



Jussi Villanen

Nomis VisitorCard -sovelluskehitys

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Tietotekniikan koulutusohjelma
Insinöörityö
12.11.2010

Tekijä Otsikko	Jussi Villanen Nomis VisitorCard -sovelluskehitys
Sivumäärä Aika	33 sivua 12.11.2010
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikan koulutusohjelma
Ohjaajat	tuotekehityspäällikkö Janne Ketola yliopettaja Kirsti Äystö
<p>Tämän insinööriyön aiheena on .NET-tekniikalla toteutettu selainpohjainen Nomis VisitorCard -vierailijakorttijärjestelmä. Tavoitteena oli toteuttaa vuoden 2010 aikana uusi versio jo olemassa olevasta Nomis Vierailijajärjestelmästä siten, että nykyinen MS Outlook -tuotteen riippuvuus saatiin purettua ja samalla laajennettua käyttöä Internet-selaimella toimivaksi. Selainpohjaisuuden ja Outlook-riippumattomuuden lisäksi sovelluksen toimintoja kehitettiin vastaamaan myös kasvavan tietoturvan vaatimuksiin.</p> <p>Uusi tuote tehtiin käyttäen nykyaikaisia sovelluskehitysympäristöjä, kuten Microsoft Visual Basic 2010:tä ja Microsoft .NET Framework 4.0:ää sekä tietokantana Microsoft SQL Server 2008 Express Edition with Advanced Servicesiä.</p> <p>Kehitysprojektin laajuus oli jo aloitusvaiheessa hyvin tiedossa, ja sen vuoksi projektiin saatiin varattua riittävästi aikaa ja resursseja. Tuloksena syntyi sovellus, joka korvaa aikaisemman tuotteen ja palvelee laajempaa asiakaskuntaa ja käyttötarkoitusta.</p>	
Hakusanat	sovelluskehitys, ohjelmointi, tietokanta, selainsovellus, .Net,

Author Title	Jussi Villanen Nomis VisitorCard software development
Number of Pages Date	33 pages 12 November 2010
Degree Programme	Information Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Janne Ketola, Software Development Manager Kirsti Äystö, Principal Lecturer
<p>This thesis is about the Nomis VisitorCard visitor card system development project. Nomis VisitorCard development is based on the .NET technology. The aim was to implement a new generation version of the system during 2010. One of the key goals was to replace the Microsoft Outlook/Exchange development platform (and system dependency upon the technology) with web-based technologies. The functions of the new system version were also developed to meet growing security requirements.</p> <p>The new product was made using modern application development environments such as Microsoft Visual Basic 2010, Microsoft .NET Framework 4.0 and Microsoft SQL Server 2008 Express Edition with Advanced Services as the database application.</p> <p>The scope of the development project was well-known at the beginning and therefore the project was allocated sufficient time and resources. The result is a system that replaces the old product and serves a broader customer base with increased use.</p>	
Keywords	application development, programming, database, browser-based software, .Net,

Sisällys

1	Johdanto	1
2	Nomis VisitorCard taustaa	2
3	Taustakartoitus	4
	3.1 Markkinoilla olevat järjestelmät	4
	3.2 Asiakasprospektien kehitysideat	4
	3.3 Aikataulu ja resurssit	5
4	Uuden järjestelmän suunnittelu	7
5	Käytetyt työvälineet ja menetelmät	10
	5.1 Kehitystyökalut	10
	5.2 Tietokanta	10
	5.3 Versionhallinta	11
	5.4 Kehitystiimi	13
	5.5 Testaus	14
6	Nomis Visitor Card -järjestelmän tekninen kuvaus	15
	6.1 Käyttöliittymät	15
	6.2 Tulostinvaatimukset	15
	6.3 Arkkitehtuuri	18
	6.4 Toimintaympäristö	20
7	Nomis VisitorCard -järjestelmän toiminnallinen kuvaus	21
	7.1 Kutsujan sovellus	21
	7.2 Osallistujan sovellus	24
	7.3 Vastaanoton sovellus	25
8	Jatkokehitys	27
9	Markkinointi	28
10	Tulokset	28

11 Päätelmä

32

Lähteet

33

Lyhenteet ja määritelmät

GUID	<i>Globally Unique Identifier.</i> Eriytyinen sovellusten käyttämä uniikki tunniste. Tunnisteen arvo on 32 merkkiä pitkä heksadesimaalimerkkijono.
IDE	<i>Integrated Development Environment.</i> Microsoft Visual Studio integroitu kehitysympäristö, jota käytetään esimerkiksi graafisten käyttöliittymien kehittämiseen.
IIS	<i>Internet Information Services.</i> Windows-palvelimen sovellus web-palveluiden, kuten web-sivustojen toiminnan hallintaan.
NDA	<i>non-disclosure agreement.</i> Tarkoittaa salassapitosopimusta, jonka tarkoituksena on turvata luottamuksellisen tiedon ja materiaalien säilyminen salaisena.
PERT	<i>Program Evaluation and Review Technique.</i> Työmenetelmä, jossa vaativa ja moniosainen projekti jaetaan yksittäisiksi aikataulutetuiksi työtehtäviksi.
SaaS	<i>Software as a Service.</i> Ohjelmiston tarjoaminen palveluna.
XHTML	<i>eXtensible Hypertext Markup Language.</i> www-sivujen merkintäkieli, joka täyttää XML:n muotoiluvaatimukset.

1 Johdanto

Tässä insinööriyössä käsitellään maaliskuussa 2010 alkanutta Nomis VisitorCard -vierailijakorttiohjelmiston uuden version sovelluskehityshanketta. Nomis VisitorCard on sovellus, jolla asiakasorganisaatiot voivat kutsua vierailijoita, tulostaa vierailijoilleen vierailijakortit sekä pitää kirjaa organisaation vierailijoista. Sovellus on jaettu neljään eri käyttöliittymään, jolla vierailijat ja oma organisaatio voivat sovellusta käyttää. Käyttöliittymiä ovat kutsujan, osallistujan ja vastaanoton sovellus sekä vastaanoton itsepalvelusovellus. Sovellukset ovat selainpohjaisia lukuun ottamatta vastaanoton sovellusta, joka on toteutettu Windows-sovelluksena. Selainsovelluksia voidaan käyttää lähes kaikilla nykyisillä (vuosi 2010) Internet-selaimilla, eikä niiden käyttö näin ollen vaadi työasemakohtaisia asiakasovellusasennuksia.

Osallistujien käyttöön tarkoitetulla käyttöliittymällä asiakkaat voivat täydentää omia yhteystietojaan, ilmoittautua saapuvaksi vierailulle sekä peruuttaa oman vierailun. Kutsujan sovelluksen käyttöliittymässä käyttäjät voivat valmistella uusia tapaamisia ja kutsua vierailijoita tapaamisiin. Vastaanoton sovellus mahdollistaa vierailijakorttien tulostuksen lisäksi raportointitoiminnot saapuvista, paikalla olevista sekä poistuneista vieraista. Sovelluksen yhtenä osana on salassapito-, niin kutsuttu NDA-rekisteri, johon kerätään tieto vierailijoiden allekirjoittamista salassapitosopimuksista sekä esimerkiksi konsulttien määräaikaisten vierailijaoikeuksien tiedoista.

Nomis Oy on vuonna 1993 perustettu suomalainen tietotekniikan palveluita tarjoava yritys. Tarjotut palvelut jakautuvat tasaisesti sovelluskehitykseen ja järjestelmätukipalveluihin. Sovelluskehityksen painopiste on ns. valmistuotteilla, joita ovat Nomis Help-Desk -työnohjausjärjestelmä sekä Nomis VisitorCard -vierailijajärjestelmä. Lisäksi Nomis Oy tuottaa asiakkaan tarpeisiin selainpohjaisia ja Windows-pohjaisia räätälöityjä sovelluskehitystuotteita. Järjestelmätuki kattaa palveluita asiakkaan ensimmäisen asenteen tuesta vaativiin asiantuntijapalveluihin.

Henkilökunnan koko Suomessa vuonna 2010 on 25. Nomis Oy:n pääkonttori sijaitsee Helsingin Sörnäisissä, tarkemmin Kalasatamassa. Suomessa Nomis Oy:n muut toimipisteet ovat Kauhavalla ja Jyväskylässä. Tytäryhtiö DRS Nordic Ab toimii Ruotsissa Hel-

singborgissa. Nomis Oy:n Aasian markkinoista vastaa Nomis Asia Ltd. jonka pääkonttori on Hong Kongissa ja sivukonttori Manilassa, Filippiineillä.

2 Nomis VisitorCard taustaa

Vanha järjestelmä

Ensimmäisen vierailijakorttisovelluksen "Nomis Vierailijajärjestelmä" kaupallinen versio julkaistiin vuosituhanteen vaihteessa, mutta sen jatkokehitys oli hidasta eikä sille ollut tehty varsinaista versiointisuunnitelmaa. Vanha järjestelmä on vieraiden ilmoittamiseen ja vastaanottoon suunniteltu sovellus, jonka avulla vierailijoille tulostetaan vierailijakortti vierailun ajaksi. Tapaamisen isäntä voi ennalta ilmoittaa vierailijan tiedot vastaanottopisteeseen yksinkertaisen Outlook-ilmoituslomakkeen avulla. Lomakkeelle kirjaetaan vierailun isännän nimi, kokousajankohta sekä kokouksen vieraiden nimet ja organisaatiot. Lomakkeella välitetyt tiedot näkyvät automaattisesti vastaanottopisteen sovelluksessa, josta vastaanottovirkailija voi tulostaa asiakkaiden vierailijakortit.

Nomis Vierailijajärjestelmä vaatii vastaanotossa toimiakseen Outlook-sähköposti-ohjelmiston sekä Windows-työaseman, johon itse vastaanottohenkilökunnan vierailijasovellus on asennettuna. Vieraiden ilmoittamista varten vaatimuksena oli ainoastaan työasemalle asennettu Outlook-ohjelmisto, johon oli asennettuna vieraiden ilmoittamiseen tarkoitettu räätälöity Outlook-lomake.

Uusi järjestelmä

Markkinoilla ilmenneen vierailijajärjestelmien kysynnän kasvun vuoksi Nomis Oy aloitti uuden vierailijajärjestelmän kehittämisen vuoden 2010 alkupuolella. Uusi järjestelmä sai nimen Nomis VisitorCard -ohjelmisto. Nimenvaihdon tarkoituksena oli ilmaista kahden täysin erilaisessa toimintaympäristössä toimivan sovelluksen eroavaisuus sekä tuoda esille myös kansainvälisesti ymmärrettävä nimi.

Ennen vuoden 2009 syksyä olemassa olevien asiakkaiden vaatimuksia vanhan järjestelmän kehittämiseksi tuli vähän ja ne olivat vaatimattomia. Tällä tarkoitan, että asiakkaat olivat varsin tyytyväisiä olemassa olevaan tuotteeseen eikä kehitykselle asetettu

suurempia paineita. Vuoden 2009 alussa saatiin kuitenkin toimeksianto tehdä yksi lisäkenttä räätälöityyn Outlook-lomakkeeseen ja tuoda tämä tieto näkyviin tulostettavalle vierailijakortille. Pyyntö oli kohtuullinen, mutta todettiin, että kyseisen toimeksiannon toteutus olisi työmäärältään melkoinen verrattaessa saatuun hyötyyn. Toteutettavan muutoksen kokonaistyömääräksi arvioitiin hiukan alle kaksi työpäivää.

Työmäärä-hyötysuhde oli heikko ja johtui lähinnä olemassa olevan sovelluksen kehitysympäristön joustamattomuudesta ja siitä, että alun perinkään sovellusta ei ollut tarkoitettu millään tavoin asiakaskohtaisesti räätälöitäväksi. Vanhan järjestelmän kehitysympäristönä toimi Microsoft Exchange/Outlook-sähköpostijärjestelmä. Varsinaiset kehittämiseen vaikuttavat haasteet olivat Microsoftin julkaisemien tuotteiden uudet versiot ja niissä ilmenneet yllättäen ja ennalta arvaamatta tulleet rajoitukset jatkokehityksen suhteen. Rajoituksia olivat esimerkiksi tietoturvaan liittyvät uudet ominaisuudet sekä käyttöliittymäasettelu. Tässä vaiheessa heräsi jo ajatus järjestelmän uudelleenkehittämisestä joustavammaksi.

Myöhään syksyllä 2009 kaksi uutta potentiaalista asiakasta kiinnostui Nomis Oy:n tarjoamista palveluista ja erityisesti Vierailijajärjestelmä-sovelluksesta. Perinteinen lyhyt Vierailijajärjestelmän esittely kalvoilla tuottikin useita varsin mielenkiintoisia jatkokysymyksiä sovelluksen ominaisuuksista – joita vanha vierailijajärjestelmä ei sisältänyt. Taustalla ollut ajatus järjestelmän jatkokehittämisestä sai tässä vaiheessa lisäpotkua, ja alkuvuonna 2010 lähdettiin toteuttamaan suunnitelmaa uuden järjestelmän kehitykselle.

3 Taustakartoitus

3.1 Markkinoilla olevat järjestelmät

Taustatutkimus tehtiin hyvin lyhyessä ajassa maaliskuu–toukokuussa 2010. Kartoitukseen osallistui kehitystiimistä vain kaksi jäsentä. Pääosin kartoitus tehtiin etsimällä internetistä tietyt piirteet täyttäviä tuotteita, jotka vastaisivat omaa kehitteillä olevaa tuotetta. Etsintä tehtiin globaalisti, eli vastaavien tuotteiden etsinnässä käytettiin myös englanninkielisiä hakutermejä. Kartoituksen tuloksena havaittiin, että markkinat eivät ole saturaatuneet tämän ohjelmiston osalta, sillä suoraa vastaavuutta omaavia tuotteita löytyi vain kolme kappaletta. Näistä löytyneistä kolmesta tuotteesta yksikään ei ollut kotimainen tuote. Käyttöliittymältään suomenkielisiä järjestelmiä ei löytynyt lainkaan.

3.2 Asiakasprospektien kehitysajat

Kyseessä olevalle tuotteelle ryhdyttiin kartoittamaan asiakastarvetta sekä nykyisistä, että mahdollisista tulevista asiakkaistamme. Kartoitusvaiheessa saatiin paljon kehitysideoita omien itse suunniteltujen uusien ominaisuuksien rinnalle. Taulukossa 1 on esitetty muutamia ominaisuuksia, joita saatiin kerättyä kartoituksen yhteydessä.

Taulukko 1: Asiakasprospektien kehitysajat.

Idea	Selitys
NDA-rekisteri	NDA-rekisteri on salassapitosopimusrekisteri. Rekisterissä ylläpidetään tietoa vieraista ja voimassaolevista sopimuksista. Jokaisen vierailijan tulee allekirjoittaa salassapitosopimus. Sopimus on voimassa määrätyn ajan, jonka jälkeen sopimus tulee uudelleen allekirjoittaa.
Vieraan tunnistaminen viivakoodilla	Ajatuksena olisi, että vieras voisi tulostaa kutsun jossa, olisi viivakoodi tunnistautumista varten. Tunnistautuminen voisi tapahtua esimerkiksi siten, että vierailijan kutsun viivakoodi luetaan viivakoodin lukulaitteella, jonka seurauksena vierailija tunnistetaan saapuneeksi.
Tapaamiskutsu sähköpostilla	Kutsuttavalle vieraalle lähetetään sähköpostikutsu suoraan kutsujan sovelluksesta.

Idea	Selitys
Web-pohjainen käyttöliittymä	Käyttöliittymien tulisi olla pääosin selainpohjaisia, jolloin sovelluksen käytettävyyttä voitaisiin helposti ulottaa organisaation ulkopuolelle.
Kutsuttavien osallistujien tietojen piilotus	Mahdollisuus syöttää käsin osallistujan tiedot niin, etteivät tiedot näy muiden osallistujien tietojen joukossa (osallistujan käyttöliittymässä).
Vierailun peruutus	Mahdollisuus peruuttaa osallistujan vierailu henkilöittäin yksitellen.
Kioski-toiminnallisuus	Sovelluksessa tulisi olla itsepalvelukäyttöliittymä siten, että asiakas voisi itse ilmoittaa itsensä saapuneeksi ja tulostaa itselleen vierailijakortin.

Kartoituksen ja saatujen kehitysideoiden perusteella oli havaittavissa, että tarve tai ainakin mielenkiinto uudelle versiolle oli selkeästi olemassa useammalla organisaatiolla.

3.3 Aikataulu ja resurssit

Kehitykseen liittyvän aikataulutuksen suunnittelu oli haasteellinen, sillä useat organisaatiot olivat heti valmiita ottamaan käyttöönsä uuden version Nomis VisitorCard-järjestelmästä. Olemassa olevien järjestelmän vaatimusten perusteella tehty raaka-arvio järjestelmän uudelleenkehityksestä oli minimissään kolme kuukautta. Järjestelmän kehitysaikataulu jaettiin suunnitteluvaiheen alussa neljään osaan

- taustakartoituksen tekeminen
- tietokannan ja sovelluksen määrittelyn ja suunnitelman loppuunsaattaminen
- järjestelmän kehittäminen
- asennuspaketin suunnittelu ja toteuttaminen, asennus pilottikäyttöön asiakkaiden ympäristöihin.

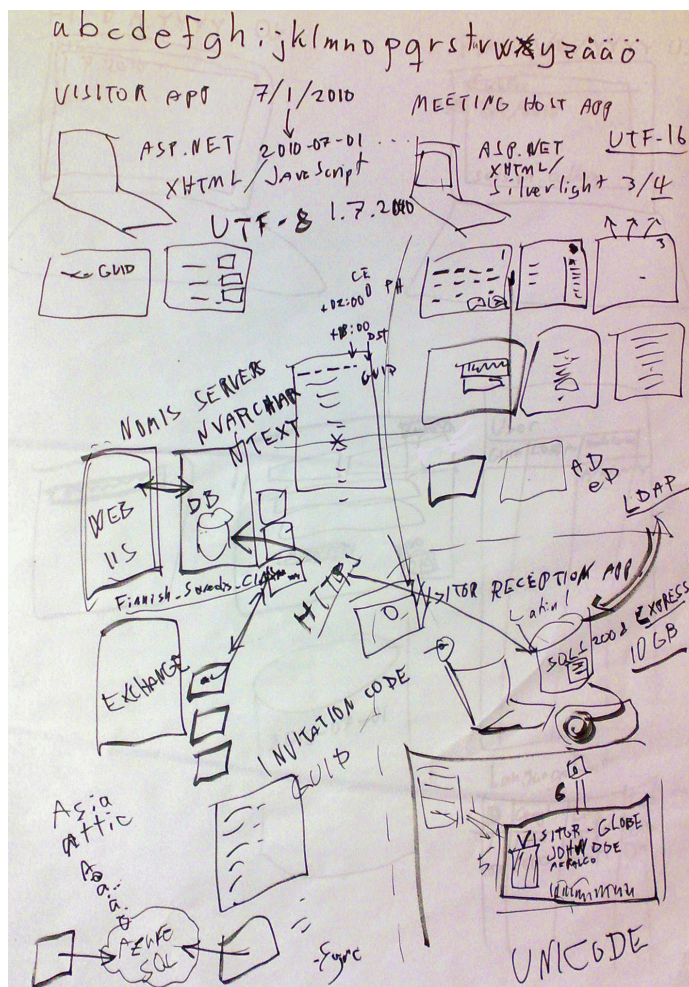
Aikaisemman vierailijajärjestelmän pienen asiakaskunnan sekä vähäisen kehitystarpeen vuoksi oli selvitettävä, miten sovelluskehitystiimi tulisi resursoida, sillä vanhalla järjestelmällä ei ollut olemassa olevaa kehitystiimiä, joka olisi voinut olla vastuussa uuden järjestelmän kehittämisestä. Sovelluskehittäjien osaaminen valittujen sovelluskehitystyökalujen sekä teknisen ympäristön osalta, vaikuttivat resurssien suunnitteluun ja va-

lintaan. Resursointiin vaikutti vahvasti myös saatu tieto siitä, millä aikataululla mahdolliset asiakkaat odottivat saavansa tuotteen käyttöönsä.

4 Uuden järjestelmän suunnittelu

Järjestelmän ensimmäinen visio tietorakennesuunnitelman sisällöstä valmistui maaliskuussa 2010. Suunnitelmassa oli otettu huomioon kaikki saadut kehitysiedat sekä muutamia itse suunniteltuja hyödyllisiä nyt tai tulevaisuudessa toteutettavia ominaisuuksia. Myynti- ja markkinointitiimi oli vahvasti mukana uutta järjestelmää suunniteltaessa, sillä tiimillä oli vahva näkemys edellisen version puutteista sekä tietysti saaduista kehitysideoista.

Fläppitaulu oli yksi merkittävä työkalu järjestelmän ensimmäisiä toiminnallisia suunnitelmia hahmoteltaessa sekä käyttöliittymän suunnittelussa. Kuvassa 1 on esimerkki ideoinnin tuloksesta.



Kuva 1: Fläppitaulu järjestelmän suunnittelun työkaluna.

Fläppitaulun käytön etuna on se, että ajatukset saadaan nopeasti esille ja ettei aikaa kulu turhaan erilaisten visuaalisten ohjelmistotyökalujen käyttöön. Lopulliset versiot taululle hahmotelluista suunnitelmista viimeisteltiin dokumentaatiota varten visuaalisilla työkaluilla.

Uuteen järjestelmäversioon toivotut ominaisuudet koostuivat lähinnä asiakkailta, mahdollisilta asiakkailta ja kehitystiimin jäseniltä tulleisiin kehitysehdoiksi. Osittain kehitysehdotukset olivatkin päällekkäisiä, tarkoittaen sitä, että kyseisille ominaisuuksille todella oli tarvetta. Kaikki kehitysehdotukset kirjattiin ja priorisoitiin vaatimusmäärittelydokumenttiin järjestelmän kehityskohteiksi.

Projektipäällikkö ryhtyi tekemään järjestelmän vaatimusmäärittelyä, joka sisältäisi kehityskohteet sekä kehityskohteiden ominaisuuksiin, järjestelmän kehitys- ja käyttöympäristöön sekä prioriteetteihin liittyvät tiedot. Vaatimusmäärittelydokumentti sisälsi tiedon markkinoinnillisista tavoitteista, järjestelmän toiminnallisista vaatimuksista, prioriteeteista, tulevista kehitysversioneista, referensseistä, työkaluista sekä teknisistä ohjereferensseistä sekä kehitysideoista.

Uuden sovelluksen suunnittelun merkittävänä lähtökohtana oli järjestelmän globaali käyttö sekä kansainvälisyys. Haasteeksi muodostui lokalisointiin liittyvät asiat, kuten päivämäärien muotoilu sekä järjestelmän mahdolliset käyttöliittymäkieleet, jotka tulisi ottaa hyvin huomioon järjestelmän suunnitteluvaiheessa. Tämä vaatimus edellytti harkintaa järjestelmässä käytettävien komponenttien sekä toiminnallisuuksien käytölle monessakin suhteessa. Esimerkkinä mainittakoon se, että tietokannan merkistöksi valittiin Unicode-merkistö, joka tukee kaikkia merkkejä. Nämä asiat vaikuttivat alusta saakka sovelluksen kehitykseen ja järjestelmän suunnitteluun.

Nomis VisitorCard -järjestelmän ensimmäinen prototyyppi valmistui kesäkuussa 2010. Prototyyppi ei vielä sisältänyt varsinaista toiminnallisuutta. Kuvassa 2 on esitetty vieraiden vastaanottopalvelun sovelluksen käyttöliittymän prototyyppi.

Tilannenäkymä - Vieraiden vastaanottopalvelun sovellus
 Nomis Vierailijajärjestelmä versio 2.0 (PROTOTYYPPI)

Käyttäjä: Matti Meikäläinen [Kirjaudu ulos](#)
 Kieli: Suomi [Muuta](#)

Perusvalinnat **Asetukset** Järjestelmänhallinta

Näytä tiedot päivämäärältä: **Tänään 14.6.2010** [Muuta](#)
 Näytä tiedot toimipisteestä: **Pasila, Helsinki** [Muuta](#)

Kokonaistilanne **Saapuvat vierailijat** Paikalla olevat vierailijat Käyneet vierailijat

Tänään 14.6.2010 toimipisteessä Pasila, Helsinki
 Vielä saapuvia vierailijoita: 27
 Tällä hetkellä paikalla olevia vierailijoita: 5
 Jo käyneitä vierailijoita: 13

Uuden vierailijan vastaanotto... Tulosta valitut avaamatta Avaa valitut vastaanotettaviksi...

Kuva 2: Nomis VisitorCard prototyyppi.

Prototyypin käyttö vaati, että Microsoft Silverlight liitännäinen oli asennettuna selaimen. Laadittu prototyyppi tarjosi jo valmiin pohjan mahdolliselle VB.NET/Silverlight -kehitysprojektille, eli prototyyppiin toteutetut näyttö- ja datarakenteet saataisiin hyödynnettyä sellaisinaan varsinaisessa kehitysprojektissa.

5 Käytetyt työvälineet ja menetelmät

5.1 Kehitystyökalut

Kehitystyökalut valittiin tarkoituksellisesti ja ensisijaisesti Microsoftin tarjoamista välineistä, koska Nomis Oy kuuluu Microsoft Gold Certified Partner -ohjelmaan. Kultatasolle päässeet yritykset ovat panostaneet asiantuntijoidensa koulutukseen sekä Microsoft-sertifiointeihin ja osoittaneet asiakasratkaisuillaan Microsoftin teknologian osaamisen. Gold Certified Partner -tason saavuttaminen on osoitus Nomis Oy:n toimittamien ratkaisujen saumattomasta yhteensopivuudesta Microsoft-ympäristön kanssa sekä Microsoftin tuotteiden osaamisesta ja onnistuneesta hyödyntämisestä haastavissa asiakasprojekteissa. Se antaa tukevan pohjan edistyksellisten ratkaisujen kehittämiseksi ja toimittamiseksi laajalle käyttäjäkunnalle sekä entistä paremmat mahdollisuudet vastata ohjelmistoille asetettuihin vaatimuksiin [1].

Sovelluskehittimeksi valittiin Microsoft Visual Studio .NET 2010 ja käytettäväksi ohjelmointikieleksi vaihtoehtoina oli joko Microsoft Visual Basic.NET tai vaihtoehtoisesti Microsoft Visual C#.NET. Järjestelmän ensimmäisessä protovaiheessa vastaanottosovellus toteutettiin käyttäen Microsoft Silverlight -tekniikkaa, joka kuitenkin myöhemmin korvattiin XHTML- sekä JavaScript-tekniikalla.

Työkalut valittiin sillä perusteella, että ne olisivat mahdollisimman uusia ja että niiden elinkaari olisi mahdollisimman pitkä kehityshetkestä eteenpäin. Valitun Microsoft Visual Studio 2010 Premium -tuotteen valmistajan tuki on vähintään 10 vuotta. Perustuki päättyy 14.7.2015 ja laajennetun tuen päättymispäivämäärä on 8.9.2020.

5.2 Tietokanta

Tietokannaksi valittiin ilmainen Microsoft SQL Server 2008 Express Edition with Advanced Services. Järjestelmän suunnittelussa päätettiin, ettei tietokantatukea anneta alkuvaiheessa muiden valmistajien järjestelmille, kuten Oraclelle. Tietokannan käytön pääperiaatteet ovat esitetty taulukossa 2.

Taulukko 2: Tietokannan suunnittelun pääperiaatteet.

Kohta	Tietokannan pääperiaatteet
1	Järjestelmän tietokanta tukee vain Microsoft SQL Server 2008 Express Edition ja sitä uudempia.
2	Kaikissa tauluissa tulee olla Primary Keynä (avaimena) GUID-tunniste.
3	Jos GUID-arvo on julkisesti näkyvillä sovelluksen käyttäjälle ja arvo on osa järjestelmän tietoturvainfrastruktuuria (esimerkiksi aktivointikoodi), tulee GUID-arvo generoida täysin satunnaisesti käyttäen NEWID()-metodia.
4	Jos tietokannan GUID-arvo on järjestelmän sisäinen arvo, mutta sitä käytetään hajautetussa järjestelmäkehityksessä (esimerkiksi LanguageData), taulun tunniste tulee generoida käyttäen NEWSEQUENTIALID()-metodia.
5	Järjestelmän tietokantataulujen tekstisarakkeet ovat määriteltä Unicode-yhteensopiville tietotyypeille (pääasiallisesti NVARCHAR, NTEXT).
6	Tietokannassa käytetyt nimet (taulut, toiminnot, ja niin edelleen) merkitään yksikkömuotoon. Esimerkiksi taulun nimi tulisi olla <i>Käyttäjä</i> eikä <i>Käyttäjät</i> .
7	Many-to-many -suhteessa olevat taulut ovat nimetty muotoon Table1_Table2, jossa, hallitseva taulu on vasemmalla (esimerkiksi User_UserRole). Tilanteissa jossa Table1_Table2-nimi muodostuisi liian pitkäksi, nimet voidaan lyhentää, jotta vältetään tarpeettoman pitkiltä nimiltä. Näin taulut nimetään esimerkiksi DocumentTemplate_Element, sen sijaan että käytettäisiin muotoa DocumentTemplate_DocumentTemplateElement.

Tietokantojen replikoinnissa käytetään Microsoft Sync Framework 2.0 -komponenttikirjastoa. Microsoft Sync Framework mahdollistaa vastaanoton sovelluksen offline-toiminnan sekä kantojen synkronoinnin päätietokannan kanssa sen jälkeen kun katkennut yhteys on jälleen palautunut.

5.3 Versionhallinta

Versionhallinnassa käytettäväksi työkaluksi oli useita eri vaihtoehtoja. Valinnassa tärkeäksi ominaisuudeksi osoittautui lähdekoodin säilytys, sillä sovelluskehittäjät tekivät osan työstään muualla kuin toimistolla yhteisen verkon sisällä. Tarvittiin siis uusi työkalu, joka mahdollisti yksittäisten lähdekoodien kopion ja yhden yhteisen päälähdekoodin, johon kopioiden uusi ohjelmakoodiosuus lisättäisiin.

Versionhallintatyökalun valintaa helpotti Wikipediasta [2] löytynyt erinomainen vertailusivu, josta löytyi kaikki tarpeellinen tieto työkalun valitsemiseksi.

darcs [16] 	actively developed	GPL	Unix-like, Windows, Mac OS X
Fossil [17] 	actively developed	BSD	POSIX, Windows, Mac OS X, Other
Git [18] 	actively developed	GPL	POSIX, Windows, Mac OS X
GNU arch [19] 	maintained but new features not added	GPL	Unix-like, Windows, Mac OS X
Mercurial [23] 	actively developed	GPL	Unix-like, Windows, Mac OS X
Monotone [25] 	actively developed	GPL	Unix-like, Windows, Mac OS X
SVK [32] 	maintenance through 2010, no new features 	Artistic/GPL	Unix-like, Windows, Mac OS X
Bazaar [7] 	actively developed	GPL	Unix-like, Windows, Mac OS X
Software	Development status	License	Platforms supported

Kuva 3: Soveltuvat versionhallintatyökalut.

Kuvassa 3 esitellyistä useista vaihtoehdoista vain kolmessa osoittautui olevan jonkinasteinen tuki Visual Studion integroidulle ympäristölle (IDE), Bazaar, Git ja Mercurial, joista Mercurialissa on paras integroidun ympäristön tuki. Lisäksi Mercurialin etuna on se, että versionhallinta tehdään muutoserien kautta nimettyjen versiokokonaisuuksien sijaan.

5.4 Kehitystiimi

Kehitystiimi koostui neljästä sovelluskehittäjästä, projektipäälliköstä sekä kahdesta myynti- ja markkinointitiimin jäsenestä. Kaikki sovelluskehittäjät, projektipäällikkö ja yksi myynnistä ja markkinoinnista vastaava henkilö työskentelivät Nomis Oy:n tytäryrityksessä Manilassa, Filippiineillä. Suomessa kehitystiimiin laskettiin kuuluvaksi vain yksi henkilö. Syy, miksi sovelluskehitys tehtiin pääasiallisesti Filippiineiltä käsin, ovat projektille annettu aikataulu ja työvoimaan liittyvät kustannukset. Työvoimakustannukset ovat Filippiineillä erittäin alhaiset verrattuna Suomessa vastaavan tiimin kustannuksiin.

Kehityksen seurannassa toteutettiin muunnelmaa niin sanotusta SCRUM-menetelmästä, jossa jokainen sovelluskehitystiimin jäsen osallistui projektiin itseohjautuvasti ja kehitettävää ohjelmistotuotetta rakennettiin hiljalleen valmiimmaksi. Kehityskohteet jaettiin työlistalle osiin siten, että kunkin osan odotettiin tuottavan vaatimusmäärittelyn mukaiset tulokset. Jokainen tiimin jäsen teki päivittäin yhteenvedon koko tiimille, joko suullisesti tai sähköpostitse päivän päätteeksi riippuen siitä, oliko henkilö toimistolla vai ei. Yhteenvedossa kukin kertoi, mitä oli tehnyt viimeisimmän yhteenvedon jälkeen, mitä tulee seuraavaksi tekemään sekä raportoi mahdollisista ongelmista kehitykseen liittyen. Säännöllinen raportointi helpotti ajoittamaan ja jakamaan töitä tekijöiden kesken sekä mahdollisti tiedon ja taidon välittämisen muille tiimin jäsenille.

Töiden seurannan ja jakamisen helpottamiseksi käytimme PERT-kaaviota, jossa toteutettavat sovelluskohteet olivat aikataulutettuina ja jaettuna vastaaville sovelluskehittäjille taulukon 3 mallin mukaisesti.

Taulukko 3: PERT-tiliö.

Aloitusaika aikaisintaan	Arvioitu kesto	Valmis aikaisintaan
Tehtävän nimi ja suorittaja(t)		
Aloitusaika viimeistään	Sallittu viive	Valmis viimeistään

Erilliset sovelluskohteet kirjattiin edellisen esimerkin tavoin ja yhdistettiin kronologisessa järjestyksessä siten, että viimeisessä vaiheessa tehtävänä oli ainoastaan valmiin sovelluksen asennuspakettien teko. PERT-kaavion käyttö todettiin hyväksi, koska kehitettävät sovelluskohteet olivat toisistaan irrallisia. Kaavion avulla pystyttiin ennustamaan koko projektin kestoa sekä havaitsemaan mahdolliset hidastavat tekijät ajoissa. Kaavion tarkoituksena oli myös havainnollistaa sovelluskohteiden työvaiheiden järjestys ja riippuvuussuhteet.

5.5 Testaus

Nomis VisitorCard -järjestelmän testaus toteutettiin osana kehitysprojektia siten, että kehitystiimin jäsenet suorittivat testauksen itse omien kehityskohteidensa osalta. Projektipäällikkö testasi kokonaisuutta ja antoi kommenttinsa sekä kehitysehdotuksensa sovelluskehittäjille. Testauksen laajuus riippui pitkälti projektin etenemisestä sekä sovelluksen valmiusasteesta. Moduuli- ja integraatiotestauksen tuloksia varten ei varsinaisesti pidetty testauspöytäkirjaa, vaan havaitut virheet korjattiin iteroimalla. Järjestelmätestaus tullaan tekemään systemaattisesti järjestelmän kuvausdokumenttien mukaan ja järjestelmätestauksesta laaditaan myös testauspöytäkirja.

6 Nomis Visitor Card -järjestelmän tekninen kuvaus

6.1 Käyttöliittymät

Nomis VisitorCard -järjestelmässä on neljä käyttöliittymää eri toimintoja varten, kutsujalle, osallistujalle, vastaanotolle ja vastaanoton itsepalvelulle omansa. Kaikki käyttöliittymät ovat yhteydessä pää tietokantaan, jonne tietosisältö tallennetaan. Järjestelmä on kieliparametroitu, mikä tarkoittaa, että käyttöliittymät voidaan kääntää lähes mille kielelle tahansa.

Kutsujan ja osallistujan sovellus vaatii työasemalta toimiakseen Internet-yhteyden sekä selaimen. Kutsujan ja osallistujan sovellus käyttää XHTML-tekniikkaa sekä JavaScript-kieltä. Kehityksessä käytetään AJAX-kirjastoja jotka, kuuluvat JavaScript-tekniikkaan.

Vastaanoton sovellus on perinteinen Windows-sovellus, joka vaatii työaseman, jonka käyttöjärjestelmäksi on asennettu Windows XP SP3 tai uudempi ja jossa on paikallisesti asennettu SQL Server 2008 Express.

6.2 Tulostinvaatimukset

Järjestelmä asettaa vierailijakorttitulostimelle tiettyjä vaatimuksia, esimerkiksi tulosteen koon. Nomis VisitorCard -järjestelmän kehityksessä päätettiin ottaa huomioon vain lämpötulostimet, jotka tulostavat lämpöpaperille. Lämpötulostimien haittapuolena on se, että lopputulos on harmaasävyinen tai musta-valkoinen. Vastaavien väritulostimien hinta oli moninkertainen verrattuna lämpötulostimeen. Lämpötulostinpaperia on saatavilla kuitenkin esipainettuna, jolloin mahdolliset värilogot tai muut kiinteät kohteet voidaan painattaa etukäteen paperille. Tulostimien valinta tuli tehdä näitä ehtoja noudattaen. Tulostimien valintaan vaikutti myös laitteiden saatavuus maantieteelliset seikat huomioon ottaen sekä laitteen mahdollisimman alhainen hinta. NDA-sopimuksen tulostamiseen suunniteltiin soveltuvaksi mikä tahansa oheistulostin.

Taulukot 4, 5, ja 6 kuvaavat erityyppisten tulostinratkaisuiden hyviä ja huonoja puolia.

Taulukko 4: Lämpötulostimet.

Hyödyt	Haitat
<p>+ Lämpötulostin ei käytä muste- tai värikasetteja.</p> <p>+ Valmiiksi muotoon leikatut paperiarakit tai tarrapaperin leikkuri. Vierailijakortit ovat tulostuksen jälkeen heti käytettävissä.</p> <p>+ Tulostuskustannukset ovat lähempänä tavallista tulostamista, kuin PVC:lle tulostettaessa.</p> <p>+ Tulostimet tukevat viivakoodin tulostusta.</p> <p>+ Lämpötulostimet ovat yleisimmin käytetty ratkaisu väliaikaisien vierailijakorttien tulostuksessa.</p> <p>+ Vierailijakortin tulostus on nopeaa.</p>	<p>- Ei ole mahdollista tulostaa väreillä, esimerkiksi vierailijan kuvaa tai yrityksen logoa.</p>

Taulukko 5: Muovikorttitulostimet.

Hyödyt	Haitat
<p>+ Laadukkaimman näköinen kortti ja tulostusjälki.</p> <p>+ Mahdollisuus väritulostukselle.</p> <p>+ Kestävät kortit sopivat esimerkiksi pitkäaikaisille vierailijoille.</p> <p>+ Olisi suositeltu vaihtoehto, jos kullekin vieraalle annettaisiin vain yksi kortti (mukana kuljetettavaksi).</p>	<p>- Muovikorttitulostin on kaikista kallein vaihtoehto (sekä tulostin-, että tulostuskustannuksilta).</p> <p>- Muovikortteja on yleisesti saatavilla 85 mm x 54 mm kokoisena; kortin koko mahdollistaa vain vähäisen tiedon tulostamisen kortille.</p> <p>- Kortin tulostaminen voi viedä kauan, väritulostus 30 sekuntia ja harmaasävytulostus 6 sekuntia.</p>

Taulukko 6: Laser- ja mustesuihkutulostimet.

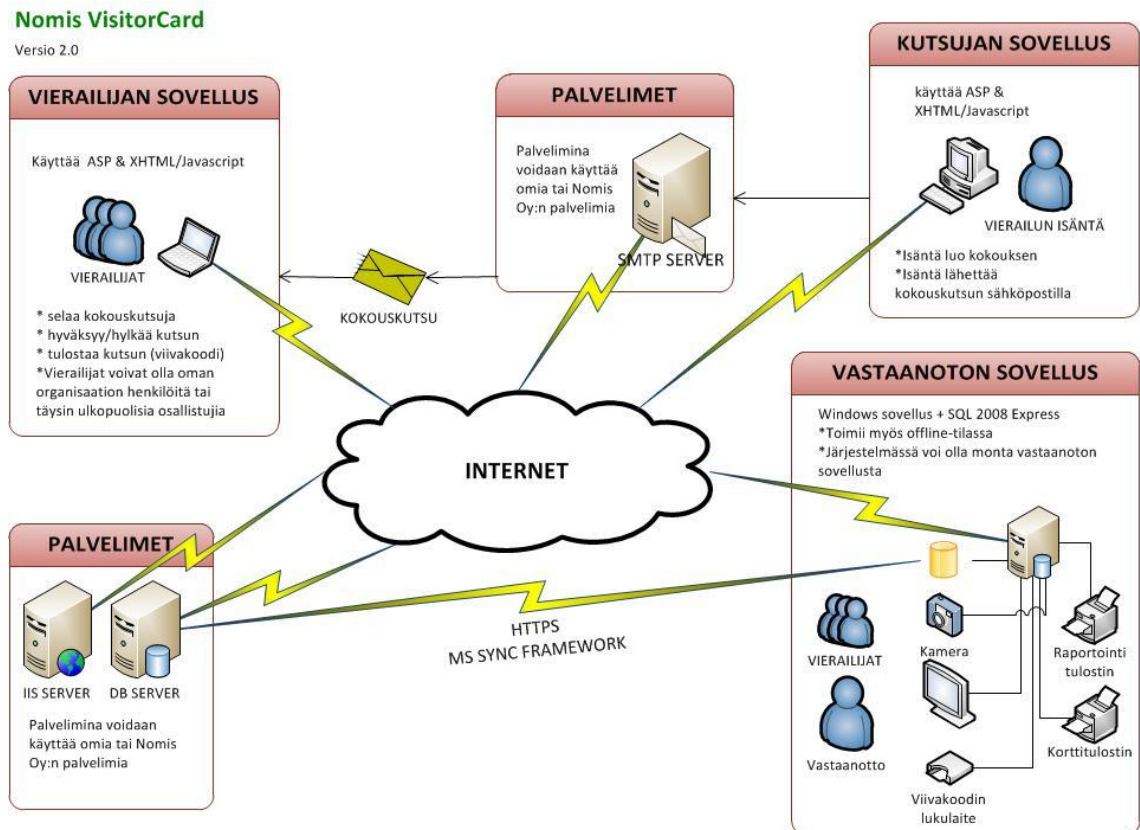
Hyödyt	Haitat
<ul style="list-style-type: none"> + Halvin ja monikäyttöisin tulostin. + Mahdollisuus väritulostukseen. + Tulostuksessa voidaan käyttää eri paperilaatua. + Käytettäessä tavallista valokuvapaperia vierailijakortin laatu olisi paras mahdollinen. + Tavallinen A4-kirjepaperin tulostus saattaisi olla kustannussäästöjen puolesta paras vaihtoehto Filippiineillä. + Tavallista mustesuihkutulostinta voidaan käyttää myös soveltuvasti pinnoitettujen muovikorttien tulostamiseen. 	<ul style="list-style-type: none"> - 4 x 6 -tuumaiselle valokuvapaperille mahtuu kaksi 3 x 4 tuuman vierailijakorttia, joten paperi pitäisi leikata tai taittaa. A4-kirjepaperin kokoisesta pitäisi leikata tai taitella vielä enemmän. Hidasta ja soveltumatonta vieraiden vastaanotossa. - Viivakoodifontit on asennettava työasemalle aina erikseen.

Tehdyn vertailun perusteella saatavilla olevista vaihtoehdoista lämpötulostustekniikkaan perustuva tulostin soveltuu parhaiten vierailijakorttien tulostukseen. Nomis VisitorCard -järjestelmän julkaisuversiossa tuettavaksi tulostimeksi valittiin Brotherin valmistama QL-570-mallin tulostin. Tulostimen maailmanlaajuinen saatavuus oli suurin tekijä tämän mallin valitsemisessa.

6.3 Arkkitehtuuri

Järjestelmäkaavio

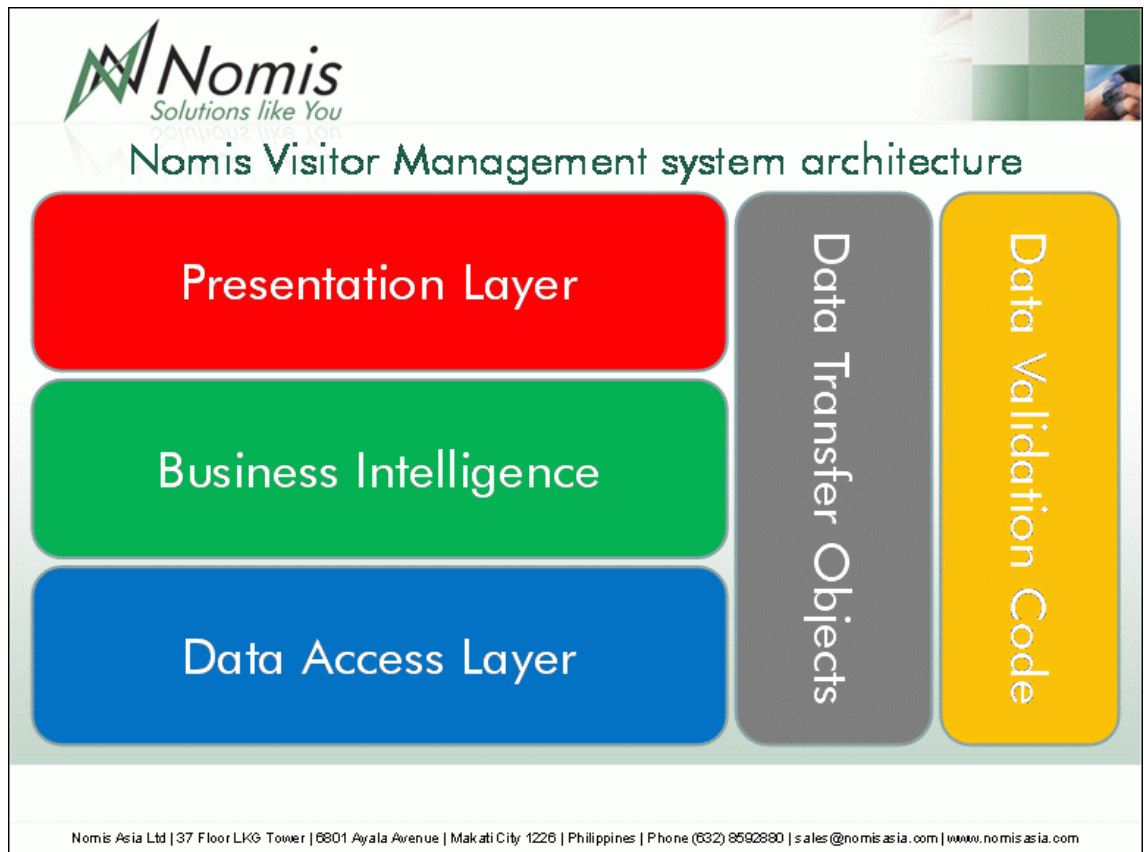
Järjestelmän toiminnallinen kaavio on esitetty kuvassa 4.



Kuva 4: Nomis Visitor Card -järjestelmäkaavio.

Kerrosarkkitehtuuri

Nomis VisitorCard -järjestelmän kerrosarkkitehtuuri toteutettiin 3-tier-logiikalla, joka tarkoittaa, että järjestelmän toiminnot ovat jaettuna kolmelle eri tasolle. Kuvassa 5 nähdään, miten tasot ovat jaettuna *Presentation Layer*-, *Business Intelligence*- ja *Data Access Layer*-tasoihin. Lisäksi on kaksi poikittaista tasoa, jotka toimivat kaikilla kolmella tasolla, *Data Transfer Object* sekä *Validation Code*.



Kuva 5: Nomis VisitorCard -kerrosarkkitehtuuri.

Presentation Layer vastaa sovelluksen käyttäjälle näkyvien toimintojen toteuttamisesta. Kyseisellä tasolla toimii järjestelmän käyttöliittymät joiden tehtävänä on esittää käyttäjälle olennaista tietoa sekä välittää eteenpäin käyttäjän antamat syötteet ja kutsut.

Business Intelligence eli niin kutsuttu suorituskerros tekee tarvittavat toimenpiteet esityskerroksen ja datakerroksen välillä. Suorituskerroksessa toteutetaan suurin osa sovelluksen loogisesta toiminnasta. Tämä kerros yhdistää kaksi muuta kerrosta.

Data Access Layer -kerroksessa tapahtuu tiedon välitys tietokantaan ja tietokannasta. Haku perustuu suorituskerrokselta saatuun käyttäjän antamaan syötteeseen. Haettu tieto palautetaan välikerroksen kautta takaisin suorituskerrokselle.

Data Transfer Objects sisältää tiedonsiirtoon liittyvät objektit.

Data Validation Code käsittää tiedon validointiin liittyvät toimenpiteet.

Nomis VisitorCard -järjestelmän kutsujan, osallistujan sekä vastaanoton sovellukset käyttävät yhteistä DLL-kirjastoa. DLL-kirjasto sisältää kaikki muut edellä mainitut osat paitsi Presentation Layerin.

6.4 Toimintaympäristö

Nomis VisitorCard -järjestelmän kutsujan sekä osallistujan sovellusten käyttöliittymät toimivat kaikilla vuonna 2010 nykyaikaiseksi luokitelluilla Internet-selaimilla. Vastaanottajan sovellus toimii Windows XP SP3- tai uudemmassa Windows -käyttöjärjestelmässä ja käyttää omaa paikallista Microsoft SQL Server Express -tietokantaa. Paikallinen tietokanta replikoi tietokantapalvelimen päätietokannan kesken.

Palvelimet

Järjestelmä asennetaan Windows-palvelimelle, esimerkiksi Windows 2008 Serveriin. Tietokanta voidaan asentaa samalle palvelimelle, ja se voi olla joko lisensoitu Windows SQL Server tai ilmainen Windows SQL Express Server. Lisäksi käyttöliittymäsivustojen hallinta vaatii IIS-palveluiden konfiguroinnin Windows-palvelimelle.

Toimitusmallit

Järjestelmän toimitusmalleina on vaihtoehtoisesti projektitoimitus tai SaaS-toimitus. Projektitoimitusmallissa järjestelmä asennetaan ja konfiguroidaan asiakkaan omalle palvelimelle, kun taas SaaS-toimitusmallissa järjestelmää käytetään Nomis Oy:n palvelimelta. Kummassakin toimitusmallissa vastaanoton sovellus asennetaan aina vastaanottopisteen työasemalle. Asennettu vastaanoton sovellus integroidaan SaaS- tai projektipalvelimelle asennettuun järjestelmään.

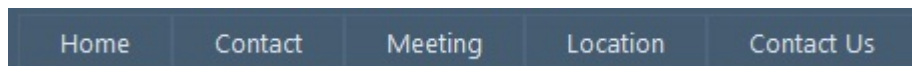
7 Nomis VisitorCard -järjestelmän toiminnallinen kuvaus

7.1 Kutsujan sovellus

Kutsujan sovelluksella tarkoitetaan selaimella toimivaa käyttöliittymää, jonka avulla tapaamisen järjestäjä voi hallita tapaamisia, tapaamiseen osallistuvien vieraiden tietoja, tapaamiseen liittyviä sijaintitietoja sekä lähettää tapaamiskutsuja sähköpostitse.

Perusnäkyvä

Kuvassa 6 näkyvät perusnäkyvän navigointilinkit ovat selkeät ja helpottavat käyttäjän siirtymistä eri osioiden välillä.



Kuva 6: Kutsujan sovelluksen navigointipalkki.

Päänäkymä (Home) palauttaa käyttäjän takaisin perusnäkyvään.

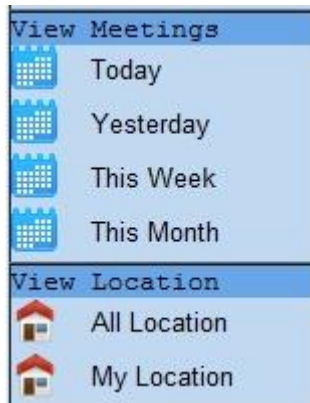
Yhteystiedot-osiossa (Contact) käyttäjä voi lisätä, muokata ja poistaa omia kontaktitietojaan.

Tapaamiset-osiossa (Meeting) käyttäjä voi lisätä, muokata ja poistaa omia tapaamisiaan. Edellisten toimintojen lisäksi käyttäjä pystyy tarkastelemaan tapaamisiin kutsumiaan osallistujia ja kutsumaan lisää osallistujia valitsemaansa tapaamiseen.

Sijainnit-osio (Location) tarjoaa käyttäjälle toiminnot erilaisten sijaintitietojen luomiseen. Sijainteja voivat olla esimerkiksi huoneen numerot ja neuvotteluhuoneiden tiedot. Sijainneille voidaan antaa lisätiedoksi organisaation nimi- ja osoitetietoja.

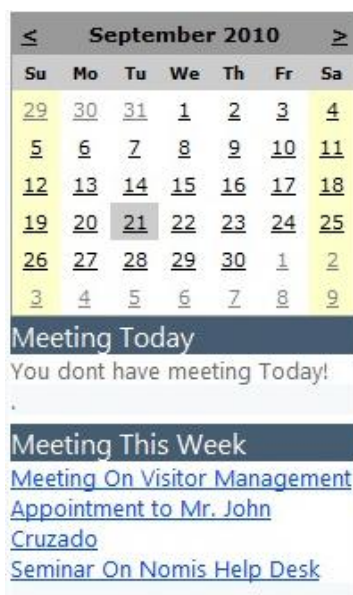
Ota yhteyttä -osio (Contact Us) on tarkoitettu organisaation omille teknisille yhteystiedoille.

Kutsujan sovelluksessa käyttäjä voi selata itse tekemiään tapaamismerkintöjä käyttöliittymän vasemman reunan pikatoiminnoilla. Kuvassa 7 nähdään tarkemmin pikatoiminnot, joilla käyttäjä voi valita tapaamisia ajankohdan mukaan, esimerkiksi "tänään", "eilen", "tällä viikolla" tai "tässä kuussa". Käyttäjä voi lisäksi valita myös eri sijainteihin merkitsemiään tapaamisia.



Kuva 7: Kutsujan sovelluksen pikatoiminnot.

Kutsujan sovelluksen käyttöliittymän oikeassa reunassa on kuvan 8 mukainen kalenterinäkö, josta käyttäjä voi valita tarkasteltavakseen haluamansa päivämäärän tapaamiset. Lisäksi oikeassa reunassa näkyy ajankohtainen tilanne kuluvan päivän sekä kuluvan viikon tapaamisista.



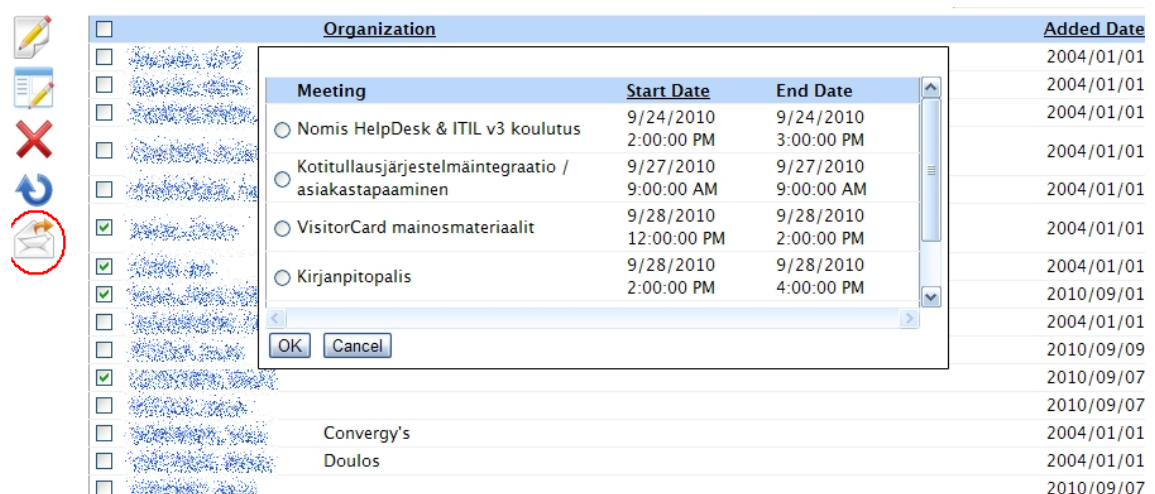
Kuva 8: Kalenterivalinta sekä lähitulevaisuuden tapaamiset.

Kalenterinäkömä, kuluvan päivän sekä viikon tapaamiset ovat aina näkyvissä sovelluksen oikeassa reunassa riippumatta siitä, mitä osiota käyttäjä tarkastelee.

Tapaamiset

Tapaamisiosiossa käyttäjä voi luoda uusia tapaamisia, muokata ja poistaa olemassa olevia tapaamisia. Tapaamiselle annetaan tiedot tapaamisen aiheesta, sijainnista, aloitus- ja lopetusajasta sekä saateteksti. Saatetekstiä on mahdollista korostaa erilaisin tekstinkäsittelytoiminnoin ja painikkein, jotka ovat yleisesti käytettyjä ja tunnistettavia. Toiminnot ovat aseteltuna saatetekstikentän yläpuolelle. Tallenna-napin painaminen tallentaa tapaamisen tiedot yhteiseen tietokantaan. Tallennuksen jälkeen tapaaminen näkyy myös vastaanoton sovelluksessa.

Kutsuttavien osallistujien lisääminen tapaamiselle tapahtuu siten, että käyttäjä valitsee osallistujalistalta haluamansa osallistujat, jonka jälkeen painaa kutsunlähetyspainiketta (kuvassa 9 merkittynä punaisella renkaalla). Lähetyspainike avaa listan käyttäjän luomista tapaamisista, joista haluttu tapaaminen valitaan. Tapaamiskutsu lähetetään osallistujille käyttäjän painaessa OK-painiketta.



Kuva 9: Tapaamisen osallistujien valintänäyttö.

Lähetettävän sähköpostin liitteenä vastaanottaja saa tallennettavan tapaamismerkinnän. Merkintä lähetetään formaatissa, joka sisältää tapaamiseen liittyvät tiedot aikatauluista ja muista tapaamiseen liittyvistä tiedoista.

7.2 Osallistujan sovellus

Osallistujan sovellus on tarkoitettu antamaan lisätietoa tapaamisesta sekä mahdollistamaan vierailijakorttien tulostuksen. Osallistujan sovelluksessa nähdään kaikki tapaamiskutsut, joihin vierailija on kutsuttu. Osallistujan sovelluksen kautta käyttäjä voi muokata omia yhteystietojaan (Kuva 10), jotka tallentuvat Nomis VisitorCard-järjestelmän tietokantaan.

Personal Information

First Name	<input type="text" value="John"/>
Middle Name	<input type="text" value="Rabang"/>
Initial	<input type="text"/>
Last Name	<input type="text" value="Cruzado"/>
Extension Name	<input type="text"/>
Email	<input type="text"/>
Job Title	<input type="text"/>
Education Title	<input type="text"/>
Additional Info	<input type="text"/>

Kuva 10: Osallistujan sovelluksen yhteystietosivu.

Tapaamisen osallistuja voi hyväksyä sekä peruuttaa osallistumisensa tapaamiseen sivuston kautta sekä halutessaan tulostaa itselleen valmiiksi viivakoodilla varustetun vierailijakortin esimerkiksi sellaisiin tapahtumiin, joissa ei ole erikseen vastaanottohenkilöä tulostamassa osallistujien vierailijakortteja.

7.3 Vastaanoton sovellus

Vastaanoton sovellus on yleisesti tarkoitettu organisaatioiden aula- tai vastaanottohenkilökunnan käyttöön. Sovelluksella hallitaan vierailijoiden vastaanottoon liittyvät toimenpiteet, kuten vierailijakorttien ja NDA-sopimusten tulostaminen. Vastaanoton sovelluksessa voidaan ylläpitää järjestelmään liittyviä yleisiä tietovaraston tietoja. Vastaanoton sovellus mahdollistaa manuaalisesti tehtävän vierailijan vastaanottamisen ja tulostustoiminnot myös ilman kutsujan tekemää tapaamisilmoitusta. Vastaanoton sovellus mahdollistaa vierailijoiden valokuvauksen, jolloin kuvaa voidaan käyttää tulostettavassa vierailijakortissa.

Perusnäköymä

Perusnäköymä sisältää vierailijoihin ja tapaamisiin liittyvät tiedot. Sovelluksen perusnäköymässä voidaan hakea halutun päivämäärän, aikavälin sekä valittujen toimipisteiden tapaamistietoja ja vierailijatietoja. Vieraita voidaan seurata myös tilatiedon perusteella, kuten saapuvat, paikalla olevat ja poistuneet.

Tapaamisia voidaan selata tapaamiset-välilehdellä (Meetings). Tapaamiseen liittyvät perustiedot, kuten aloitus- ja lopetusaika, tapaamisen järjestäjä sekä sijaintitiedot, ovat luettavissa listausnäytössä. Tapaamiseen osallistuvat vieraat saadaan esille valitsemalla tarkasteltava tapaaminen listalta.

Järjestelmän tietovarasto

Järjestelmän tietovaraston ylläpito hallitaan vastaanoton sovelluksessa. Ylläpidettäviä asioita ovat vierailija-, toimipiste-, sijaintitiedot sekä vieraiden, sijaintien ja tapaamisten luokittelu.

Vierailijakortti ja NDA-sopimus

Vastaanoton sovelluksesta vierailijoille voidaan tulostaa henkilökohtainen vierailijakortti, jonka vieras voi kiinnittää näkyvälle paikalle. Vierailijakortin tiedot on kiinteästi sijoitet-

tu sovelluksen ensimmäisessä 2.0 -versiossa, eikä sisältöä tietojen sijainnin osalta voi muokata. Kuvassa 11 nähdään esimerkki tulostetusta vierailijakortista.



Kuva 11: Esimerkki tulostettavasta vierailijakortista.

Vierailijakortin (kuva 11) osat ovat:

1. kutsuvan organisaation nimi
2. osallistujan nimi ja organisaatio
3. suunnitellun tapaamisen ajankohta sekä isännän nimi (kutsuja)
4. lisätieto tapaamisen sijainnista
5. vierailijan tiedot viivakoodina
6. kuva vierailijasta
7. järjestelmän luoma turvatunniste sekä vierailijakortin tulostusaika ja paikka (järjestelmän luoma turvatunniste mahdollistaa väärennettyjen vierailijakorttien tunnistamisen).

8 Jatkokehitys

Vaatimuksien uudelleen määrittely

Nomis VisitorCard -järjestelmän version 2.0 kehityksen edellä tehty vaatimusmäärittely sisältää ominaisuuksia, jotka on suunniteltu tehtäväksi järjestelmän tulevissa kehitysversioissa. Vaatimusmäärittelyä tarkennetaan ennen jokaisen uuden version kehityksen aloittamista ja soveltuvat kehityskohteet valitaan.

Versiointi

Uuden järjestelmän versionumerointi alkoi versionumerolla 2.0. Tällä haluttiin antaa asiakkaille selkeä vaikutelma version kehityksessä tapahtuneista muutoksista. Versiointi suunniteltiin jatkuvan juoksevasti kaksinumeroisena. Taulukossa 7 on esitetty versiosuunnitelma seuraavalle kahdelle julkaistavalle versiolle.

Taulukko 7. Versiosuunnitelma.

Versio	Suunniteltu julkaisupäivämäärä	Version kuvaus
2.0	30.11.2010	Ensimmäinen julkaisuversio
2.1	xx.2.2011	Lisäominaisuuksien julkaisuversio
2.2	xx.yy.2011	Lisäominaisuuksien julkaisuversio
...		

Liittymät

Nomis VisitorCard -järjestelmän ulkoiset liittymät muihin tietojärjestelmiin tullaan toteuttamaan järjestelmän tulevissa versioissa. Tällaisia liittymiä ovat esimerkiksi liittymät Microsoft Active Directory -aktiivihakemistoon sekä Microsoft Exchange -sähköposti-järjestelmän osoitekirjoihin ja kalentereihin.

9 Markkinointi

Nomis Oy:n myynti- ja markkinointitiimi toimi projektin alusta lähtien tiiviisti yhteistyössä sovelluskehitystiimin kanssa. Markkinointimateriaalia syntyi samanaikaisesti kehityksen edetessä.

Markkinoinnin tavoitteena oli saada esittelykelpoista materiaalia käyttöön jo ennen varsinaisen julkaisuversion valmistumista. Tavoitteet saavutettiin dokumentaarisessa muodossa, mutta esimerkiksi järjestelmän esittelyversio jäi toteutettavaksi vasta järjestelmän julkaisun yhteydessä.

10 Tulokset

Taustatyön tulokset

Taustatyön tuloksena löydettiin kolme ohjelmistotuotetta, joista löytyi vastaavuuksia suunnitellulle sovellukselle. Taustakartoituksessa olisi ollut vielä paljon tehtävää, mutta aikatauluvaateen sekä resurssien puutteen takia kartoitusta ei pystytty viemään niin pitkälle kuin olisi haluttu. Laajempi taustatyö olisi todennäköisesti antanut paremman kuvan olemassa olevista järjestelmistä, niiden ominaisuuksista sekä mahdollisista puutteista. Kattavamman taustatyön tuloksena olisi saatu tärkeää lisätietoa vaatimusmäärittelyn suhteen.

Vaatimusten toteutuminen

Lokakuussa 2010 verrattiin sovelluksen kehitysversion ominaisuuksia vaatimusmäärittelyssä esitettyihin kehityskohteisiin. Vertailussa voitiin todeta, että 75 % kaikista vaatimusmäärittelyn asiakaskohtaisista kehityskohteista oli täysin toteutettu ja testattu. Asiakaskohtaisia kehityskohteita oli kaiken kaikkiaan 24 kappaletta, joista siis 6 jäi toteutumatta. Toteutumatta jääneet kehityskohteet siirrettiin seuraavaan versioon toteutettavaksi. Siirretyt kohteet liittyivät lähinnä muihin tietojärjestelmiin toteutettavista integraatioista.

Aikataulu

Projektin suunniteltu aikataulu ei ollut kovin kriittinen, joten aikataulussa pystyttiin hyvin joustamaan. Joustavuuden vuoksi suunniteltua aikataulua eikä toteutunutta kirjattu tarkasti. Nomis VisitorCard -järjestelmän suunniteltu esittelyversio oli tarkoitus saada esiteltäväksi syyskuun loppuun mennessä Nomis Oy:n asiakaspäivillä. Esittelyversio ei ehtinyt valmistua tavoiteaikana, joten esittelyssä hyödynnettiin kehitteillä olevasta järjestelmästä otettuja kuvankaappauksia. Nomis VisitorCard -järjestelmän julkaisuversio 2.0 valmistuu kuitenkin suunnitellusti ajallaan marraskuussa 2010.

Työmenetelmät

Projektin etenemisen suunnittelussa käytettyä PERT-menetelmää ei pystytty hyödyntämään halutulla tavalla projektin kiireellisen aikataulun vuoksi. Toteutettavat työt olisi pitänyt pilkkoa vielä pienempiin ja tarkempiin palasiin, jotta töiden työmäärän ennustaminen olisi ollut järkevästi ennustettavissa. Pienempiin palasiin pilkkominen puolestaan olisi vaatinut suunnitelman laatimiseksi lisää työaika, joka kuitenkin haluttiin ennemminkin käyttää sovelluksen kehittämiseen.

Järjestelmän testasivat sovelluskehittäjät sekä projektipäällikkö. Testauksessa keskityttiin toiminnallisuuksien testaamiseen eikä kovinkaan paljoa käyttöliittymien laajempaan testaukseen. Tavoitteena oli testauttaa järjestelmän käyttöliittymät toimistohotellin henkilökunnalla, mutta toteutus lykkääntyi vasta seuraavan version yhteydessä tehtäväksi.

Työkalut

Työkaluiksi pyrittiin valitsemaan mahdollisimman uusia kehitystyökaluja, joita markkinoilla vuonna 2010 löytyi. Uusien työkalujen valinta mahdollistaa ensinnäkin pitkän jatkokehityksen samoilla välineillä ja samalla mahdollisimman pitkän tuen valmistajan puolelta. Haittapuolena toisaalta on se, että uusien työvälineiden toimintojen ja menetelmien opettelu tuhlaa aikaa kehitykseltä.

Dokumentointi

Projektin kehityksen yhteydessä käytetty vaatimusmäärittelydokumentti oli merkittävin sovelluksen ominaisuuksien toteutumisen kannalta. Vaatimusmäärittelydokumenttia päivitettiin säännöllisesti, ja se tulee toimimaan myös seuraavien versioiden kehityksen suunnittelussa. Sovelluskehitysprojektissa ei tehty varsinaista projektisuunnitelmaa, mutta roolijaot sekä aikataulurajat käytiin suullisesti läpi kehitystiimin kesken, mitä tuki PERT-kaavio.

Projektin lopputulos

Projektin lopputuloksena syntyi sovellus, jonka toiminnot vastasivat vaatimusmäärittelydokumentissa kirjattuja sovelluskehityskohteita. Sovelluksen käyttöliittymät ovat visuaalisesti selkeitä ja käyttäjän kannalta helposti opittavissa. Sovelluksen osittainen seläinpohjaisuus mahdollistaa järjestelmän käytön erilaisissa käyttöympäristöissä.

Oman osuuden arviointi

Oma osuuteni projektin osalta kohdistui enimmäkseen taustatietojen kartoitukseen, järjestelmän suunnitteluun ja markkinointiin. Seuraavaksi puran auki tarkemmin omaa osuuttani tuotekehitysprojektissa.

Taustatietojen kartoitukseen käytin kumulatiivisesti laskien yhden viikon. Markkinoilla olevien vastaavien tuotteiden selvityksessä internet oli päätietolähde, mutta sain myös tietoa kilpailutilanteesta erilaisissa tietotekniikka-alan tapahtumissa. Suurin osa käytetystä ajasta kului kuitenkin asiakastapaamisissa, joissa keräsin mahdollisimman paljon tietoa asiakkaiden toiveista ohjelmiston toiminnallisuutta koskien.

Järjestelmän suunnittelussa oma työpanokseni astui esiin siinä vaiheessa, kun ryhdyttiin luomaan kokonaiskuvaa sovelluksen eri osioiden toiminnallisuuksista. Toiminnallisuuden suunnittelussa toin esille asiakkaalta keräämääni arvokasta tietoa vaadituista ominaisuuksista. Kehityksen edetessä toimin tiedon välittäjänä asiakkaan ja sovelluskehitystiimin välissä ja annoin välittömän palautteen asiakkaalta saamani kommenttien perusteella.

Osallisuuteni järjestelmän suunnittelussa ja taustakartoituksessa antoi valmiit työvälineet ja tiedot sovelluksen markkinoinnin edistämiseksi. Vaikka järjestelmä itsessään oli vielä pieniin palasiin lohkottuna, pystyin tekemään tietojeni pohjalta riittävän markkinointimateriaalin myynnin ja markkinoinnin käynnistämiseksi.

11 Päätelmä

Tässä insinööriyössä käsiteltiin Nomis Oy:n toteuttamaa Nomis VisitorCard -sovelluksen uudelleenkehitysprojektin vaiheita. Työssä esiteltiin taustakartoitukseen liittyviä toimenpiteitä, kuten markkinoilla olemassa olevat vastaavat järjestelmät, tarvekartoitus sekä kehitysideat. Kartoituksen tuloksena päätettiin käynnistää Nomis VisitorCard -järjestelmän 2.0-version sovelluskehitysprojekti. Sovelluksen aikaisemman version haasteellinen sovelluskehitysympäristö vaikutti siihen, että uusi versio päätettiin aloittaa puhtaalta pöydältä ja täysin toisen tyyppisillä sovelluskehitystyökaluilla. Uuden version toiminnan pääpiirteet perustuivat kuitenkin vahvasti vanhan järjestelmän toimintoihin.

Projektin sovelluskehitys toteutettiin Nomis Oy:n tytäryrityksessä Manilassa, Filippiineillä. Suomesta käsin projektiin osallistui Suomessa tapahtuvasta myynnistä ja markkinoinnista vastaava. Tiimin työskentely oli varsin itseohjautuvaa, mutta työtilanne raportoitiin ja kehityksen edistymistä seurattiin päivittäin. Sovelluskehittäjät testasivat sovelluksen yksittäiset osat koodaa ja korjaa -menetelmällä. Projektipäällikkö vastasi kehitys- ja muutostehtävien jakamisesta sovelluskehittäjille sekä kokonaisuuden testaamisesta.

Sovellus saatetaan julkaisuvalmiiksi marraskuussa 2010. Ensimmäiset asiakkaille tehtävät asennukset on odotettu toteutettaviksi joulukuussa 2010. Sovelluksen kehitykselle tehtiin versiosuunnitelma, jota ryhdytään noudattamaan välittömästi Nomis VisitorCard -järjestelmän version 2.0 valmistuttua.

Projektin lopputuloksena valmistuu vaatimusmäärittelyn mukainen sovellus, joka soveltuu huomattavasti suuremmalle asiakaskunnalle kuin aikaisempi versio. Sovelluksen selainpohjaisuuden ansiosta käyttö ei ole rajoittunut vain yrityksen sisäiseen toimintaan, vaan se voidaan ulottaa myös asiakkaaseen saakka. Järjestelmäriippuvaisuus saatiin myös purettua, mikä puolestaan mahdollistaa erityyppisten kalenterijärjestelmien käytön sovelluksen osana.

Lähteet

- 1 Microsoft Certified- ja Gold Certified –kumppanit. 2010. Verkkodokumentti. Microsoft Corporation.
<<http://www.microsoft.com/finland/resellers/partners.mspx>>. Luettu 29.9.2010.
- 2 Comparison of revision control software. Verkkodokumentti. Wikipedia.
<http://en.wikipedia.org/wiki/Comparison_of_revision_control_software>. Päivitetty 28.9.2010. Luettu 29.9.2010.

