeseen kuten muihinkin Silverlight-sovelluksiin asynkronisena, jotta käyttöliittymä pysyy kokoajan käytettävissä. Ennen kun kutsutaan ViewModellia, varmistetaan että sovellus on saanut yhteyden OData-syötteeseen. Kun tiedonkeruu on tehty onnistuneesti, luodaan DataContext jolla kutsutaan ViewModellia. ViewModel-kutsuun sisällytetään syötteestä saatu tulos ja hakukenttään kirjoitettu hakusana.

ViewModel on yksinkertainen luokka, jolla voidaan määritellä mitä tietoja halutaan lopulliseen olioon, joka palautetaan käyttöliittymälle.

```

public SignSearchViewModel(IEnumerable<Sign> Signs, string
SearchText)
{
    Items = new ObservableCollection<Sign>();

    foreach (Sign viittoma in Signs)
    {
        if (viittoma.name.Contains(SearchText))
        {
            Items.Add(viittoma);
        }
    }
}

public ObservableCollection<Sign> Items {get; private set;}

```

ViewModel luokassa luodaan kokoelmaolio, johon tallennetaan halutut tiedot kyselyn tuloksista. Hakusivun tapauksessa käydään silmukalla läpi syötteestä tulleet tiedot ja katsotaan missä tietueissa kenttä "name" sisältää hakukenttään kirjoitetun tekstin. Löydetty tietueet lisätään kokoelmaan, joka palautetaan takaisin hakusivun DataContext muuttujaan.

Koska Silverlight-sovelluksissa käytetään asynkronisia kyselyitä, pitää tietokannasta haettujen tietojen tulla käyttöliittymään ilman, että sivua joudutaan päivittämään. Tähän käytetään Data Binding -ominaisuutta, jonka avulla DataContextissa olevan kokoelmaolion tietoja voidaan tuoda näkyviin käyttöliittymään ilman, että näkymää joudutaan päivittämään.

```
<ListBox x:Name="ListBoxSearchResults"
ItemsSource="{Binding Items}"
SelectionChanged="ListBoxSearchResults_SelectionChanged">
  <ListBox.ItemTemplate>
    <DataTemplate>
      <Grid>
        <Grid.RowDefinitions>
          <RowDefinition/>
          <RowDefinition/>
        </Grid.RowDefinitions>

        <TextBlock Grid.Row="0" Text="{Binding name}"
Style="{StaticResource PhoneTextLargeStyle}"/>

        <TextBlock Grid.Row="1" Text="{Binding category}"
Style="{StaticResource PhoneTextSmallStyle}"/>

      </Grid>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```

Listbox-kontrolliin on helppo liittää tietoja kokoelmasta, koska se tukee Data Binding -ominaisuutta. Aluksi kontrollille annetaan tietolähde, joka on aikaisemmin ViewModel-luokassa luotu kokoelmaolio nimeltä "Items". Lähteen antamisen jälkeen voidaan kontrollissa käyttää kaikkia tietoja, jotka sijaitsevat kokoelmassa. Seuraavaksi kontrolliin luodaan tietomalli, joka on malli

siitä miten tiedot tullaan esittämään käyttöliittymässä. Hakusivun tapauksessa listataan ainoastaan tietueen nimi ja kategoria omiin tekstikehyksiin, joka tapahtuu yksinkertaisesti viittaamalla "name" ja "category" tietoihin, jotka sijaitsevat kokoelmassa. Näin ollen listbox-kontrolli tulostaa samalla tavalla allekkain kaikki tietueet jotka löytyy kokoelmasta (Kuva 6.7).



Kuva 6.7: Signbook-sovelluksen hakutulos "ra" -hakusanalla

7 JATKOKEHITYS

Puhelinsovellus on kehittynyt paljon esiselvitystyön aikana. Esiselvitystyön alussa sovellus kehitettiin Windows Mobile -käyttöjärjestelmälle, josta kuitenkin siirryttiin uuteen Windows Phone 7 -ympäristöön, kun se oli mahdollista. Samalla kun sovelluksella siirryttiin uuteen ympäristöön, siihen suunniteltiin paljon uusia ominaisuuksia, joita uudella ympäristöllä voidaan hyödyntää.

Esiselvityksen aikana ilmaantui ongelma, joka liittyy ODataan. Ongelmana on tietokannan suuntaan tehtävät kyselyt, joita ei vielä pystytty tekemään. Ongelmaan etsittiin ratkaisua pitkään ja siihen kokeiltiin erilaisia vaihtoehtoja, mutta sitä ei vielä ole saatu ratkaistua. Ongelma voi johtua useasta eri asiasta, jonka takia ongelman paikantaminen on hyvin hankalaa. Vika voi johtua tietokannasta, ODatasta tai puhelinsovelluksen taustakoodista. OData on tosin vielä kehitysasteella, joten odotettavissa on, että sen kehittymisen myötä tulevat tarkemmat ohjeet sen käyttämiseen, joita ei vielä ole puhelinsovellukseen lainkaan.

Puhelinsovellus on jatkokehityksen kannalta edennyt hyvin, koska sen käyttöliittymän suunnittelussa otettiin huomioon sen laajentaminen. Sovelluksen Panorama-kontrollin valikkoon voidaan tarvittaessa lisätä tiliä ja sivuja, joista pystytään esimerkiksi avaamaan uusi sivu, joka käyttää Pivot-kontrollia. Tällä tavoin sovellusta voidaan kehittää edelleen.

Ennen kun Signbook-puhelinsovelluksesta aletaan tehdä tuotantoversiota, tulee sovellus luoda uudelleen käyttäen virallisia kehitystyökaluja. Lopulliset kehitystyökalut toivat mukanaan kaikki viralliset kontrollit.

8 YHTEENVETO

Tässä opinnäytetyössä oli tarkoituksena luoda rikas puhelinsovellus Windows Phone 7 -alustalle. Sovelluksen ulkoasun kehittämiseen käytettiin Microsoftin julkaisemaa käyttöliittymän suunnitteludokumenttia, jonka avulla luotiin käyttöliittymä käyttäen Microsoftin suosittamia tyylejä. Puhelinsovellus yhdistettiin tietokantaan Open Data Protocol -verkkopalvelulla, jonka avulla puhelimeen saatiin sama sisältö kuin verkkosovelluksella on.

Työstä voi havaita, miten helposti rikkaita puhelinsovelluksia pystyy luomaan Windows Phone 7 -alustalle, kun käytössä ovat kehitystyökalut, kuten Visual Studio ja Expression Blend. Windows Phone 7 -järjestelmä tarjoaa hyvin rikkaan käyttöliittymän ja käyttökokemuksen käyttäjälle Silverlightin avulla.

Puhelinsovelluksista on hankala yhdistää suoraan ulkopuoliseen tietokantaan verkon yli. Käyttäen Open Data Protocol -verkkopalvelua pystyttiin hakemaan informaatiota Signbookin pääjärjestelmän tietokannasta tehokkaasti. Open Data Protocolia ei ole tehty yksistään .NET-ympäristöön. Mikään ei estä OData:n käyttämistä myös muissa puhelinkäyttöjärjestelmissä ja työpöytäsovelluksissa. Syötteen avulla kolmannen osapuolen sovelluskehittäjät voivat luoda omia sovelluksiaan eri alustoille käyttäen samaa Signbookin sisältöä.

Signbook-puhelinsovellus on tässä vaiheessa vielä keskeneräinen joiltain osilta, mutta tästä esiselvityksestä on varmasti apua sovelluksen jatkokehityksessä.

LÄHTEET

- [1] Rikkaiden internet-sovellusten toteuttaminen Microsoft-teknologioilla – Petteri Lehtonen [www-dokumentti]
Saatavilla: <http://urn.fi/URN:NBN:fi:amk-2010062112412> (Luettu 07.2010)
- [2] Windows Phone 7 [www-dokumentti]
Saatavilla: http://en.wikipedia.org/wiki/Windows_Phone_7 (Luettu 06.2010)
- [3] Smartphone Platform U.S. Market Share [www-dokumentti]
Saatavilla:
http://www.comscore.com/Press_Events/Press_Releases/2010/7/comScore_Reports_May_2010_U.S._Mobile_Subscriber_Market_Share (Luettu 07.2010)
- [4] Android (operating system) [www-dokumentti]
Saatavilla: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)) (Luettu 06.2010)
- [5] iOS (Apple) [www-dokumentti]
Saatavilla: [http://en.wikipedia.org/wiki/IOS_\(Apple\)](http://en.wikipedia.org/wiki/IOS_(Apple)) (Luettu 06.2010)
- [6] Application Platform Overview for Windows Phone [www-dokumentti]
Saatavilla: [http://msdn.microsoft.com/en-us/library/ff402531\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/ff402531(v=VS.92).aspx) (Luettu 06.2010)
- [7] UI Design and Interaction Guide for Windows Phone 7 v2.0 [pdf-dokumentti]
Saatavilla: <http://go.microsoft.com/?linkid=9713252> (Luettu 07.2010)
- [8] Open Data Protocol - Introduction [www-dokumentti]
Saatavilla: <http://www.odata.org/developers/protocols/overview> (Luettu 07.2010)
- [9] Open Data Protocol – URI Conventions [www-dokumentti]
Saatavilla: <http://www.odata.org/developers/protocols/uri-conventions> (Luettu 07.2010)
- [10] CodePlex – Windows Phone 7 controls [www-dokumentti]
Saatavilla: <http://phone.codeplex.com/> (Luettu 06.2010)