

MAGENTO VERKKOKAUPPA JA AJANVARAUSKALENTERI

Onnistamo Oy

Jarkko Tissari

Opinnäytetyö

Ammattikorkeakoulututkinto



Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Jarkko Tissari	
Työn nimi Magento verkkokauppa ja ajanvarauskalenteri	
Päiväys 01.05.2011	Sivumäärä/Liitteet 34
Ohjaaja(t) Lehtori Jussi Koistinen ja lehtori Sami Lahti	
Toimeksiantaja/Yhteistyökumppani(t) Toimitusjohtaja Juha Huttunen, Onnistamo Oy	
Tiivistelmä <p>Tämän opinnäytetyön aiheena oli suunnitella ja toteuttaa Onnistamo Oy:n palveluille verkkokauppa, johon lisättiin ajanvarausjärjestelmä. Verkkokauppa on jaettu ylläpito- ja asiakasosioihin. Työ ja määrittely toteutettiin yrityksen toimitusjohtajan kanssa. Verkkokaupan alustana käytettiin Magento-verkkokauppaa. Ohjelmointi toteutettiin pääosin PHP-, JavaScript- ja jQuery-tekniikoilla.</p> <p>Asiakasosion avaintoimintona voidaan pitää palveluiden selaamista ja tilaamista. Tilattu palvelu tallentuu tietokantaan ja se näkyy hallintapaneelissa. Asiakasosiossa asiakas voi varata ajan tilatulle palvelulle.</p> <p>Ylläpito-osion ominaisuuksiin kuuluu uusien palveluiden ja asiakkaiden lisääminen, poistaminen ja seuranta sekä tilauksien vastaanottaminen ja seuranta. Ylläpito-osio sisältää myös hallintapaneelin ajanvarauskalenterille.</p> <p>Lopputuloksena saatiin toimiva verkkokauppa ja ajanvarausjärjestelmä.</p>	
Avainsanat Verkkokauppa, Magento, PHP, JavaScript, jQuery, XML	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Jarkko Tissari			
Title of Thesis Magento eCommerce and Appointment System			
Date	May 1, 2011	Pages/Appendices	34
Supervisor(s) Mr Jussi Koistinen, Lecturer, Mr Sami Lahti, Lecturer			
Project/Partners Mr Juha Huttunen, CEO, Onnistamo Oy			
<p>Abstract</p> <p>The aim of this final project was to design and produce an e-commerce for Onnistamo Ltd's services, in which also an appointment system was added. The store is divided into maintenance and client partitions.</p> <p>The work and functionalities were done with the company's CEO. Magento e-commerce platform was used as the online store platform. The programming was carried out by using mainly PHP, JavaScript and jQuery techniques.</p> <p>The browsing and ordering of services can be considered as the key function of the client partition. The ordered service is stored in the database, and it is shown in the Dashboard. In the client partition the customer can make an appointment for the ordered service.</p> <p>The maintenance section features include adding, deleting and following-up new services and new customers as well as receiving and tracking the orders. The maintenance section also contains the control panel of the appointment system. The end result of this project was a functioning e-commerce and appointment system.</p>			
<p>Keywords E-commerce, Magento, PHP, JavaScript, jQuery, XML</p>			

SISÄLTÖ

1	JOHDANTO	6
2	ASIAKKAAN TARPEET	7
2.1	Verkkokauppa.....	7
2.2	Ajanvarauskalenteri.....	7
3	ALUSTA JA TEKNOLOGIAT.....	8
3.1	Alusta	8
3.2	Muut alustavaihtoehdot	10
3.2.1	osCommerce.....	10
3.2.2	Joomla!.....	10
3.3	Tietokanta	11
3.4	Ohjelmointikielet	11
3.4.1	PHP	11
3.4.2	XML	11
3.4.3	JavaScript	12
3.4.4	jQuery.....	12
3.4.5	Ajax	13
3.5	Css.....	13
3.6	Testipalvelin.....	13
4	SUUNNITTELU.....	14
5	TOTEUTUS	15
5.1	Työkalut	15
5.2	Verkkokauppa.....	15
5.2.1	Valmistelut.....	15
5.2.2	Asennus	15
5.2.3	Räätälöinti	17
5.3	Verkkomaksut.....	19
5.4	Ajanvarauskalenteri.....	24
6	TESTAUS	30
7	POHDINTA	31
	LÄHTEET	33

1 JOHDANTO

Internetin käyttö yleistyy jatkuvasti. Tilastokeskuksen tutkimuksen mukaan Internetin käyttäjiä on Suomessa jo 86 prosenttia. Yleisimpiä netin käyttötapoja ovat sähköposti, verkkopankki ja tiedonhaku tavaroista tai palveluista. (Tilastokeskus 2010.) Internet-liittymien yleistyessä lähes jokaiseen kotitalouteen on moni kuluttaja siirtynyt asiomaan verkossa ja tekemään ostoksensa siellä perinteisten kauppojen sijaan. Verkossa asiointi on aiempaa vaivattomampaa ja yhä kasvavampi osa kaupankäyntiä. Verkkokaupassa tuotteiden ja palveluiden esittely on mahdollista toteuttaa monipuolisemmin kuin perinteisessä myymälässä.

Asiakas voi vertailla tuotteita verkossa kätevästi ja valita itselleen parhaaksi katsomansa tuotteen. Tilatut tuotteet saadaan toimitettua asiakkaalle sujuvammin ja nopeammin noutopisteiden lisääntyessä. Useat yrittäjät ja yritykset haluaisivat siirtää tuotteet ja palvelut myös verkkoon asiakkaiden saataville. Harvalla yrittäjällä on kuitenkaan resursseja ja aikaa pystyttää verkkokauppaa yrityksen voimavaroilla.

Tämä opinnäytetyö käsittelee Onnistamo Oy:n verkkokaupan pystyttämistä Open Source (avoin lähdekoodi) -ohjelmistoalustaa hyväksi käyttäen. Työn tarkoituksena oli pystyttää asiakkaalle helposti ylläpidettävä kauppa ja liittää kauppaan palaverille tarkoitettu ajanvarauskalenteri. Työ koostuu kahdesta osasta. Ensimmäinen osa käsittelee verkkokaupan alustan valintaperusteet, pystytyksen ja räätälöinnin. Alustavaihtoehtoja ei esitellä yksityiskohtaisesti, koska ohjelmistot ovat toiminnoltaan samankaltaisia. Työssä esitellään valitun alustan hyvät ja huonot puolet yleisesti ja pystytykseen liittyvät asiat, asennukseen ja räätälöintiin liittyvät seikat. Työssä otetaan kantaa verkkokaupan ylläpidettävyyteen, laajennettavuuteen ja tuen saatavuuteen.

Työn toinen osa käsittelee ajanvarauskalenterin ja verkkomaksujen suunnittelun, toteutuksen ja liittämisen verkkokauppaan. Ajanvarauskalenteri on palveluille tarkoitettu palaveriaikojen varausjärjestelmä, jolla asiakas voi itse varata palaveriajan tilaamalleen palvelulle. Verkkomaksumuoduli sisältää yleisimmät suomalaisten pankkien verkkomaksut.

2 ASIAKKAAN TARPEET

Onnistamo Oy on Kuopiossa toimiva yritys, jonka liiketoiminta keskittyy tuotekehitykseen. Yrityksen edustaja kuvaa toimintaansa näin: ”Kun tavoitteena on uusi tuote, palvelu tai toimintatapa, joka menestyy kaupallisesti tuottaen reilusti pätäkkaa = Innovaatio!” (Huttunen 2010).

2.1 Verkkokauppa

Asiakkaan tarpeena oli saada helppo ja vastuullista liiketoimintaa tukeva Internet - ostoskeskus, joka tarjoaa laajan tuote- ja palveluvalikoiman niin kutsutuille business to business ja business to customer -sektoreille.

Verkkokaupan tarkoitus on tehdä palveluiden ostamisesta mahdollisimman helppoa, mikä edistää kustannustehokasta liiketoimintaa. Verkkokaupalla asiakas saa joustavan ratkaisun myynnille, sillä ostot voivat tapahtua maantieteellisesti missä päin maailmaa tahansa ajankohdasta riippumatta.

2.2 Ajanvarauskalenteri

Asiakkaan tarpeisiin kuului myös ajanvarauskalenteri, joka on liitetty verkkokauppaan. Kalenterin avulla verkkokaupan asiakas voi varata itselleen sopivan palaveriajan, jonka aikana käydään läpi ostettuun palveluun liittyviä asioita.

Ajanvarauskalenterin tarkoituksena on helpottaa palaveriaikojen varausta ja ylläpidettävyyttä asiakkaan nykyiseen varausjärjestelmään nähden. Kalenterin avulla ostaja saa joustavan ratkaisun palaveriajan varaamiseen, joka ei sido ostajaa paikkaan tai aikaan varausta tehdessä.

3 ALUSTA JA TEKNOLOGIAT

Alustan ja teknologioiden valinnassa päätettiin hyödyntää avoimen lähdekoodin ohjelmistoja ja työkaluja. Käytettävien teknologioiden valinnassa otettiin huomioon sovelluksen eri osat: alusta, tietokanta, verkkokaupan ja ajanvarauskalenterin käyttöliittymät. Teknologioita valittaessa päätettiin myös, mitä kehitysympäristöä sovelluksen toteutuksessa käytetään. Ensiksi tuli määrittää, mitä avoimen lähdekoodin ohjelmistoa käytetään sovelluksen alustana.

Seuraavaksi määritettiin tietokantateknologia, joka on yhteensopiva alustan kanssa. Tämä on yksi tärkeimmistä teknologioista, koska sitä käytetään sovelluksen tarvitseman tiedon tallentamiseen. Kun tietokantateknologia oli määritetty, päätettiin teknologiat käyttöliittymille, joilla varsinainen toteutus tehdään. Teknologian piti olla yhteensopiva alustan kanssa. Päätöksissä otettiin huomioon teknologian suorituskyky ja käytettävyys, jotta sovelluksen käyttö olisi vaivatonta, koska käyttäjien tietokoneet voivat olla suorituskyvyltään heikkoja.

Viimeiseksi kehitysympäristöön valittiin ohjelmistokehitykseen tarpeellisiksi katsotut työkalut ja alustat. Kehitysympäristönä käytettiin kannettavaa tietokonetta, jonka käyttöjärjestelmänä oli Windows XP Professional.

3.1 Alusta

Sovelluksen alustaksi valittiin avoimen lähdekoodin ohjelmisto Magento, koska se tarjoaa laajennettavuutta, joustavuutta ja hyvät ominaisuudet liiketoiminnan kasvulle. Magenton monipuoliset ominaisuudet tarjoavat yrityksille täydellistä ulkoasun, sisällön ja toiminnallisuuden hallittavuutta. Alustassa on selkeä ylläpidon käyttöliittymä, joka sisältää tehokkaat hakukoneoptimointi- ja markkinointipalvelut sekä tuoteluettelon hallintavälineet. Magento antaa yrityksille mahdollisuuden luoda omiin tarpeisiinsa räätälöityjä sivustoja ja laajentuu suurempienkin yritysten väliseen kaupankäyntiin. (Varien 2011.)

Magento ohjelmiston kattavuus on useisiin Open Source -pohjaisiin verkkokaupparatkaisuihin verrattuna omaa luokkaansa. Se on täysin globaali sovellusalusta, joka mahdollistaa liiketoiminnan laajentamisen esimerkiksi ulkomaille tai yritys voi halutesaan räätälöidä kaupan vastaamaan asiakkaidensa erityistarpeita. Magento on käännetty yli 60 kielelle ja se tukee useita valuuttoja, veroja ja maksutapoja, joten se mahdollistaa täyden kansainvälistymisen online-kanavan yritykselle. (Varien 2011.)

Magenton toiminnallisuuksiin kuuluvat kaupankäyntiä edistävät valmiit työkalut, esimerkiksi integroitu Google Analytics, ja laaja raporttien selaamismahdollisuus. Raportit näyttävät kaupan suorituskyvyn ja antavat mahdollisuuden reagoida asiakkaiden tarpeisiin paremmin. Raporteista ylläpitäjä voi seurata muun muassa myyntiraportteja, parhaiten myydyimpiä ja eniten katsottuja tuotteita, asiakkaiden antamia tuotearvosteluja ja hakusanoja sekä vero- ja laskutusraportteja. (Varien 2011.)

Magentossa on joustava tuotemyyntimoottori, joka mahdollistaa tuoteluetteloinnin. Se on tehty tukemaan liiketoimintaa ja helpottamaan asiakasta tuoteselailussa. Moottori tarjoaa käyttäjälle tuotearvostelut, toivelistat tuotteille, varastosaatavuudet ja paljon tuotteiden selaamiseen liittyviä ominaisuuksia. Magenton ominaisuudet ja joustava käyttöliittymän mallinnus mahdollistavat erittäin käyttökelpoisen sivujen tekemisen täydellisellä muokattavuudella. Sillä voi esimerkiksi luoda kategorioiden staattisia aloitussivuja, tuotekohtaisia sivustoja, tuotteiden vertailusivuja ja näyttää tuotteet listana tai taulukkona. (Varien 2011.)

Magenton käytännöllinen käyttöliittymä mahdollistaa tehokkaan hallinnoinnin suoraan verkossa. Yhdessä hallintapaneelissa onnistuu useamman varaston hallinta, sekä paikallisten ja kansainvälisten kauppasivustojen hallinta. Magenton päivitys on helppoa, sillä sovelluksen yhden napsautuksen päivitystoiminto mahdollistaa saumattoman päivittämisen uudempaan versioon. (Varien 2011.)

Magenton hyviä puolia ovat sen monipuolisuus ja laajennettavuus. Magenton monipuolinen kokonaisuus antaa käyttäjälleen mahdollisuuden luoda täysin omaan tarpeeseen sopivat sivustot. Magento tarjoaa kotisivuillaan myös laajennuksia, joilla saa enemmän toiminnallisuutta ja näyttävyyttä sivuille. Osa niistä on maksullisia ja osa ilmaisia, joten kustannuksetkaan eivät välttämättä kasva. Magenton hyviä puolia ovat myös sen kotisivuilla olevat tukipalvelut ja keskustelupalstat, joilta saa todella paljon neuvoa tarvittaessa niin ylläpitoon kuin kehitykseenkin liittyvissä asioissa.

Magenton heikkous voi olla sen suurehko koko, jos käyttäjän tarpeet ovat pienet. Magento on suunnattu enemmän suurehkojen kuin pienikokoisten yritysten käyttöön. Vaikka Magento saa ladattua ilmaiseksi, se ei ole täysin ilmainen. Maksuja kertyy, jos täytyy palkata ulkopuolinen ohjelmoija tai graafikko muokkaamaan kaupan käyttöliittymää. Maksuja syntyy palvelinmaksuista, toimituksista ja mahdollisista verkkomaksusopimuksista.

3.2 Muut alustavaihtoehdot

Projektin aloitusvaiheessa Magenton lisäksi tutkittiin eri alustavaihtoehtoja. Tutkitut alustat olivat OsCommerce ja Joomla!, jotka ovat myös avoimen lähdekoodin ohjelmistoalustoja.

3.2.1 osCommerce

osCommerce on lisenssimaksuton verkkokauppaohjelma, jonka useat lisäominaisuudet mahdollistavat yksilöllisen verkkokaupan asiakkaan tarpeisiin. osCommerce mahdollistaa www-sivujen yhdistämisen, joten kotisivujen hallinta saadaan samaan järjestelmään. (Ponce de Leon 2011.)

osCommerce toimii millä tahansa palvelimella, johon on asennettu MySQL ja PHP (Ponce de Leon 2011). osCommerce ei ole perusominaisuuksiltaan monipuolinen ja ominaisuuksia lisättäessä tietokannan rakenne muuttuu, joka tekee tietokannasta sekavan ja tietojen siirto vaikeutuu (Aarne 2007). Myös osCommercen maksumoduulien turvallisuudesta on löydetty haavoittuvuuksia, missä harhautetaan verkkokauppasovellusta kertomalla maksun onnistuneen. (CERT-FI 2009). Lisättävien ominaisuuksien mukana tullut tietojen hallinnan vaikeutumien ja mahdollinen verkkokauppasovelluksen harhauttaminen verkkomaksuissa vaikuttivat alustavalintaan.

3.2.2 Joomla!

Joomla! on monipuolinen julkaisujärjestelmä, jota ylläpidetään pelkällä Internet-selaimella. Sen avulla voidaan lisätä ja päivittää sivujen sisältöä ilman ulkopuolisen apua. Joomla! sopii niin yksityishenkilöille kuin yrityksille tai yhteisöille. Käyttömahdollisuuksia on rajattomasti. Sillä voidaan tehdä normaaleja verkkosivuja, yhteisöjä ja verkkokauppoja. (Joomlaportal 2007.)

Joomla! tarvitsee toimiakseen Apache palvelimen sekä MySQL- ja PHP-tuen (Joomlaportal 2007). Joomla!-alustan käyttäminen verkkokauppana vaatii vapaasti ladattavan VirtueMart:n, joka on Joomla!-sisältöhallintaohjelman lisäosa (Arhi 2011). Se on yksinkertainen verkkokauppasovellus, joka ei ole täysin valmis käytettäväksi yritysten kaupankäyntiin eikä sitä ole testattu sataprosenttisesti tietoturvaohjelmilta (Walker 2009). Joomla! vaihtoehtoa ei valittu projektissa käytettäväksi alustaksi, koska se tarkoitettu lähinnä normaalien Internet-sivustojen tekemiseen ja ylläpitoon. VirtueMart-lisäosan testauksen puutteet vaikuttivat myös päätökseen.

3.3 Tietokanta

Tietokannaksi valittiin MySQL-tietokantaohjelmisto, sillä se sopii hyvin yhteen Magento alustan kanssa. Toinen valintaperuste oli sen kustannukset, koska se on ilmainen yksityiselle käyttäjälle ja se on helppokäyttöinen. MySQL on maailmanlaajuisesti erittäin suosittu SQL-tietokannan hallintajärjestelmä verkkopohjaisissa sovelluksissa

MySQL on monipuolinen ja suorituskykyinen relaatiotietokanta, jota käytetään WWW-palvelujen taustalla. MySQL tietokannanhallintajärjestelmät vastaavat tietokantajärjestelmien kaikkien tietojen tallennuksesta, turvallisuudesta, eheydestä, ajantasaisuudesta, toipumiskyvystä sekä niihin käsiksi pääsystä (Page & Hughes 1999).

3.4 Ohjelmointikielät

3.4.1 PHP

PHP (*Hypertext Preprocessor*) on komentosarjakieli eli skriptikieli. Sitä käytetään palvelinympäristöissä ja web-sivujen luontiin. PHP-ohjelmakoodi käännetään vasta ohjelman suoritusvaiheessa ja tulos palautetaan käyttäjälle esitettäväksi. Yleisimmin PHP:tä upotetaan HTML-sivujen sisään käyttäen PHP:lle varattuja (<?php , ?>) aloitus- ja lopetustageja. (Heinisuo 2004.)

PHP:n hyviä puolia ovat sen laajat luokkakirjastot http-protokollalle ja ulkoisille tietokannoille, mitkä helpottavat useiden toiminta-alueiden, kuten päivämäärien ja merkkijonojen käsittelyssä. PHP sisältää muun muassa MySQL, PostgreSQL, SQLite ja Oracle tietokantatuuet.

3.4.2 XML

XML (eXtensible Markup Language) on merkintäkieli tekstimuotoisille dokumenteille. Sitä käytetään dokumenttien loogisen rakenteen mallintamiseen ja formaattina dokumenttien tallentamiseen. Keskeisenä ideana on saada tekstimuotoinen tieto tallennettua tietorakenteina dokumenttiin, jotta tiedot ovat helposti luettavissa. (Nykänen 2001.)

3.4.3 JavaScript

JavaScript on Web-ympäristössä käytettävä ohjelmointikieli, johon viitataan usein termillä skriptikieli. JavaScriptin on alun perin kehittänyt yhdysvaltalainen yhtiö Netscape Communications Corporation. Tärkein mahdollisuus on lisätä toiminnallisuutta dynaamisille Web-sivustoille. Skripti laitetaan HTML-sivuille `<script>`-tagien väliin tai sijoitetaan omaan tiedostoonsa. JavaScript on clientin puolen ohjelmointikieli, koska se on suunniteltu toimimaan käyttäjän tietokoneella eikä palvelimella. Sen vuoksi käyttäjän selaimessa pitää olla tuki JavaScriptille. (Negrino, Smith 2007.)

JavaScriptin avulla ohjelmoija voi luoda muokattuja HTML-sivuja, joiden sisältö riippuu käyttäjän toiminnoista. Kielen avulla voi myös luoda verkkopohjaisia sovelluksia, käsitellä lomakkeita ja tehdä kalentereita sekä aikaleimoja dokumentteihin. (Moncur 2000.)

3.4.4 jQuery

jQuery on maailman suosituin JavaScript-kirjasto, joka toimii yhtenäisesti kaikilla eri selaimilla. Se on suunniteltu yksinkertaistamaan clientin puolen skriptiä HTML:ssä. jQuery julkaistiin vuonna 2006, ja sen alkuperäinen luoja on John Resig. (Chaffer 2009.)

jQuery on ilmainen avoimen lähdekoodin ohjelmisto, jolla on kaksoislisenssi. Sen lisenssit ovat MIT ja GPL, jotka ovat molemmat vapaiden ohjelmistojen lisenssejä. (Chaffer 2009). GPL antaa kenelle tahansa oikeuden käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmia ja niiden lähdekoodia. MIT sallii teoksen käytön myös kaupallisissa suljetun lähdekoodin ohjelmistoissa ja ei vaadi lähdekoodin julkistamista vaikka muokattua teosta levitetäisiin eteenpäin.

jQuery:n syntaksi on suunniteltu siten, että sen avulla käyttäjän on helppo navigoida sivulla, luoda animaatioita, käsitellä elementtejä ja kehittää Ajax sovelluksia. jQuery antaa kehittäjille mahdollisuuden luoda animaatioita, edistyksellisiä tehosteita ja korkeita tasoisia, vaikuttavia widgettejä. Nämä auttavat vaikuttavien ja dynaamisten web-sivustojen luonnissa. (Chaffer 2009.)

3.4.5 Ajax

Ajax (Asynchronous JavaScript And XML) on yksi ohjelmointitermeistä, joka sisältää joukon web-sovelluskehityksen tekniikoita. Sen avulla voi tehdä vuorovaikutteisempia web-sovelluksia kuin pelkällä PHP-kielellä. Ajax-tekniikalla tehdyt toiminnot perustuvat sovelluksen tietojen käsittelyyn, mitkä tapahtuvat suurimmalta osalta käyttäjän selaimella ja lähetetyt pyynnöt palvelimelle ovat lyhyitä. Tällöin koko verkkosivua ei tarvitse ladata uudelleen. (Negrino & Smith 2007.)

Ajax:n tavoitteena on lisätä web-sovelluksen vuorovaikutteisuutta sekä nopeuttaa ja lisätä käytettävyyttä toiminnallisuuksiin. Ajaxin avulla saadaan verkkosovelluksista työpöytäsovelluksen tapaisia nopeasti latautuvia ja käyttökokemukseltaan näppäriä. (Negrino & Smith 2007.)

3.5 Css

Css (Cascading Stylesheets) on merkintäjärjestelmä, jolla voidaan määrittää selaimen dokumenteille tyyliohjeita. Nykyään css-tiedostot ovat erillään HTML-tiedostoista, minkä vuoksi tiedostokoot pienevät ja latautuvat nopeammin. Samalla tiedostojen ylläpidettävyys helpottuu, koska päivitettäviä tiedostoja on vain yksi tai muutama. (Sarja 2010.)

Css-tiedostoon sijoitetaan kaikki verkkosivujen elementtien tyyliin liittyvät seikat. Esimerkiksi tagille p voidaan määrittää kirjainkoko ja väri tai table tagin reunojen paksaus. Tyyli-tiedoston avulla sivustojen ulkoasut saadaan selkeiksi. (Sarja 2010.)

3.6 Testipalvelin

Testipalvelimenä käytettiin WAMP (Windows Apache MySQL PHP) -ohjelmistokokonaisuutta. Se on muunnos vastaavasta Linuxille tarkoitettusta ohjelmistokokonaisuudesta LAMP:sta. Kokoelma ohjelmista muodostaa palvelimen, jolla voidaan ajaa web-sovelluksia. (Sarja 2010.)

WAMPin sisältämät ohjelmat vaikuttivat testipalvelimen valintaan, koska siinä tarvittavat sovellukset ovat yhdessä paketissa. Testipalvelimessa on Apache webpalvelimen, MySQL tietokantarajapinnan ja PHP, Perl ja Pythonin komentosarjakielet. WAMPin asennus ja käyttö on helppoa, ja mukana tulee valmiina graafinen käyttöliittymä MySQL tietokannalle.

4 SUUNNITTELU

Projektin aloitusvaiheessa tehtiin projektisuunnitelma projektin kulusta. Suunnitelmassa käytiin läpi projektissa käytettävät tekniikat, vaiheet, aikataulut ja tavoitteet. Suunnitteluvaiheen tarkoituksena oli löytää parhaat ja kustannustehokkaimmat ratkaisut monipuolisen verkkokaupan toteutukseen. Ensisijaisesti suunnittelun kohteena oli asiakkaan toinen projekti, johon verkkokauppaa olisi toteutettu. Senhetkisen määrittelyn perustana oli saada kattava ja toimiva verkkokauppasovellus, joka vastaisi ulkoasultaan ja toiminnoiltaan asiakkaan toiseen projektiin liittyviä määritelmiä.

Suunnitelmat muuttuivat projektin edetessä ja asiakas tahtoi verkkokaupan omaan käyttöönsä. Määrittelyt vaihtuivat ja lähtökohdaksi tuli Onnistamo Oy:n oma kotisivu. Kotisivujen ulkoasu ja palvelut tuli sisällyttää verkkokauppasovellukseen samalla tavalla, jotta ne olisivat yhdenmukaiset. Samankaltaisuudella haetaan sivustojen yhtäläisyyttä, jolloin vanhat käyttäjät löytävät helposti palvelut ja yritys on huomannut kyseisen käyttöliittymän toimivaksi.

Verkkokaupasta ei laadittu tarkempaa määrittelydokumenttia, koska yrityksen kotisivujen käyttöliittymän toimintaperiaate tuli siirtää alustalle. Myös alustan valmiit toiminnot olivat monipuolisia, mikä helpotti tiettyjä toimintoja. Verkkokauppasovellusta laajennettiin maksumoduulilla. Maksumoduulin tarkoituksena oli saada liitettyä suomalaiset verkkopankkimaksutoiminnot kauppaan. Verkkokauppaan liitettiin myös ajanvarauskalenteri.

Ajanvarauskalenterista ei laadittu tarkempaa määrittelydokumenttia, koska se ei ollut tarpeen. Ajanvarauskalenterin tuli sisältää perustoimintoina päivämäärän ja kelloajan valinnan viikkonäkymästä sekä näyttää varatut ajat. Ajanvarauskalenteriin tuli tehdä myös hallintapaneeli, jonka avulla ylläpitäjä pystyy vaihtamaan, poistamaan ja selaamaan varattuja aikoja sekä lisäämään uusia aikoja.

Verkkomaksuista ei tarvinnut tehdä määrittelydokumenttia, koska verkkomaksuihin tarvittavat rajapintakuvaukset ja määrittelyt saatiin valmiiksi verkkomaksuihin perehtyneiltä yrityksiltä. Suomalaisen pankkien verkkomaksuihin erikoistuneet yritykset ovat Suomen Verkkomaksut Oy ja Checkout Finland Oy. Verkkomaksuissa päätettiin käyttää Checkout Finland Oy:n tarjoamaa rajapintaa ja maksupalvelua.

5 TOTEUTUS

Tässä luvussa käsitellään Magenton verkkokaupan ja ajanvarauskalenterin pystyttämiseen ja toteutuksen toimenpiteitä.

5.1 Työkalut

Sovelluksen kehityksessä käytettiin avoimen lähdekoodin kehitystyökaluja. Koodien kirjoittamiseen valittiin Notepad++, joka on ilmainen lähdekoodieditori ja tukee sovelluksessa tarvittavia ohjelmointikieliä.

WAMP asennettiin testipalvelimeksi, jota voitiin käyttää tietokannan ja sovelluksen testaukseen.

5.2 Verkkokauppa

5.2.1 Valmistelut

Valmistelut aloitettiin lataamalla pakatut tiedostot verkkokauppa sovelluksesta ja tietokannan esimerkkidatut Magenton kotisivuilta. Latausta ennen tuli ottaa huomioon järjestelmävaatimukset. Magento tarvitsee Apache-palvelimen version 1.x tai 2.x ja vähintään 5.2.0 version PHP-tulkista. Latausten jälkeen pakatut kansiot tulee purkaa palvelimelle.

Ennen alustan asennusta piti luoda tietokanta sovellukselle. Valmisteluissa oli otettava huomioon, että Magento käyttää PHP-kirjastoja pdo_mysql ja mcrypt, jotka tuli asettaa päälle testipalvelimen php.ini-tiedostoon. Samalla tuli asettaa päälle Apachen moduuleista rewrite_module. Kun valmistelut olivat valmiit ja tietokanta luotu, voitiin siirtyä asennukseen.

5.2.2 Asennus

Verkkokaupan varsinainen toteutus aloitettiin alustan pystyttämällä. Asennus aloitettiin avaamalla nettiselain ja antamalla puretun kansion osoite, joka avaa asennuspaketin. Testipalvelimella osoite voi olla esimerkiksi <http://localhost/magento>. Asennus eteni helposti ja yksinkertaisesti ohjeita noudattaen. Asennuksen edetessä hy-

väksyttiin ohjelmiston lisenssiehdot ja asetettiin sijainti- ja konfigurointitiedot sekä luotiin ylläpitäjätunnus.

Yleisimpiä ongelmia voi kuitenkin tulla eteen asennuksen konfigurointi vaiheessa, jos käyttää Windows -järjestelmää, koska Magento on tarkoitettu käytettäväksi Linux-palvelimilla. Esimerkiksi asennuksessa esiintyi ongelma testipalvelimen osoitteen kanssa, minkä avulla sovellusta käytetään. Yleensä osoitteelle annetaan osoite *http://localhost/asennuskansio*, mutta Windows-järjestelmässä osoiterivi tulikin korvata *http://127.0.0.1/asennuskansio*, jossa 127.0.0.1 on oman koneen aina saavutettavissa oleva ns. loopback-osoite (KUVA 1).

The screenshot shows the 'Configuration' step of the Magento installation process. On the left, there is a navigation menu under 'Installation' with options: Download, License Agreement, Localization, Configuration (highlighted), Create Admin Account, and You're All Set!. Below the menu, there is a link for 'Having trouble installing Magento? Check out our Installation Guide'.

The main configuration area is titled 'Configuration' and contains three sections:

- Database Connection:**
 - Host *: localhost
 - Database Name *: magento
 - User Name *: root
 - User Password: (empty)
 - Tables Prefix: (empty)
 - (Optional. Leave blank for no prefix)
- Web access options:**
 - Base URL *: http://127.0.0.1/store/
 - Admin Path *: adminpanel
 - Additional path added after Base URL to access your Administrative Panel (e.g. admin, backend, control etc.):
 - Enable Charts: Enable this option if you want the charts to be displayed on Dashboard.
 - Skip Base URL Validation Before the Next Step: Check this box only if it is not possible to automatically validate the Base URL.
 - Use Web Server (Apache) Rewrites: You could enable this option to use web server rewrites functionality for improved search engines optimization. Please make sure that mod_rewrite is enabled in Apache configuration.
 - Use Secure URLs (SSL): Enable this option only if you have SSL available.
- Session Storage Options:**
 - Save Session Data In: File System

KUVA 1 Asennuksen konfigurointi osio

Magenton hallintopaneeliin pääsee kirjautumaan lisäämällä selaimen osoiteriville, varsinaisen osoitteen perään, konfigurointiosiossa annetun hallintapaneelin polun *http://127.0.0.1/store/index.php/adminpanel*. Hallintapaneeliin kirjautuessa voi esiin-

tyä ongelma, ettei kirjautuminen onnistu oikeilla tunnuksilla, jos sovelluksen palvelinta käytetään lokaalina.

Ongelma voidaan ratkaista avaamalla tiedosto *Varien.php*, joka sijaitsee asennuskansioon *app\code\core\Mage\Core\Model\Session\Abstract* kansiossa, ja laittamalla kommentteihin seuraavat evästeiden parametrien kohdat:

```
// 'domain' => $cookie->getConfigDomain(),
// 'secure' => $cookie->isSecure(),
// 'httponly' => $cookie->getHttponly().
```

5.2.3 Räättälöinti

Kirjautumisen onnistuessa selaimelle avautuu seuraavanlainen ikkuna (KUVA 2).

The screenshot shows the Magento Admin Panel dashboard. At the top, there is a search bar and a navigation menu with options like Dashboard, Sales, Catalog, Mobile, Customers, Promotions, Newsletter, CMS, Reports, and System. Below the navigation, there are several warning messages and a 'Latest Message' section. The main dashboard area is titled 'Dashboard' and includes a 'Choose Store View' dropdown set to 'All Store Views'. The dashboard contains several widgets: 'Lifetime Sales' showing \$0.00, 'Average Orders' showing \$0.00, 'Last 5 Orders' with a table showing 'No records found.', and 'Last 5 Search Terms' with a table listing search terms and their usage. The search terms table is as follows:

Search Term	Results	Number of Uses
cloqs	1	391
nike west	1	22
sony	3	1092
rebel	1	280
shone	6	462

Below the search terms table, there are tabs for 'Bestsellers', 'Most Viewed Products', 'New Customers', and 'Customers'. The 'Most Viewed Products' tab is active, showing a table with columns for 'Product Name', 'Price', and 'Quantity Ordered', which currently displays 'No records found.'.

KUVA 2 Magenton hallintapaneelin etusivu

Kuten kuvasta huomaa, Magento on todella kattava jo itsessään ja se on suunnattu myös suurien yritysten kaupankäyntiin. Koska Magentolla on kansainvälinen levinneisyys, se tarjoaa kielituen monille maille. Kielipaketti on helposti saatavilla Magenton kotisivuilta, se sijaitsee *Community*-valikon *Translations*-kohdassa. Ladatussa paketissa on kaksi *fi_FL* nimistä kansiota, jotka ovat tallennettuina verkkokauppaa vastaa-

vaan hakemistorakenteeseen. Kansiot tulee tallentaa palvelimelle oikeisiin alihakemistoihin, jotka ovat *app/locale* ja *app/design/frontend/default/default/locale*. Tallennuksen jälkeen pitää muuttaa Magenton hallintapaneelissa olevaa sijaintiasetusta. Sijaintiasetus löytyy kohdasta *System* → *Configuration* → *General* → *Locale Options* josta muutetaan kentät vastaamaan Suomen tietoja. Kielipaketin asennuksen toivuus kannattaa varmistaa myös tyhjentämällä sovelluksen välimuisti kohdasta *System* → *Cache Management* → *Flush Magento Cache*. Kielipaketeista pitää huomioida, etteivät suomennokset ole sataprosenttisesti valmiita, joten myös itselle jää jonkun verran suomentamista.

Magenton ulkoasun muokkaus on yksinkertaista. Magento tarjoaa muutamia eri ulkoasuja asennuspaketin mukana ja lisäksi Magenton kotisivut tarjoavat valmiita ulkoasuja. Halutessaan omanlaisen käyttöliittymän pitää muokata valmiita ulkoasutiedostaja tai tehdä kokonaan itse. Kaikki ulkoasuun liittyvät asiat kuten tyylitiedostot ja kuvat käyttöliittymälle sekä grafiikka löytyvät *skin* nimisestä kansioista. Uuden ulkoasun muokkaaminen aloitettiin luomalla kansiot hakemistorakenteeseen. Esimerkkinä ovat Onnistamo Oy:n hakemistot:

```
app/design/frontend/onnistamo/onnistamo_theme/,
skin/frontend/onnistamo/onnistamo_theme/.
```

Molemmille kansioille luotiin myös tarvittavat alakansiot, joissa määritellään tarkemmat käyttöliittymät tiedot. Nämä kansiot ovat:

```
/app/design/frontend/onnistamo/onnistamo_theme/layout
/app/design/frontend/onnistamo/onnistamo_theme/template
/skin/frontend/onnistamo/onnistamo_theme/css/
/skin/frontend/onnistamo/onnistamo_theme/images/
```

Hakemistorakenteessa *onnistamo* on ulkoasun rajapinta ja *onnistamo_theme* viittaa ulkoasun teeman. *Layout* kansiossa on sivujen rakenteet ja *template* osiossa sivujen varsinainen käsittely. *Layout* -kansiossa oleva *page.xml* tiedosto näkyy kaikilla sovelluksen sivuilla, joten siihen on hyvä liittää tarvittavat javascript kirjastot ja tyylitiedostot sekä muut menetelmät. Kirjastojen lisääminen tehtiin kuvan kolme mukaisesti (KUVA 3). Ensisijaisena ollut projekti aloitettiin samoilla toimenpiteillä toteutusvaiheessa. Vaikka suunnitelmat muuttuivat toteutuksen edetessä, toteutukseen käytetty aika ja työmäärä antoivat jo paremman kuvan Magenton rakenteesta ja toiminnallisuuksista. Tästä johtuen oli helppo ja nopea vaihtaa verkkokaupan aihe Onnistamo Oy:lle.

```

<default translate="Label" module="page">
  <label>All Pages</label>
  <block type="page/html" name="root" output="toHtml" template="page/3columns.phtml">
    <block type="page/html_head" name="head" as="head">
      ...
      <action method="addJs"><script>jquery/jquery.js</script></action>
      <action method="addJs"><script>jquery/domwindow.js</script></action>
      ...
    </block>
  </block>
</default>

```

KUVA 3 Ajanvarauskalenterin avaamiseen tarvittavien jQuery tiedostojen määrittäminen page.xml tiedostoon.

Ulkoasu muutokset tehtiin *style.css*-tiedostoon. Muutoksia tehtiin ylä-, ala- ja sivupalkkien kokoon, sijaintiin ja sisältöihin. Näkyvimpiä muutoksia, jotka voidaan havaita käyttöliittymässä, olivat taustojen vaihtaminen ja kuvien lisäämiset sekä fonttien tyyliasetuksien vaihdot. Käyttöliittymään asetettiin kolme eri taustaa, jotka viittaavat värikkään omaan palvelukategoriaan. Taustojen aktivoiminen toteutettiin kategorian vaihtumisen mukaan. Fonttien tyyliasetukset muutettiin vastaamaan Onnistamo Oy:n kotisivujen fonttiasetuksia.

5.3 Verkkomaksut

Verkkokauppaan lähdettiin laajentamaan verkkomaksuominaisuudella. Magenton laajentaminen onnistuu lisäämällä sovellukseen moduuleita. Verkkokauppaan tehtiin maksumoduuli, vaikka valmiiksi testattuja ja toimivia maksumoduuleita on markkinoilla. Syynä oman maksumoduulin luomiseen olivat kustannussyyt, koska kyseessä on vielä demokauppa ja maksullisia yhteyksiä ei tarvita. Maksumoduulin tarkoituksena oli päästä testaamaan maksuliikenneyhteyksiä verkkomaksuihin tarkoitetun testipalvelimen avulla.

Moduulien asetukset Magentoan tarvitsevat xml-kielellä vaadittavia asetuksia, jotka sisältävät moduulin nimen, määrittelyn ja aktiivisuudesta kertovan elementin (KUVA 4). Määrittelyllä kerrotaan kuuluuko moduuli Magentoan ydinmoduuleihin vai räätälöityihin moduuleihin. Kaikki räätälöidyt moduulit määritetään local-ryhmään. Jos sovelluksessa on räätälöityjä moduuleja, Magento tarkistaa ne ennen ydinmoduuleiden käyttöä. Silloin sovellus saa mahdolliset tiedot Magento ydinmoduuleita ylikirjoittavista tai laajennettavista moduuleista. Moduuli sijoitetaan hakemistorakenteeseen */app/etc/modules/*.

```
<?xml version="1.0"?>
<config>
  <modules>
    <onnistamo_Maksumoduuli>
      <active>true</active>
      <codePool>local</codePool>
      <depends>
        <Mage_Payment />
      </depends>
      <version>0.1.0</version>
    </onnistamo_Maksumoduuli>
  </modules>
</config>
```

KUVA 4 Maksumoduuliin vaaditut asetukset, jossa määritetään moduulin nimi ja aktiivisuus.

Jokainen moduuli tarvitsee erilliset konfigurointitiedostot, jotka luodaan myös xml-kielillä. Tiedostossa määritellään moduulin luokat ja metodit, joita moduuli tulee hyödyntämään. Konfigurointitiedostossa määriteltiin myös kaikki moduulin tarvitsevat tiedostot ja oletusarvot (KUVA 5). Oletusarvoissa olevat kauppias- ja salaustunnisteiden arvot ovat testipalvelimen arvoja.

```

....
<global>
  <models>
    <maksumoduuli>
      <class>Onnistamo_Maksumoduuli_Model</class>
    </maksumoduuli>
  </models>
  <helpers>
    <maksumoduuli>
      <class>Onnistamo_Maksumoduuli_Helper</class>
    </maksumoduuli>
  </helpers>
....
....
  <layout>
    <updates>
      <maksumoduuli_redirect>
        <file>maksumoduuli.xml</file>
      </maksumoduuli_redirect>
    </updates>
  </layout>

  <translate>
    <modules>
      <Onnistamo_Maksumoduuli>
        <files>
          <default>Onnistamo_Maksumoduuli.csv</default>
        </files>
      </Onnistamo_Maksumoduuli>
    </modules>
  </translate>

</frontend>

<adminhtml>
  <translate>
    <modules>
      <Onnistamo_Maksumoduuli>
        <files>
          <default>Onnistamo_Maksumoduuli.csv</default>
        </files>
      </Onnistamo_Maksumoduuli>
    </modules>
  </translate>
</adminhtml>

<default>
  <payment>
    <maksumoduuli>
      <active>0</active>
      <model>Onnistamo_Maksumoduuli_Model_Standard</model>
      <title>Suomen verkkomaksut</title>
      <merchant_id>375917</merchant_id>
      <merchant_secret>SAIPPUAKAUPPIAS</merchant_secret>
      <delivery_time>14</delivery_time>
      <sendmailorder_confirmation>true</sendmailorder_confirmation>
      <payment_action>authorize_capture</payment_action>
      <allow_payments_under_1_eur>false</allow_payments_under_1_eur>
    </maksumoduuli>
  </payment>
</default>
....

```

KUVA 5 Konfigurointitiedostoon määriteltyjä tiedostoja ja oletusarvoja.

Maksumoduulille tehtiin myös toinen konfigurointitiedosto *system.xml*, joka sisältää hallintapaneelin puoleiset asetukset ja näkymät. Maksumoduulin hallintapaneelin konfigurointitiedostoon määriteltiin mitä järjestelmän moduulia maksumoduuli käyttää hyväkseen ja maksumoduulin kentät, jotka pysyvät vakiona (KUVA 6). Yleensä vakiokenttiä ovat kauppiaan tunnus ja salausavain, jolla pankit tunnistavat verkkokaupasta suoritettavat maksut.

```

<?xml version="1.0"?>
<config>
  <sections>
    <payment>
      <groups>
        <maksumuoduli translate="label" module="paygate">
          <label>verkkomaksut</label>
          <sort_order>670</sort_order>
          <show_in_default>1</show_in_default>
          <show_in_website>1</show_in_website>
          <show_in_store>0</show_in_store>

          <fields>
            <active translate="label">
              <label>Enabled</label>
              <frontend_type>select</frontend_type>
              <source_model>adminhtml/system_config_source_yesno</source_model>
              <sort_order>10</sort_order>
              <show_in_default>1</show_in_default>
              <show_in_website>1</show_in_website>
              <show_in_store>0</show_in_store>
            </active>

            <title translate="label">
              <label>Title</label>
              <frontend_type>text</frontend_type>
              <sort_order>20</sort_order>
              <show_in_default>1</show_in_default>
              <show_in_website>1</show_in_website>
              <show_in_store>0</show_in_store>
            </title>

            <merchant_id>
              <label>Merchant ID</label>
              <frontend_type>text</frontend_type>
              <sort_order>59</sort_order>
              <show_in_default>1</show_in_default>
              <show_in_website>1</show_in_website>
              <show_in_store>0</show_in_store>
            </merchant_id>

            <merchant_secret translate="label">
              <label>Secret</label>
              <comment>Merchants secret key</comment>
              <sort_order>60</sort_order>
              <show_in_default>1</show_in_default>
              <show_in_website>1</show_in_website>
              <show_in_store>0</show_in_store>
            </merchant_secret>
          </fields>
        </maksumuoduli>
      </groups>
    </payment>
  </sections>
</config>

```

KUVA 6 System.xml-tiedostossa sijaitsevien kenttien ja käytettävän moduulin määrittäviä.

Maksumoduulille tehtiin *Standard.php*-tiedosto, jota käytetään verkkomaksuihin tarvittavien rajapinta-arvojen määrittämiseen. Tiedoston tehtävä on hakea tilauksesta tarvittavat arvot ja muodostaa maksurajapinta form, joka lähetetään eteenpäin maksupalveluun tilausta maksettaessa. Maksumoduulin maksupalveluna käytettiin Checkout Finland Oy:n tarjoamaa palvelua (<https://payment.checkout.fi>). Kun rajapintaan saatiin kaikki tarvittavat tiedot taulukon muistiin, luotiin taulukosta resurssikonteksti *stream_context_create()*-funktioilla, koska kontekstin avulla saadaan muokattua maksupalveluun liittyviä asetuksia, jolloin saadaan vietyä oikeat tiedot palveluun (Gilmore 2005). Asetukset lähetettiin palveluun kutsumalla *file_get_contents("https://payment.checkout.fi", false, \$context)*-funktioita, joka palauttaa <https://payment.checkout.fi/> toimialueen hakemistosivun sisällön.

Maksumoduulille tehtiin moduulin toimintoja ohjaava kontrolleritiedosto, *Standard-Controller.php*. Tiedoston tarkoituksena on ohjata tilauksen maksaminen verkkomak-

sutoimintoon, jos tilaaja on kassan yhteydessä valinnut maksutavaksi verkkomaksun. Kontrolleria käytetään myös takaisinohjauksessa verkkokauppaan maksusuorituksen jälkeen. Tiedostossa tapahtuva ohjaus verkkomaksuun määriteltiin *redirectAction()*-funktiossa (KUVA 7), jossa haetaan *Redirect.php*-tiedoston sisältö (KUVA 8). *Redirect.php* tiedosto palauttaa HTML-koodina maksutapavaihtoehdot suomalaisiin verkkopankkeihin (KUVA 9). Ennen HTML-koodin generointia *Redirect.php*-tiedosto kutsuu *Standard.php*-tiedostosta aiemmin esitellyn <https://payment.checkout.fi/toimialueen/hakemistosivun/sisallon>, XML-asiakirja, joka luetaan objektiin *simplxml_load_string()*-funktion avulla. Objekti sisältää myös maksupalvelussa käytössä olevat verkkopankit. Palautettava HTML koodi generoidaan objektista käyttäen hyväksi *foreach*-metodia, jossa eritellään pankkien osoitteet ja tunnukset.

```
public function redirectAction()
{
    ...
    $this->loadLayout();
    $this->getLayout()->getBlock('content')->append($this->getLayout()->createBlock('maksumoduuli/standard_redirect'));
    $this->renderLayout();
}
```

KUVA 7 Redirect.php-tiedoston kutsu *redirectAction()*-funktiossa

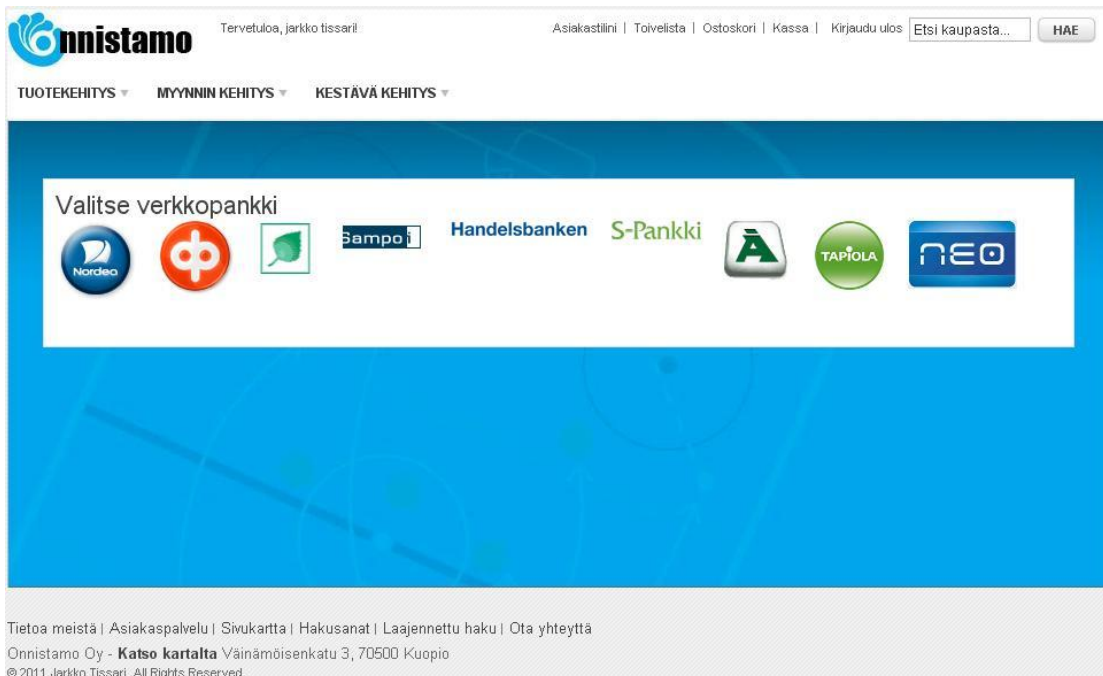
```
<?php
class Onnistamo_Maksumoduuli_Block_Standard_Redirect extends Mage_Core_Block_Abstract
{
    protected function _toHtml()
    {
        $session = Mage::getSingleton('checkout/session');
        $standard = Mage::getModel('maksumoduuli/standard');
        $order = Mage::getModel('sales/order');
        $order->loadByIncrementId($session->getLastRealOrderId());

        $standard->setOrder($order);

        try {
            $xml = $standard->getCheckoutForms();
        } catch (Exception $ex) {
            throw new Exception($ex);
        }

        $html = '<div class="block" style="padding: 10px; background-color: white;">';
        $html .= "<h1>{$this->__('Valitse verkkopankki')}</h1>";
        foreach($xml->payments->payment->banks as $bankX)
        {
            foreach($bankX as $bank)
            {
                $html .= "<div style='float: left; margin-right: 20px; min-height: 100px; text-align: center;'><form action='{$bank['url']}' method='post'><p>\n";
                foreach($bank as $key => $value)
                {
                    $html .= "<input type='hidden' name='{$key}' value='{$value}' />\n";
                }
                $html .= "<input type='image' src='{$bank['icon']}' /></p></form></div>\n";
            }
        }
        $html .= '<div style="clear: both;"></div></div>';
        return $html;
    }
}
```

KUVA 8 Redirect.php-tiedostosta palautettavat verkkomaksuvaihtoehdot



KUVA 9 Maksupalvelun tarjoamat verkkomaksuvaihtoehdot

Verkkomaksupalvelusta paluu takaisin verkkokauppaan ja mahdolliset maksun keskeytykset toteutettiin controllerin *responseAction()*-funktioon. Funktiossa käytetään osaa samoista rajapintatiedoista kuin maksua tehtäessä. Palvelun palvelin tarkistaa turvatarkisteen, jolloin saada varmistettua viestin oikeellisuus. Jos turvatarkiste on oikea, ohjataan käyttäjä maksun tilaa kuvaavalle sivulle. Maksun ollessa suoritettu ohjataan käyttäjä tilauksen *success.phtml*-sivulle. Maksun keskeytyessä ohjataan ostoskoriin. Verkkomaksujen täydellisen toiminnallisuuden kuvaaminen jätettiin pois tietoturvallisuuden vuoksi.

5.4 Ajanvarauskalenteri

Ajanvarauskalenteri toteutettiin PHP-, jQuery- ja Ajax-tekniikoilla.

Ajanvarauskalenterin pohjaksi luotiin *calendar.php*-tiedosto. Tiedostoon liitettiin tarvittavat jQuery- ja css-tiedostot (KUVA 10). Pohjaan määriteltiin div-elementit kalenterille ja kalenteriajan varausikkunalle, tiedot-ikkuna, jossa on ohjeet kalenterin toiminnolle. *Varauskalenteri.js* tiedostoa käytetään näyttämään elementtien sisällöt ja se ohjaa *jquery.weekcalendar.js* tiedostoa, jossa varsinaiset kalenterin toiminnot käsitellään.


```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>

<link rel='stylesheet' type='text/css' href='reset.css' />
<link rel='stylesheet' type='text/css' href='http://ajax.googleapis.com/ajax/libs/jqueryui/1.7.2/themes/start/jquery-ui.css' />
<link rel='stylesheet' type='text/css' href='jquery.weekcalendar.css' />
<link rel='stylesheet' type='text/css' href='varauskalenteri.css' />

<script type='text/javascript' src='http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js'></script>
<script type='text/javascript' src='http://ajax.googleapis.com/ajax/libs/jqueryui/1.7.2/jquery-ui.min.js'></script>
<script type='text/javascript' src='jquery.weekcalendar.js'></script>
<script type='text/javascript' src='varauskalenteri.js'></script>

</head>

```

KUVA 10 Ajanvarauskalenterin jQuery- ja css-tiedostot liitettyinä calendar.php-tiedostoon.

Ajanvarauskalenterille luotiin oma tietokantataulu sovelluksen tietokantaan (KUVA 11), johon varaukset tallennetaan. Tauluun luotiin yksilöidyt tunnustekentät varaukselle ja varaajalle. Tauluun lisättiin kentät varaukseen tarvittaville tiedoille, jotka olivat varauksen päivämäärä, aloitus- ja lopetusajat sekä otsikko ja huomio-osio.

meeting_reservations	
PK	ID
	userID date starttime endtime title body

KUVA 11 Ajanvarauksen tietokantataulu

Ajanvarauskalenteri päätettiin liittää maksuosioon verkkokaupan puolelle niin, että tilauksen tekijä voi halutessaan varata palaveriajan tilausta tehdessään (KUVA 12). Ajanvarauskalenterin painike sijoitettiin sovelluksen valmiina olevaan kassatapahtumaan. Kassatapahtumasta korvattiin tilauksen toimitustapaosio ajanvarauskalenterilla. Muutokset olivat pääsääntöisesti toimitustapaosion toiminnallisuuksien poistamista, mitkä korvattiin yksinkertaisesti lisäämällä kyseinen ajanvarauskalenterin painike.

Onnistamo Tervetuloa, Jarkko Tissari | Asiakastilini | Toiveilista | Ostoskorini (1 tuote) | Kassa | Kirjautu ulos | Etsi kaupasta... | HAE

TUOTEKEHITYS ▾ MYNNIN KEHITYS ▾ KESTÄVÄ KEHITYS ▾

Kassa

- Tilaajan tiedot MUOKKAA
- Vastaanottajan tiedot MUOKKAA
- Toimitustapa**
- Maksun tiedot
- Tarkastele tilausta

Palaverin varaaminen
0,00 €

[← Takaisin](#)

TILAUKSEN ETENEMINEN

LASKUTUSOSOITE |

Jarkko Tissari
Jesdass
Kuopio, 70100
Suomi
Puh. 2345678

TOIMITUSOSOITE |

Jarkko Tissari
Jesdass
Kuopio, 70100
Suomi
Puh. 2345678

PALAVERIAIKA |

Pvm: 2011-05-05 | 11:00 - 12:00

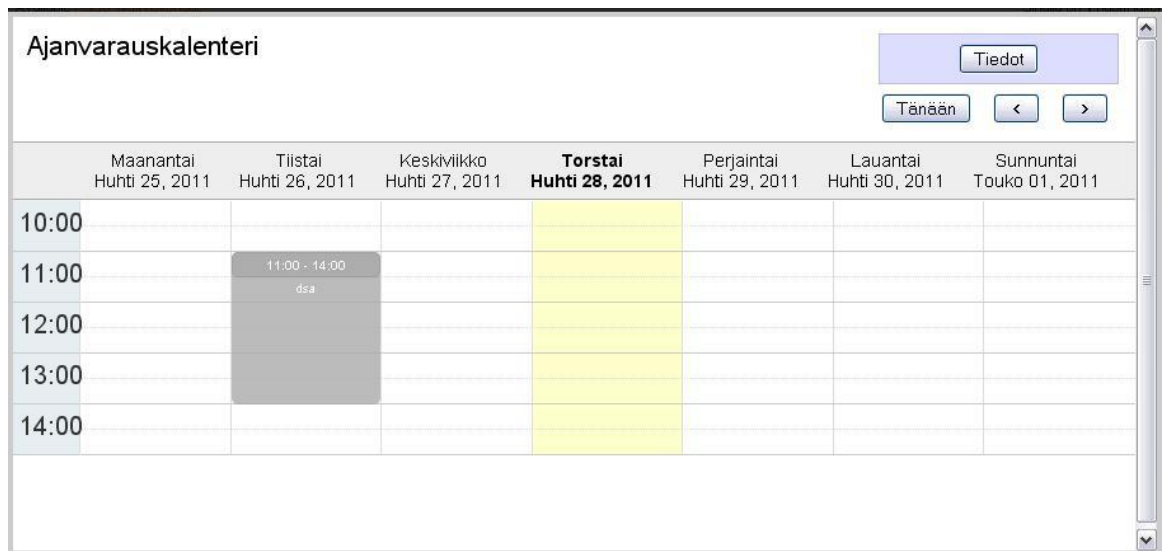
MAKSUTAPA

Tietoa meistä | Asiakaspalvelu | Sivukartta | Hakusanat | Laajennettu haku | Ota yhteyttä
Onnistamo Oy - **Katso kartalta** Väinämöisenkatu 3, 70500 Kuopio
© 2011 Jarkko Tissari. All Rights Reserved.

KUVA 12 Kassatoiminnossa olevan ajanvarauskalenterin avaamiseen tarkoitettu nappi

Kalenteri avautuu kassatoiminnossa olevan *Varaa aika* napin painalluksesta. Avautuva kalenteri toteutettiin jQuery domwindow liitännäisen avulla, jolloin verkkokauppaan tuli lisätä tarvittavat jQuery kirjastot ja avaamiseen tarvittava koodi. Kirjastot lisättiin aiemmin mainittuun *page.xml* tiedostoon (KUVA 3). Toiminnan varmistamiseksi kutsuttiin *jQuery.noConflict()*-funktioita, etteivät jQuery-kirjastot sekaantuisi keskenään.

Kalenterin pääikkunassa (KUVA 13) ovat kuluva viikon päivät ja kelloajat, joille ajan voi varata. Pääikkunaan tehtiin myös navigointinapit, joilla käyttäjä pääsee selamaan viikonpäiviä sekä tietoja kalenterin toiminnasta. Kalenterin toimintaperiaate tehtiin käyttäjystävälliseksi ja yksinkertaiseksi. Varauksien haku kalenteriin toteutettiin Ajax tekniikalla, jonka avulla haettiin varaukset tietokannasta taulukkoon (KUVA 14). Taulukko haettiin json-muotoon, koska siten saatiin helpommin käsiteltävä taulukkoa jQuery tiedossa.



KUVA 13 Ajanvarauskalenterin pääikkuna

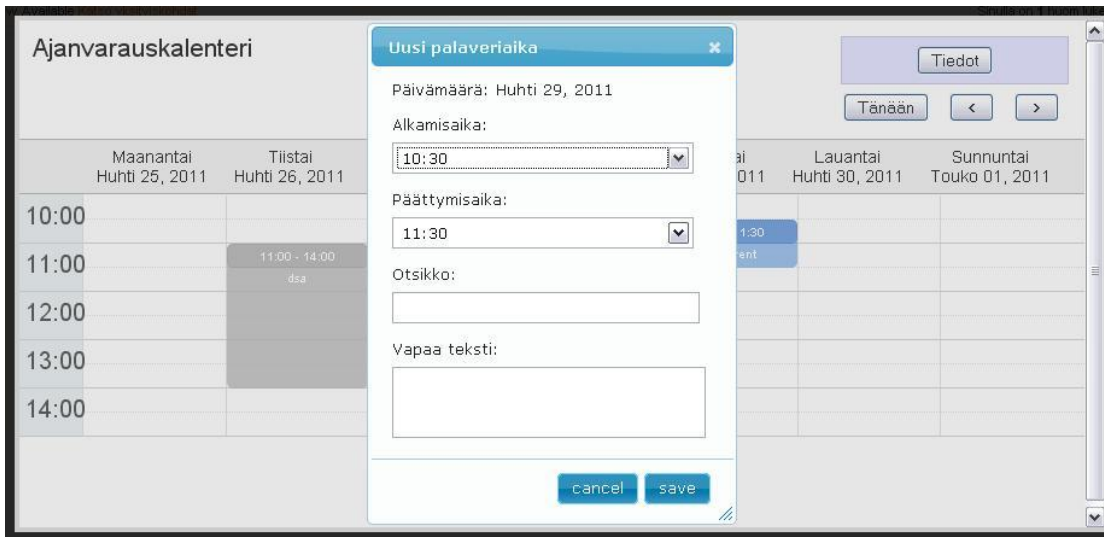
```
// varauskalenteri.js
function getEventData() {
    var year = new Date().getFullYear();
    var month = new Date().getMonth();
    var day = new Date().getDate();

    var meets = [];
    var events = [];

    $.ajax({
        type: "POST",
        traditional: true,
        url: "meetings.php",
        async: false,
        data: meets,
        dataType: "json",
        success: function(data) {
            meets = data;
        }
    });
}
```

KUVA 14 Varattujen aikojen haku tietokannasta json taulukkoon Ajax tekniikka käytäen.

Käyttäjän painaessa haluttua päivää ja kelloaikaa avautuu varausikkuna (KUVA 15), jossa käyttävä voi antaa tarkemmat palaverin aloitus- ja päättymiskelloajat, sekä otsikon ja vapaan tekstin, esimerkiksi huomautuksia. Palaveriajan tallentaminen tietokantaan toteutettiin Ajax-tekniikalla. Ajax-funktiolle annettiin parametreina kutsuttava tiedosto, joka sisältää tallennukseen tarvittavat tietokanta lauseet. Parametreina annettiin myös tarpeelliset tiedot (KUVA 16). Näitä tietoja ovat varattavan ajan päivämäärä, aloitus- ja lopetusajat, otsikko, huomiot ja käyttäjän yksilöity tunnistenumero (ID).



KUVA 15 Ajanvarauskalenterin varausikkuna

```

*/
_updateEventInCalendar : function (calEvent) {
    var self = this;
    var options = this.options;
    self._cleanEvent(calEvent);

    ...

    $.get("sql.php", {
        func: "varaus",
        date: year + "-" + month + "-" + day,
        start: calEvent.start.getHours() + ":" + startminutes,
        end: calEvent.end.getHours() + ":" + endminutes,
        title: calEvent.title,
        body: calEvent.body,
        id: calEvent.id,
        uid: this.options.customerID
    });

    ...
},

```

KUVA 16 Varausten lisäksi tarvittavien tietojen ja tietokantaan tallennukseen käytettävän tiedoston parametrit ovat määritelty funktion.

Palaverinajan muokkaus toteutettiin myös samaa funktiota käyttäen, mutta erona oli kutsuttavan php -tiedoston käsittelyosion logiikka. Muokkaamisessa tarvitaan myös parametrina käytettävää kalenteriajan yksilöityä ID:tä. Aikojen muokkauksessa tuli ottaa huomioon, että muut käyttäjät eivät pääse vaihtamaan toisten käyttäjien varattuja aikoja. Tämä ratkaistiin lisäämällä kalenterin ominaisuuksiin *readonly* parametri, jolla muokattavuus yksilöidään.

Varauksen poistamiseen käytetty toiminto on yksinkertaisempi ja eroaa lisäksi siitä ja muokkaamisesta siten, että tarvittavia parametreja ei tarvita kuin kaksi, jotka viittaavat varaukseen ja varauksen poistajaan. Molemmat tunnistetaan omilla ID numeroilla.

Hallintapaneelin ajanvarauskalenteri toteutettiin samalla pohjalla kuin kaupan puolella oleva kalenteri ja ne ovat käyttöliittymältään samanlaiset. Kalenterin ominaisuuksista poistettiin *readonly* parametri, jolloin ylläpitäjät pääsevät muuttamaan asiakkaiden tietoja tarpeen vaatiessa. Muutoksia tehtiin myös tietokantaa käsitteleviin tiedostoihin, jolloin ylläpitäjät voivat vapaasti lisätä, muokata ja poistaa asiakkaiden aikoja. Hallintapaneelin ajanvarauskalenterin tiedostojen sijainti on eri paikassa kuin varsinaisen kaupan puolen kalenterin tiedostot ja tietoturvallisuuden vuoksi kalenterin linkitys tehtiin hallintapuolen tiedostoihin. Hallintapaneelissa kalenterin avaaminen sijoitettiin menu-valikkoon (KUVA 17).



KUVA 17 Hallintapaneelin menu-valikkoon sijoitettu linkki ajanvarauskalenterille

6 TESTAUS

Koska projektin aikataulu pitkittyi, ei varsinaista järjestelmätestausta toteutettu täydellisesti. Testaussuunnitelmia ei tehty, koska sovellusta teki vain yksi henkilö. Sovellusta ja sen osia testattiin jatkuvasti uusien toimintojen ja muutoksien valmistuessa. Eniten testattu osio oli ajanvarauskalenterin toiminnot, joita testattiin monipuolisesti erilaisilla tilanteilla. Kun virhetilanteita esiintyi, ne korjattiin heti tai mahdollisimman pian havainnosta. Yleisimmät virheet olivat virheelliset tietokantalauseet ja aikojen päällekkäisyydet.

Verkkomaksujen testaus toteutettiin siihen tarkoitetulla testauspalvelimella, johon otettiin yhteys testaustunnuksilla. Kun todettiin verkkomaksuliikenteen avautuvan verkkokaupasta verkkomaksupalveluun ja tilauksen pystyi maksamaan testausmielessä, päätettiin maksumoduulin toimivuuden onnistuneen. Löytyneet virheet liittyivät yleensä virheellisiin tietokannan hakulauseisiin tai matemaattisiin laskutoimituksiin.

Sovellusta testasi myös työn tilaaja, jonka mukaan alustan valinta osoittautui onnistuneeksi ja prototyypiksi kaupan toteutus vaikuttaa erittäin monipuoliselta ja kehitysmahdollisuuksia on avoinna moneen suuntaan. Ajanvarauskalenteri sai myös myönteistä palautetta osakseen, koska tilaajan mukaan kalenteriominaisuus on houkutteleva ja hän haluaa ottaa sen ensimmäisenä käytäntöön.

Koska testauksen suhteellisen vähän ja sovelluksen jalostaminen ei ole tällä hetkellä tärkeimpien kehityskohteiden listalla prototyypistä julkaisuversioon asti, päätettiin käyttöönottoa siirtää myöhempään vaiheeseen. Sovelluksen kehittämistä tullaan kuitenkin jatkamaan resurssien vapauduttua vähintäänkin siten, että kauppa integroidaan Onnistamo Oy:n nettisivuihin senhetkisen tuote- ja palvelutarjonnan mukaan.

7 POHDINTA

Opinnäytetyön tekeminen onnistui tavoitteiden mukaisesti, vaikka alun suunnitelmat muuttuivat, eikä varsinaisia määrittelydokumentteja tehty. Sovellukseen saatiin toteutettua halutut ominaisuudet ja toiminnallisuudet. Opinnäytetyötä tehdessä oppi dokumentoinnin tärkeyden ja Magento verkkokaupparatkaisun toimintaperiaatteet sekä sen rakennetta.

Isoimpana onnistumisena voidaan pitää projektia kokonaisuudessaan, koska Magento alustana ja ajanvarauskalenteriin käytetyt tekniikat olivat uusia. Verkkomaksujen ohjelmointi oli myös kokonaan uusi alue. Alustan valinta oli onnistunut, koska se on hyvin ja kohtuullisen helposti muokattavissa oleva verkkokaupparatkaisu.

Ensimmäiset haasteet ilmenivät jo alustan pystyttämisessä, koska testipalvelinta käytettiin Windows-järjestelmässä, ja Magento on suunniteltu Linux-järjestelmissä käytettäväksi. Toiseksi haasteeksi voisi mainita suunnitelmien muutokset, jotka hidastivat varsinaisen projektin etenemistä. Ensimmäinen suunnitelma verkkokaupprojektissa ei saanut tuulta alleen, joten verkkokaupan määrittelyt vaihdettiin. Työmäärä, joka tehtiin ennen suunnitelmaa, oli kuitenkin opettavainen Magento ohjelmistoratkaisun toiminnasta ja rakenteesta.

Haasteita olivat myös uudet tekniikat, jotka tuli opetella lähes nollasta. Ajanvarauskalenterin käyttöliittymään ja toimintoihin opeteltiin jQuery-tekniikan aivan alusta. Toisena tekniikkana käytettiin Ajaxia, josta oli hieman aiempaa kokemusta. jQuery ongelmien kanssa taisteltiin viikkoja. Yksi syy ongelmiin oli uusi ohjelmointitekniikka ja jQuery-kirjastojen järjestykset, joka sekoitti kirjastojen toimintaa keskenään.

Verkkomaksujen toteutus oli aluksi haastavaa, koska tiedon saanti oli vähäistä. Muutamien työtuntien jälkeen saatiin apua ottamalla yhteyttä pankkeihin ja verkkomaksuihin perehtyneisiin yrityksiin, ja heiltä saatiin tarvittavat rajapintakuvaukset verkkomaksuille.

Tavoitteet täyttyivät projektin osalta lähtötilanteen toivomusten mukaisesti. Projektin edetessä tuli uusia kehitettäviä toimintoja, jotka myös saatiin valmiiksi. Uutena toimintona voisi mainita verkkomaksut suomalaisiin pankkeihin.

Omina tavoitteina oli projektin läpivieminen alusta loppuun sovitussa ajassa ja kaikkien vaadittujen toimintojen tekeminen. Tavoitteet saavutettiin ja lisäksi tuli oppia jQuery-, Ajax-, XML-tekniikoista paljon lisää.

Sovelluksen jalostamista jatketaan resurssien vapauduttua vähintäänkin siten, että kauppa integroidaan Onnistamo Oy:n nettisivuihin sen hetkisen tuote- ja palvelutarjonnan mukaan. Kuitenkin ennen sovelluksen julkaisemista on maksujärjestelmälle ja ajanvarauskalenterille suoritettava kattava testaus. Ennen julkaisemista ulkoasumuutoksia saatetaan tehdä tarvittaessa.

Opinnäytetyön läpivienti oli opettava ja samalla sopivan haasteellinen. Suunnitelmat ja määrittelyt olisi voitu tehdä paremmin, jolloin ei jäisi suuria aukkoja toteutukselle. Siten myös jatkokehityksen kannalta olisi valmiina hyvät alkuperäiset suunnitelmat dokumentoitu.

LÄHTEET

Aarne 2007. *Magento forum: oscommerce -> magento, kokemuksia?* [verkkajulkaisu]. [viitattu 30.04.2011] Saatavissa:
<http://www.magentocommerce.com/boards/viewthread/16767/>

Arhi, I, 2011. *Joomla sisältöhallinta ohje* [verkkajulkaisu]. [viitattu 12.04.2011] Saatavissa:
<http://www.ohje.info/content/category/9/16/87/lang,finnish/>

CERT-FI 2009. *Tietoturva nyt!: Korjaus osCommercea koskevaan artikkeliin* [verkkajulkaisu]. [viitattu 30.04.2011] Saatavissa:
<http://www.cert.fi/tietoturvanyt/2009/06/ttn200906041906.html>

Chaffer, J. 2009. *Learning jQuery 1.3*. Birmingham UK. Packt Publishing Ltd.

Gilmore, W. J. 2005. *PHP ja MySQL: tehokas hallinta*. Jyväskylä. Gummerus Kirjapaino Oy.

Heinisuo, R. 2004. *PHP ja MySQL: tietokantapohjaiset verkkopalvelut*. Jyväskylä. Gummerus Kirjapaino Oy.

Huttunen, J. 2010. *Onnistamo Oy* [verkkajulkaisu]. [viitattu 05.04.2011]. Saatavissa:
http://onnistamo.fi/Onnistamo_Oy.html

Joomlaportal 2007. *Joomlaportal: Mikä on Joomla?* [verkkajulkaisu]. [viitattu 30.04.2011] Saatavissa:
<http://www.joomlaportal.fi/content/view/93/39/>

Negrino, T. Smith, D. 2007. *JavaScript: Tehokas hallinta*. Jyväskylä: Gummerus Kirjapaino Oy.

Nykänen, O. 2001. *XML ToolKit*. Jyväskylä: Tummavuoren kirjapaino.

Peltomäki, J. & Nykänen, O. 2006. *Web-selainohjelmointi*. Jyväskylä: Docento.

Ponce de Leon, H. 2011. *osCommerce: osCommerce Online Merchant* [verkkajulkaisu]. [viitattu 12.04.2011] Saatavissa:
<http://www.oscommerce.com/solutions/oscommerce>

Sarja, J. 2010. *Av-viestinnän keittokirja: WAMP* [verkkajulkaisu]. [viitattu 05.04.2011]. Saatavissa:
<http://www.avkymppi.net/joomla/wamp.html>

Sarja, J. 2010. *Av-viestinnän keittokirja: CSS* [verkkajulkaisu]. [viitattu 05.04.2011]. Saatavissa:
<http://www.avkymppi.net/joomla/html/css.html>

Tilastokeskus, 2010. *Tieto- ja viestintätekniikan käyttö: Jo joka toinen suomalainen käyttää Internetiä useasti päivässä* [verkkajulkaisu]. [viitattu 23.04.2011] Saatavissa:
http://tilastokeskus.fi/til/sutivi/2010/sutivi_2010_2010-10-26_tie_001_fi.html

Varien, 2011. *Product features: Magento Features* [verkkajulkaisu]. [viitattu 12.04.2011] Saatavissa:
<http://www.magentocommerce.com/product/features/>

Walker, R. 2009. *VirtueMart 1.1 User Manual* [verkkajulkaisu]. [viitattu 30.04.2011] Saatavissa:
http://virtuemart.net/documentation/User_Manual/Introduction.html

www.savonia.fi

