



LAUREA
AMMATTIKORKEAKOULU

Uuden edellä

Web CRM -järjestelmän toteuttaminen Lemonsoft-rajapintaan

Laatu, Ville

2013 Kerava

Laurea-ammattikorkeakoulu
Kerava

Web CRM -järjestelmän toteuttaminen Lemonsoft- rajapintaan

Ville Laatu
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Joulukuu, 2013

Ville, Laatu

Web CRM-järjestelmän toteuttaminen Lemonsoft-rajapintaan

Vuosi 2013 Sivumäärä 35

Opinnäytetyön tavoitteena oli kehittää web-pohjainen asiakkuuden- ja tuotehallinnan mahdollistava järjestelmä. Järjestelmän tuli käyttää tietokantana yrityksen toiminnanohjausjärjestelmän, Lemonsoftin tietokantaa. Toteutuksessa tietoliikenteen saamiseksi järjestelmien välille, käytettiin Lemonsoftin rajapintaa. Opinnäytetyössä kerrotaan, kuinka rajapinta asennetaan ja luodaan kehitettävään järjestelmään, sekä käydään läpi mitä ohjelmointikieliä ja toteutustekniikoita järjestelmää kehittäessä on käytetty.

Järjestelmä kehitettiin Dieta Oy:lle, koska yrityksen johto oli etsinyt ratkaisua, millä keinoin välittää yrityksen pääkonttorissa pyörivän toiminnanohjausjärjestelmän tiedot kentälle myyjien saataville. Yritys on Suomen toiseksi suurin ammattikeittiölaitteiden ja -tarvikkeiden myyntiin erikoistunut yhtiö.

Järjestelmä toteutettiin monien vaiheiden jälkeen. Toteutus lähti liikkeelle asentamalla ja pystyttämällä virtuaalipalvelin, sekä asentamalla tarvittavat paketit ja ohjelmistot rajapinnan toimintaa varten. Järjestelmän ohjelmointi koostui monien eri ohjelmointikielien yhdistelmästä, jolla saatiin aikaiseksi web-pohjainen järjestelmä. Monet henkilöt olivat tuotekehityksessä mukana järjestelmätestaajan roolissa ja heidän palautteidensa avulla järjestelmää kehitettiin eteenpäin aina järjestelmän julkaisuun asti. Ensimmäinen versio järjestelmästä julkaistiin elokuussa 2013, seitsemän kuukautta projektin aloituksen jälkeen. Julkaisun jälkeen järjestelmä on ollut yrityksen jokapäiväisessä liiketoiminnassa mukana ja järjestelmää käyttävät arviolta noin puolet yrityksen henkilöstöstä.

Opinnäytetyöstä valmistui Dieta Oy:lle asiakkuuden- ja tuotehallinnan mahdollistava järjestelmä. Kehitetty järjestelmä täytti kaikki johdon sille asettamat vaatimukset, lisäksi järjestelmä tarjosi käyttäjilleen kaikki heidän toivomat ominaisuudet. Järjestelmä koettiin heti julkaisun jälkeen hyvin hyödylliseksi työkaluksi koko konsernille.

Asiasanat: asiakkuudenhallinta, ohjelmointi, rajapinta.

Ville, Laatu

Developing a Web CRM software to Lemonsoft Web Services

Year	2013	Pages	35
------	------	-------	----

The main goal of this Bachelor's thesis was to develop a Web-based customer and product management system. The system has to use the company's enterprise resource planning, Lemonsoft's database. To achieve necessary communication between the systems, it was required to use web services that Lemonsoft offers. This thesis describes how to install and create web services to software that was being developed, as well as goes through the programming languages and implementation techniques that have been used while developing the system.

The system was developed to a company called Dieta Ltd , because the management of the company was searching a solution for how to convey the company's enterprise resource planning data to the field around Finland. The company is Finland's second largest professional kitchen equipment and material sales specialized company.

The system was developed after many different phases. The implementation started with installing a virtual server. First it was necessary to install appropriate repositories and software to achieve the functionality for the web services. The developed system was created by combining many different program languages. Many people were involved in product development as system testers. With the feedback they gave, the system was developed forward until the system was published. The first version of the system was released in August 2013, seven months after the project started. After the release of the system, it has been involved in the company's day to day business, and is used by approximately half of the company's employees.

As a result of this Bachelor's thesis developed a customer and product management system for Dieta Ltd. The developed system filled all the requirements that the company management had set. In addition the system offered all the attributes to users that they had hoped for. After the release the system was seen a very useful tool for the entire company.

Keywords: customer relationship management, programming, web services.

Sisällys

1	Johdanto	6
1.1	Tutkimuksen taustat ja tavoitteet	6
1.2	Tutkimuskysymykset	6
1.3	Keskeiset käsitteet	7
1.4	Tutkimusmenetelmä	8
1.5	Tutkimuksen lähdeaineisto.....	9
1.6	Tutkimuksen reliabiliteetti ja validiteetti	9
1.7	Tutkielman eteneminen	10
2	Toimeksiantajayritys	11
2.1	Dieta Oy	11
2.2	Kokonaisesitys yrityksen toiminnasta.....	12
2.3	Olemassa oleva tietojärjestelmä.....	12
2.4	Asiakkuudenhallinta organisaatiossa.....	13
2.5	Tavoitteet, joihin kehitettävällä asiakkuudenhallintajärjestelmällä pyritään	14
2.5.1	Asiakastapahtumien raportointi	14
2.5.2	Kentällä olevien myyjien tehokkuuden ja kommunikoinnin lisääminen.	14
2.6.3	Laiteriippumaton järjestelmä.....	15
3	Toteutuksessa käytetyt ohjelmointikielet	15
3.1	XML	15
3.2	SOAP.....	16
3.3	WSDL	16
3.4	PHP	17
3.5	Jquery mobile.....	18
3.6	Javascript, HTML ja CSS	19
3.7	SQL	19
4	Järjestelmäintegraatio	20
4.1	Tutkimuksen ERP-järjestelmän tarjoama rajapinta	20
4.2	Kehitettävän järjestelmän ja nykyisen ERP:n välinen tietoliikenne	21
4.3	Rajapinnan käyttöönottaminen.....	22
4.4	Rajapinnan käyttäminen	26
5	Tietoturva	27
6	Johtopäätökset ja Loppusanat	29
6.1	Johtopäätökset	29
6.2	Loppusanat	30

1 Johdanto

1.1 Tutkimuksen taustat ja tavoitteet

Tutkimuksen tavoitteena oli toteuttaa Dieta Oy:lle Web-pohjainen asiakkuudenhallintajärjestelmä. Tulevan järjestelmän tärkeimpänä tehtävänä oli tukea kentällä olevien myyjien kommunikointia ja mahdollistaa heille nopea ja helppo pääsy yrityksen toiminnanohjausjärjestelmän tietoihin. Yhtenä tärkeänä kriteerinä pidettiin myyjien raportointia toiminnastaan. Kehitettävän järjestelmän tulisi mahdollistaa myös raportointi ominaisuudet asiakastapahtumista. Myyjien toteuttamat raportit asiakaskäynneistä tai tapahtumista kirjataan järjestelmän avulla suoraan pääkonttorin toiminnanohjausjärjestelmään, mahdollistaen nopeamman toiminnan back-office puolella. Järjestelmä tarjoaa kirjattuja raportteja eri osa-alueiden myyjille luoden heille kilpailuetua. Kirjatuista raporteista he pystyvät tarkkailemaan, missä päin omaa aluetta voisi olla potentiaalisia asiakkaita, joiden kanssa tehdä kauppaa.

Kehitettävän järjestelmän tietokantana käytetään yrityksessä jo olevan Enterprise Resource Planning (ERP) tietokantaa. Tutkielmassa tullaan esittelemään, millä tavoin järjestelmää on kehitetty ja kuinka rajapintojen avulla voidaan luoda järjestelmäintegraatio kahden järjestelmän kesken.

1.2 Tutkimuskysymykset

Tutkimusta aloittaessa on hyvä pystyä ilmaisemaan tutkimuksessa mahdollisesti tulevat ongelmat, pää- ja ala-ongelmineen, kysymysmuodossa. Tämä antaa tutkijalle hyvät edellytykset tarkistaa tutkimuksen alussa ja lopussa, onko näihin kysymyksiin vastattu ja päästiinkö haluttuun lopputulokseen, joka tutkimuksen alussa asetettiin. (Saukkonen 2013.)

Tutkimuksen keskeisimmät kysymykset ja ongelmat ovat,

- Kuinka rajapinta voidaan toteuttaa Lemonsoft-järjestelmään?
- Kuinka voidaan kehittää asiakkuudenhallintajärjestelmä?

Opinnäytetyön ala-ongelmia ovat;

- Millä tavoin rajapinta asennetaan ja kuinka rajapinnan avulla voidaan toteuttaa järjestelmäintegraatio?
- Mitä välineitä asiakkuudenhallintajärjestelmän toteuttaminen ja pystyttäminen vaatii?

- Minkälaisia ohjelmointikieliä tai taitoja toteuttajalta vaaditaan?
- Kuinka suojataan järjestelmään pääsy ulkopuolisten henkilöiden osalta?

Tässä opinnäytetyössä on pyritty vastaamaan mahdollisimman helposti ja ymmärrettävästi tutkimuskysymyksiin ja ongelmiin tutkielman edetessä. Opinnäytetyössä tutkimuskysymyksiin tullaan vastaamaan esimerkkien, kuvien ja teoreettisen osuuden avulla. Tutkimuskysymysten vastausten lähdeaineisto on peräisin tutkimuksen toteuttajan kuvakaappauksista, omista pääteelmistä, sekä kokemuksista tutkimuksen toteutuksen tiimoilta. Teoreettista materiaalia on kerätty Internet- ja kirjallisuuslähteistä ja niiden avulla on pyritty helpottamaan tutkimuskysymysten ymmärtämistä.

1.3 Keskeiset käsitteet

CRM = Customer Relationship Management (Asiakkuudenhallintajärjestelmä)

ERP = Enterprise Resource Planning (Toiminnanohjausjärjestelmä)

PHP = Hypertext preprocessor (Palvelinpuolen ohjelmointikieli) (PHP 2013.)

XML = Extensible Markup Language (Tiedon rakenteellinen merkintäkieli) (w3 2013.)

SOAP = Simple Object Access Protocols (Tietoliikenneprotokolla, joka mahdollistaa proseduurien etäkutsuja) (w3 2013.)

JQM, JQuery Mobile = Javascript & CSS-kirjasto HTML 5-koodille (Jquery Mobile 2013.)

WSDL = Web Services Description Language (XML-pohjainen tietoverkkojen kuvauskieli) (w3 2013.)

HTML = HyperText Markup Language (Hypertekstin merkintäkieli, jonka avulla kuvataan pääosin nettisivujen muotoilua) (w3 2013.)

CSS = Cascada Style Sheets (Tyylikirjasto HTML-koodia varten) (w3 2013.)

SQL = Structured Query Language (Kyselykieli relaatiotietokantoja varten) (w3 2013.)

CSV = comma-separated values, voidaan käyttää myös nimellä character-separated values, koska erottajan ei tarvitse välttämättä olla aina pilkku. (Tekstiedosto tietokantoja varten,

jossa erotetaan tieto pilkuilla toisistaan ja jota voidaan käyttää tiedon siirtämisessä ja tallentamisessa) (Dominic 2010.)

IIS = Internet Information Services (Windows-pohjaisille palvelimille tarkoitettu web-palvelinohjelmisto, jonka avulla voidaan verkkosivut tai järjestelmät nostaa julkiseen verkkoon muiden saataville) (Microsoft 2013.)

1.4 Tutkimusmenetelmä

Tutkimusmenetelmäksi projektissa valittiin laadullinen eli kvalitatiivinen menetelmä ja tutkimus toteutettiin toimintatutkimuksena. Projektin suunnitteluvaiheessa kävi selväksi, että koko tutkimus jää minun hoidettavaksi. Oli myös selvää, että tässä tutkimuksessa täytyi luoda jotain konkreetista, joka vaikutti tutkimusmenetelmän valintaan. Näistä syistä menetelmäksi valikoitui toimintatutkimus. Toimintatutkimus on menetelmänä lähes poikkeuksetta jokaisessa tuotekehityksessä mukana. Toimintatutkimus nimensä mukaisesti vaatii työtä ja toimintaa toisin kuin tutkimukset, joissa perehdytään johonkin ilmiöön tai tapahtumaan, millä tavoin tai miten tutkittava kohde käyttäytyy erilaisissa tilanteissa. Tässä tutkimuksessa ei ollut mahdollisuutta tutkia, miten kehitetty järjestelmä vaikuttaa yrityksen työntekijöiden jokapäiväisessä työskentelyssä, sekä ajanpuutteen, että aiheen rajaamisen takia.

Tutkimusmenetelmän valintaan vaikutti lisäksi se, ettei tätä tutkimusta olisi voinut toteuttaa pelkän numeerisen datan avulla, niin jäi kvantitatiivinen eli määrällinen tutkimusmetodi hyvin vähäiselle käytölle tutkimusta toteuttaessa. Laadullisen toimintatutkimuksen etuna nähtiin se, miten aineistoa kerätään ja käsitellään. Toimintatutkimuksessa aineisto on paljon tiiviimmin yhdessä toisiinsa keräyksestä toteutukseen kuin kvantitatiivisessa menetelmässä. Tämä auttaa ja nopeuttaa tutkittavan kohteen kehittämistä ja ratkaisujen löytämistä. Laadullisen aineiston tarkoitus ei ole myöskään tarjota suoraa ratkaisua tutkittavaan kohteeseen, vaan antaa mahdollisuuksia ja näkemyksiä ymmärtää tutkittavaa kohdetta paremmin.

Menetelmänä toimintatutkimus on yksi kiistanalaisimmista tutkimusmetodeista. Toimintatutkimuksessa pyritään nimensä mukaisesti toiminnan runsauteen, jonka avulla saadaan muutoksia ja kehitystä aikaiseksi. Toiminnan runsauden alla on lukemattomia eri tutkintatyylejä, joilla kohdetta voidaan tutkia tarkemmin. Lukemattomien tutkintatapojensa johdosta toimintatutkimukset ovat todella harvoin samanlaisia. Toimintatutkimuksissa voivat erota toisistaan esityslistat, syklit, kohteeseen lähestymistapa ja tutkijan omat näkemykset tai tulkinnat tutkittavasta kohteesta. Toimintatutkimus pyrkii auttamaan tutkijaa täyttämään sekä tieteelliset ja käytännölliset ongelmat tutkimusta tehdessä, sekä auttamaan tutkijaa tekemään oikeita valintoja eri näkemysten ja vaihtoehtojen pohjalta. (Bonnet & MacIntosh 2007, 66.)

Toimintatutkimus hyödyntää yleensä dokumentointiin ja kohteen tutkimiseen etnografisen ja tapauskohtaisen tutkimusmenetelmän tekniikoita.

Useissa tapauksissa toimintatutkimus määritellään seitsemään eri vaiheeseen:

1. Kohdistetaan painopiste nopeasti saatavilla olevaan materiaaliin.
2. Kerätään tietoa erilaisista lähteistä ja käytetään etnografisia ja tapauskohtaisen tutkimuksen tekniikoita materiaalin keräämiseen.
3. Analysoidaan, dokumentoidaan ja tarkistetaan dokumentoitu tieto välittömästi uudelleen.
4. Kehitetään ja toteutetaan tulkitsevia analyttisiä ryhmiä.
5. Järjestetään tieto ja tulkitut ryhmät tapauksiksi. Jaetaan ryhmät järjestelmällisiksi tapahtumiksi tai esineiksi, sekä mahdollisesti liitetään ne lopulta yhteen.
6. Aloitetaan tai jatketaan tutkittavan kohteen kehittämistä lyhyen ja pitkän -aikavälin toimintojen osalta.
7. Aloitetaan sykli alusta. (Freebody 2006, 85-86.)

1.5 Tutkimuksen lähdeaineisto

Tässä tutkimuksessa materiaalia kerättiin vastaavalla tavalla kuten kirjassa ”Qualitative Research Design for Software Users” mainitaan: ”Ensin luetaan ja opetellaan tutkimusalueen aiheita ja otsikoita”. (Davidson & Di Gregorio 2008, 37) Ensi kosketuksen jälkeen aiheeseen alkoi IT-infrakstuuri selkiytyä. Alkuaineisto, jota kerättiin juuri edellä mainitun kirjan tavoin, auttoi hahmottamaan miten ja millä välineillä tutkimus voitaisiin toteuttaa. Aineistona käytettiin, sekä sähköistä, että kirjallista materiaalia. Saadun materiaalin avulla oli mahdollista aloittaa toimivan Customer Relationship Management (CRM) järjestelmän kehittäminen. Tutkimusvaiheessa materiaalia kerättiin etenkin palvelurajapintoihin liittyen. Suurin osa kerätystä aineistosta oli tästä syystä sähköistä ja siihen perehdyttiin paljon tarkemmin. Aineiston tarkempaa tutkimista tietyillä osa-alueilla tarvittiin, koska tutkimuksen alussa etenkin palvelurajapinta ja rajapinnat olivat hyvin vieraita tutkimuksen toteuttajalle. Sähköisen aineiston etuna huomattiin heti tutkimuksen alussa, että se osoittautui myös helpoimmaksi ja nopeimmaksi tavaksi saada tutkimuksen toteuttamiseen tarvittavaa tietoa.

1.6 Tutkimuksen reliabiliteetti ja validiteetti

Tutkimuksen reliabiliteetilla tarkoitetaan tutkimuksen luotettavuutta. Tämä tarkoittaa tutkimuksen mahdollista uudelleen toteuttamista toisen tutkijan tekemänä. Jotta tutkimus olisi mahdollisimman luotettava, tulisi toisen tutkijan päästä lähes samaan lopputulokseen kuin alkuperäisen tutkimuksen toteuttajan. (Hirsjärvi, Remes & Sajavaara 2009, 231–233.)

Tämän tutkimuksen kohdalla reliabiliteettia voi olla hieman hankala mitata, koska tutkimus on toteutettu toimintatutkimuksena, tehden lukuisia erilaisia toimintasyklejä, joita ei voida millään mahduttaa mukaan tutkielmaan. Tutkimuksen toistettavuus on myös hieman haasteellista, varsinkin vuosien päästä, koska siihen vaikuttivat tutkimuksen aikaiset versiot ohjelmointikielistä ja yrityksen järjestelmistä. Toteuttajalla oli myös lähes täydet valtuudet toteuttaa järjestelmän ulkoasu kokonaisuudessaan, sekä käyttäjä vuorovaikutteisuuden tarvittavien toimintojen rakentaminen parhaaksi katsomallaan tavalla. Luotettavuutta parantava seikka on se, että vaatimusmäärittelyt olisivat toiselle tutkijalle lähes vastaavat, jolloin lopputulos työssä voisi mahdollisesti olla toiminnallisesti ja tiedonsiirtoon vaikuttavien tekijöiden osalta lähellä tätä tutkimusta.

Tutkimuksen validiteetti tarkoittaa tutkimuksen pätevyyttä. Validiteetilla pyritään tutkimuksen totuuden tavoitteluun. Tutkimuksen valmistuttua tulisi kaikkien sen väittämien ja päätelmien olla oikeita. Tutkimuksessa ei tulisi ilmetä virheitä, kuten väärinkäsityksiä yleisistä näkemyksistä tai periaatteista. Tutkimuksessa tulisi välttää tutkimusaiheeseen liittyviä täysin vääriä ja epäolennaisia kysymyksiä. Laadullisessa tutkimuksen pätevyyttä ja oikeaoppisuutta on kuitenkin hieman vaikeampi mitata kuin määrällisen, jossa tulokset yleensä esitetään luotettavassa numeraalisessa viitekehyksessä. Laadullisen tutkimuksen toteuttajan tulisi tutkimustaan esittäessä olla uskottava ja vakuuttava, sekä luoda lukijalle mahdollisimman ymmärrettävä ja tarkka kuva tutkimuksesta, johon lukija voi myös samaistua. (Saaranen-Kauppinen & Puusniekka 2006.)

Tämä tutkimus toteutettiin laadullisena toimintatutkimuksena, joten sen validiteetin todentaminen oikeaksi tai vääräksi on hieman hankalaa. Tutkielmassa on kuitenkin pyritty saamaan mahdollisimman tarkka kuvaus tutkimuksesta, kuten millä tavoin se on toteutettu ja mitä asioita tulee huomioida toteuttaessa vastaavanlaista projektia. Tutkimuksen validiteettia on yritetty parantaa antamalla tutkielma luettavaksi täysin aiheesta tietämättömille henkilöille ja pyydetty heitä kommentoimaan epäselviä asioita. Saatujen kommenttien avulla tutkielman kuvaustapaa on pyritty parantamaan helpommin ymmärrettäväksi tekstin lukijoille. Tutkimuksen validiteetti toteutui kuitenkin siltä osin, että tutkimuksessa kehitetty WebCRM-järjestelmä on mukana yrityksen jokapäiväisessä liiketoiminnassa.

1.7 Tutkielman eteneminen

Tutkielma lähtee liikkeelle toimeksiantajayrityksestä, jossa kerrotaan yrityksestä tarkemmin. Kuvauksessa käydään läpi yrityksen toimivat tietojärjestelmät ja millainen vaikutus asiakkaidenhallinnalla on yrityksessä. Ensimmäisessä osiossa kerrotaan myös tavoitteista, johon tulevalle järjestelmällä pyritään.

Tämän jälkeen tutkielmassa kerrotaan toteutuksessa käytetyistä ohjelmointikielistä. Perehtyessä eri ohjelmointikieliin annetaan joistakin ohjelmointikielistä yksinkertaisia esimerkkejä lukijalle ja selitetään tarkemmin mihin mikäkin ohjelmointikieli vaikuttaa. Kahdesta järjestelmän pääkielestä, Hypertext preprocessor:sta (PHP) ja JQuery Mobilesta, on mukana hieman suurempi katsaus. Pääohjelmointikielistä PHP vastaa palvelinpuolesta hoitaen myös järjestelmäintegraation. Palvelinpuolen ohjelmointi tarkoittaa tiedon liikkumista sovellusten ja tietokantojen välillä. Toinen pääohjelmointikieli JQuery Mobile, taas vastaa lähes kokonaan järjestelmän visuaalisesta puolesta, sisältäen ulkoasun ja tehden sivustosta vuorovaikutteisen ja toiminnallisesti helppokäyttöisen järjestelmän käyttäjälle.

Seuraavaksi tutkielmassa käydään läpi järjestelmäintegraatiota. Kerrotaan miten se toimii ja mikä järjestelmäintegraatio ylipäättensä on. Osiossa käydään myös läpi, millaisia ovat tutkimuksessa käytetyn ERP-järjestelmän tarjoamat rajapinnat ja millainen ERP-järjestelmä on kyseessä. Tämän jälkeen käydään läpi rajapinnan määrittämistä, asentamista ja sen toimintaa. Perehdytään tarkemmin, mitä tietoa ERP ja kehitettävän järjestelmän välissä todellisuudessa liikkuu. Näytetään esimerkkien kautta, kuinka kehitetty järjestelmä voidaan liittää Lemonsoft toiminnanohjausjärjestelmään, hyödyntäen sen tarjoamaa rajapintaa PHP + Simple Object Access Protocols (SOAP) + Web Services Description Language (WSDL)-tekniikalla.

Lopuksi vielä kerrotaan tietoturvasta. Osiossa käydään ohjelmoinnin kannalta olennaisia tietoturvariskejä läpi, kuten kuinka torjua StructuredQuery Language (SQL)-injektio hyökkäykset ja kuinka tärkeää on luoda jonkinlainen tunnistautumismekanismi järjestelmään, suojaten järjestelmää ulkopuolisilta henkilöiltä.

Tutkielma päättyy johtopäätöksiin projektista ja loppusanoihin sekä mietteisiin tutkimuksesta, siitä miten tutkimus eteni ja päästiinkö haluttuun lopputulokseen toteuttajan kannalta.

2 Toimeksiantajayritys

2.1 Dieta Oy

Dieta Oy on suomalainen yksityisomistuksessa oleva ammattikeittiölaitteiden ja astioiden toimittaja. Yritys maahantuo, markkinoi ja myy keittiökokonaisuuksia laitteista tarvikkeisiin. Asiakkaina toimivat muut yritykset, pääosin ammattikeittiöt, joista esimerkkinä kunnat ja ravintolat. Yrityksen ensisijaisena tavoitteena on saada myytyä kokonaisia keittiöprojekteja asiakkaille. Keittiöprojektien toteuttaminen lähtee liikkeelle mittojen mukaan tehdystä keittiösuunnitelmasta, ja jatkuu määräaikaishuoltojen muodossa läpi koko keittiön elinkaaren. Yrityksen tarkoituksena on olla ammattikeittiöprojektien ammattilainen.

Laitteiden ja astioiden lisäksi Dietalla on oma ammattikeittiöpatojen suunnitteluun, kehittämiseen ja valmistamiseen perustuva tytäryhtiö Dietatec, jonka Dieta Oy omistaa sataprosenttisesti. Ammattikeittiöpatojen valmistaminen oli vuonna 1993 perustetun Dieta Oy:n peruslähdekohtana. Dieta sai alkunsa visioistaan valmistaa maailman parhaita patoja. Periaate laadusta ja edelläkävijän asemasta on ammattikeittiöpatojen osalta säilynyt markkinoilla noista ajoista asti. Yritys jälleenmyy vain tuotteita, joiden laadusta se voi olla varma. Laadun täytyy olla maailman huippua päästäkseen Dieta Oy:n myyntilistalle.

Dieta Oy valmistaa myös ruostumattomasta teräksestä (rst) erilaisia räätälöityjä kalusteita asiakkailleen. Rst-kalusteet valmistetaan konserniin kuuluvalla tehtaalla Virossa. Padat valmistetaan Dietan tytäryhtiö, Dietatecin omalla tehtaalla Helsingin Herttoniemessä. Yrityksen omistaa neljä osakasta, jotka kaikki ovat kokopäiväisesti sitoutuneita ja toimivat konsernin johtotehtävissä. Pääkonttori sijaitsee Helsingissä, Roihupellossa, jossa työskentelee suurin osa henkilöstöstä. Myös Venäjällä sijaitsee Dietan myyntiedustusta.

Dieta konserniin kuuluvat: Dieta Oy, Coffeepalace Oy, Dietatec Oy, Dieta AS ja Assi AS. Koko konsernin liikevaihto on n. 35 milj. euroa ja koko konserni työllistää noin 140 henkilöä. Dieta Oy:n liikevaihto on n. 30 milj. euroa ja työllistää kokonaishenkilöstöstä vähän alle sata. Strategisesti tärkeimmät myytävät tuotteet ovat padat, uunit ja pesukoneet. Nämä tuotteet vastaavat noin 20 % kokonaisymyynnistä.

2.2 Kokonaisesitys yrityksen toiminnasta

Yrityksen liiketoiminta perustuu konsultoivaan myyntiin, jossa pitkällä aikavälillä myös asiakkaan menestyksellä on ratkaiseva merkitys. Tämän vuoksi asiakkaan sitouttaminen yritykseen on erittäin tärkeää. Yrityksen liikevaihto jakautuu kutakuinkin tasan julkisen ja yksityisen sektorin välillä painopiste kuitenkin hieman julkisen puolella. Dieta Oy tarjoaa ravitsemusalan toimijoille kokonaisvaltaisia ratkaisuja, joissa yrityksen mukaan toimivat niin suuret linjat kuin pienetkin yksityiskohdat.

2.3 Olemassa oleva tietojärjestelmä

Dieta Oy:llä on olemassa yksi kokonaisvaltainen järjestelmä, Lemonsoft. Koko yrityksen IT-infrakstuuri perustuu tähän toiminnanohjausjärjestelmään. Lemonsoft on toiminnanohjausjärjestelmä, joka on kehitetty auttamaan yrityksiä hallitsemaan ja kehittämään liiketoimintaansa. Se on ylivoimaisesti tärkein työkalu Dietalle. Lemonsoftin avulla pystytään seuraamaan, tallentamaan, muokkaamaan, hakemaan ja hallitsemaan koko yrityksen liiketoimintaa aina laskutuksesta asiakkuudenhallintaan asti. Lemonsoft tarjoaa Dietalle seuraavia ominaisuuksia asiakkuudenhallinnan lisäksi: myyntilaukset, tarjouslaskennat, projektihallinnan, palkanlas-

kennan, taloushallinnon, henkilöstöhallinnan ja varastohallinnan. Lemonsoft tarjoaa myös rajapinnan itseensä, joka mahdollistaa erilaisten järjestelmien kehittämisen ja integroinnin.

Kokoonsa nähden järjestelmä on yllättävän käyttäjäystävällinen ja suuria ongelmia sen kanssa ei ole kohdattu. Vain muutamien suurempien päivitysten yhteydessä on koettu pieniä ongelmia. Syy ongelmiin on ollut järjestelmän räätälöinti paremmin Dietan käyttöön soveltuvaksi. Yleispäivitysten myötä jotkin räätälöinnit eivät välttämättä ole toimineet toivotulla tavalla tai ne ovat yliajaneet päivityksen tuomat uudet ominaisuudet, jotka ovat mahdollisesti estäneet tarvittavien toimintojen toteuttamisen. Dietalle Lemonsoft järjestelmän räätälöinti alkoi jo järjestelmän ostovaiheessa. Dieta oli Lemonsoftin ensimmäisiä suuria asiakkaita ja tämän takia se tarjosi räätälöintiä järjestelmään, jotta se soveltuisi paremmin vastaamaan Dietan tarpeita. Räätälöinti esimerkkejä ovat esimerkiksi web-sivustolle mahdollistetut rajapintahaut ja tuotteiden muokkaus ominaisuudet sen osalta.

Järjestelmää käytetään koko konsernissa. Jokaiselle tytäryhtiölle on kuitenkin luotu oma tietokantansa. Tietokantoja ja koko järjestelmää ylläpidetään ja hallinnoidaan Dietan pääkonttorissa Roihupellossa. Dieta Oy:n oma tietokantapohja tuotiin alkujaan Lemonsoft järjestelmään vuonna 2006, jolloin järjestelmää vaihdettiin. Se oli peräisin vanhasta, vähän vastaavanlaisesta toiminnanohjausjärjestelmästä. Järjestelmävaihtoon päädyttiin, koska konsernin kasvaessa nähtiin, ettei sen aikainen Nova-järjestelmä enää tyydyttänyt isomman yrityksen tarpeita, joita kasvava yritys tarvitsi. Nova-järjestelmä oli ominaisuuksiltaan huomattavasti suppeampi ja osaa tiedoista ylläpidettiin ja laskentaa hoidettiin jopa Excel-taulukoiden avulla. Sen aikaiset Excel-taulukot eivät olleet edes mitenkään liitoksissa Novan tietokantaan.

2.4 Asiakkuudenhallinta organisaatiossa

Uutta web-pohjaista järjestelmää lähdettiin kehittämään, koska se nähtiin tärkeänä osana asiakkuudenhallintaa. Asiakkuudenhallintaa pidetään tärkeänä, koska ilman asiakkaita ei olisi koko yritystään. Tällä hetkellä asiakastietoja voi selata vain Lemonsoftin avulla, jossa on monia välilehtiä tyhjää tai turhaa tietoa. Uuden järjestelmän myötä yritetään tehostaa kentällä olevien myyjien työskentelyä asiakkaiden kanssa. Myyjät tulevat saamaan tämän jälkeen asiakastiedot haltuunsa mistäpäin Suomea vain. Järjestelmän myötä myyjän ei myöskään tarvitse raportoida tai kertoa kollegoilleen mahdollisista uusista myyntikohteista. Kaupan tekeminen tuotteiden, varaosien ja huollon myötä helpottuu, koska myyjä tulee näkemään tuotteiden varastossa olevat kumulatiiviset saldomäärät. Tuotesaldojen avulla myyjä voi tehdä tilauksen suoraan asiakkaalle luvaten tuotteelle varman toimituspäivän. Ostohistorian ja huoltohistorian perusteella myyjä voi myös ehdottaa asiakkaalle mahdollista lisämyyntiä jollekin jo aiemmin tilatulle tuotteelle, joka taas voisi helpottaa asiakkaan keittiötoimintaa huomattavasti.

tavasti. Järjestelmällä tavoitellaan asiakkuudenhallinnan parantamista myös myyjien osalta, joka on jäänyt hieman heikoksi vaikeaksi koetun tietojen saatavuuden vuoksi. Kaikkien uuden järjestelmän toimintojen myötä, myyjä pystyy toimimaan asiakkaan kanssa tehokkaammin tarjoten hänelle mahdollisimman hyvää palvelua. Asiakkaan saadessa hyvää ja nopeata palvelua voi asiakas tuntea olevansa tärkeä osa koko yritystä, joka taas voi johtaa pitkään yhteistyöhön osapuolten välille.

2.5 Tavoitteet, joihin kehitettävällä asiakkuudenhallintajärjestelmällä pyritään

Suurimmat tavoitteet kulmineituvat kolmeen kohtaan, joista ensimmäinen on asiakastapahtumien kirjaus. Tämän ominaisuuden avulla johto saa tietoa myyjien liikkeistä ja toiminnoista. Toisena tavoitteena on kentällä olevien myyjien lisääntyvä keskinäinen kommunikointi, joka taas johtaa tehokkaampaan asiakasrajapinta työskentelyyn ja viimeisenä tavoitteena on laiteriippumattomuus. Myyjä käyttää keskimäärin 70 % työajastaan oman toimistonsa ulkopuolella. Tämä vaatii järjestelmältä sopeutumiskykyä erinäisiin laitteisiin, kuten puhelimiin, tietokoneisiin ja tabletteihin.

2.5.1 Asiakastapahtumien raportointi

Myyjien tehdessä kauppvoja kentällä tulevat he uuden järjestelmän myötä saamaan helpotusta asiakastapahtumien raportointiin. Kirjatessaan myynti- tai asiakastapahtumia tallentuu raportti suoraan tietokantaan. Tallentuneista raporteista myös johto voi nopeasti nähdä miten kukin myyjä työskentelee. Tapahtuman tallennuttua tietokantaan, tapahtumassa näkyy kuka on myynyt, mitä on myynyt ja missä on myynyt. Johdolle on räätälöity oma näkymä sovellukseen kirjatusta raporteista, joita voidaan katsoa ja esittää graafisesti Qlikview nimisellä ohjelmalla.

2.5.2 Kentällä olevien myyjien tehokkuuden ja kommunikoinnin lisääminen

Myyjän raportoidessa asiakastapahtumaa hänelle annetaan mahdollisuus auttaa muiden osajalueiden myyjiä. Raportoiija voi antaa lisätietoja myyntiraportissaan, kuten onko kyseinen asiakas, jonka kanssa myyjä on tehnyt kauppaa mahdollisesti kiinnostunut myös muista Dieta Oy:n myymistä tuotteista. Myyjät luokitellaan Dietalla laite-, astia- ja tarvike-myyjiin. Tästä syystä on hyvin tärkeätä saada tietoa mahdollisista muistakin myyntiosa-alueista muille myyjille ja johdolle.

Uuden järjestelmän myötä myyjillä on mahdollisuus jättää täyttämättä tarvittavat asiakas-kentät raportista, jos kyseinen asiakas on jo ennalta ollut kirjattu Dietan järjestelmiin. Pelkkä asiakasnumeron lisääminen raporttiin riittää, jotta järjestelmä voi taustalla hakea numeron perusteella tarvittavat tiedot.

Asiakkaan tehdessä kauppvoja ensimmäistä kertaa Dietan kanssa myyjä täyttää asiakkaasta tarvittavat kentät, sekä kertoo tapahtumaa koskevat tiedot. Tällöin myyjä pystyy samalla kertaa tallentamaan asiakkaan ja tapahtuman järjestelmään, jolloin hänen ei tarvitse tehdä molempia erikseen.

2.6.3 Laiteriippumaton järjestelmä

Järjestelmän yksi tärkeimpiä asioita oli sen laite riippumattomuus. Yrityksen toiveesta järjestelmä toteutetaan responsiivisesti JQuery Mobilen Javascript ja Cascada Style Sheets (CSS) kirjastoja käyttäen. Tämän vuoksi myyjän ei tarvitse aina kantaa kannettavaa tietokonetta mukana tapaamisesta toiseen. Laiteriippumattomuuden vuoksi pelkkä puhelin riittää saamaan asiakkaasta tarvittavat tiedot ja myyntitapahtumatkin voi kirjata nopeasti järjestelmään.

Järjestelmään toteutettu ajo-ohjeiden saanti- ja navigointi-työkalu, sekä karttapalveluiden toiminta vaativat myös laiteriippumattomuutta. Käyttäjän tuli pystyä käyttämään näitä palveluita pääkonttorissa, sekä ollessaan matkalla konttorin ulkopuolella. Karttapalveluita ja ajo-ohjeiden saantia oli toivottu erityisesti myyjien ja tuotepäälliköiden osalta, joiden työ vaatii asiakaskäyntejä ympäri Eurooppaa.

3 Toteutuksessa käytetyt ohjelmointikielät

3.1 XML

Extensible Markup Language (XML)-kieli on kehitetty helpottamaan laajojen tietomassojen jäsentämistä ja sen perimmäinen tarkoitus on varastoida ja liikuttaa tietoa. ”XML tarjoaa tarkennetun pääsyn dokumentteihin, kun sitä käytetään ns. XML-haku menetelmänä palauttaen halutun dokumentin” (Lalmas & Mounia 2009, 1.). XML:n käyttäminen perustuu sovellusten väliseen kommunikoitiin, jota varten tarvitaan kuitenkin rajapinta. Yleisimmät rajapinta ratkaisut perustuvat WSDL + SOAP /AJAX-tekniikkaan, varsinkin web-järjestelmä integraatioiden kohdalla. WSDL tarjoaa verkkoon helposti kuvattavan näkymän rajapinnoista, joiden avulla XML hakuja voi suorittaa. Yleisimmät hakutekniikat ovat SOAP ja AJAX, jotka suorittavat toimintansa suoraan verkon yli. SOAP:n ja AJAX:n käyttö vaatii muita ohjelmointi kieliä, kuten C#, PHP, Java tai Javascript. Esimerkkinä XML-dokumentista on kuvassa 1 sähköpostin lähetys.

Toisena esimerkkinä voidaan pitää kirjaa, joka jaetaan otsikkoon, johdantoon ja useisiin kapaleisiin.

```
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Kuva 1. XML esimerkki. (w3 2013.)

3.2 SOAP

SOAP (Simple Object Access Protocols) on yksinkertainen protokolla, joka mahdollistaa tiedonsiirron ohjelmien välillä. SOAP lähettää ja vastaanottaa rakenteellista dataa rajapintojen välillä. Tiedonsiirtoformaattina SOAP käyttää XML:ää. ”Internetin yleistymisen johdosta, HTTP on tällä hetkellä yleisin tapa lähettää ja vastaanottaa tietoa SOAP viestejä” (Snell, Tidwell & Kulchen 2007, 34.).

Suurimmaksi osaksi SOAP:n käyttäminen perustuu Internetin yleistymisen myötä tilanteeseen, jossa halutaan kahden eri sovelluksen voivan kommunikoida keskenään verkon yli. Kommunikointiin tarvitaan tällöin rajapinta, jolloin SOAP tarjoaa mahdollisuuden tietojen siirtämiseen sovellusten välille.

3.3 WSDL

WSDL (Web Services Description Language) on XML-formaatti, joka antaa kuvauksen verkkopalveluista kokoelmana tietoliikenne viestinnän päätepisteitä. Päätepisteiden avulla voidaan tehdä palvelupyyntöjä toiseen palveluun. WSDL-palvelu tarjoaa dokumentoidun kuvauksen tietoa jakaville järjestelmille. Tämän takia ”WSDL-dokumentti on tärkeä osa jokaista rajapintaa, jota halutaan tarjota kuluttajille” (Evjen, Ferguson, Kay, Sharkey, Thangarathianm & Verner 2007, 633.), jotta kehittäjä pystyy näkemään millä pyynnöillä voi tehdä tarvittavia kutsuja toisesta järjestelmästä.

WSDL-dokumentilla kehittäjä pystyy helposti hakemaan toiminnot, joiden avulla saa halutun tiedon omaan järjestelmäänsä tai päinvastoin. Lemonsoft tarjosi tässä tapauksessa rajapinnan, joka mahdollistaa muiden järjestelmien liittämisen itseensä. Lemonsoftin WSDL-dokumentin pohjalta pystyttiin toteuttamaan kehitetty Web-CRM järjestelmä.

3.4 PHP

PHP (Hypertext preprocessor) on palvelinpuolen ohjelmointi kieli ja se on myös hyvä työkalu tehdessä dynaamisia ja vuorovaikuttaisia web-sivustoja. PHP on laajasti käytetty ohjelmointikieli. Se on ilmainen ja tehokas verrattuna suurimpaan kilpailijaansa. Suurimpana kilpailijana toimii Microsoft ASP, joka pohjautuu pääasiassa Microsoftin tuottamiin järjestelmiin ja tietokantoihin. Tästä syystä PHP onkin saanut suuren edun ja suosion kehittäjien keskuudessa. PHP:sta löytyy myös laaja valikoima ilmaisia ja sille ensisijaisesti suunnattuja lisäominaisuuksia, joita järjestelmien rakentamiseen tarvitaan. ”PHP on mahtava kieli, ja se tarjoaa kehittäjille mahdollisuuden rakentaa lähestulkoon mitä vain he ikinä voivat kehittääkään tietokannan ja HyperText Markup Language (HTML):n pariin” (Lengstorf 2010, 1.). Suosituimpana muotona PHP:ta käytetäänkin HTML:n sisään upotettuna koodina ja sitä käytetään yleisimmin keskustelemaan palvelimen tietokantojen kanssa, jonka avulla sivustoista tai järjestelmistä voidaan saada paljon vuorovaikutteisemmat. Kielenä PHP ei ole lainkaan Front-End, tyyppinen kuten CSS. Kuvassa 2 on yksinkertainen esimerkki tietokantahausta, jossa haetaan henkilöä nimellä HTML lomakekentän antaman arvon mukaan.

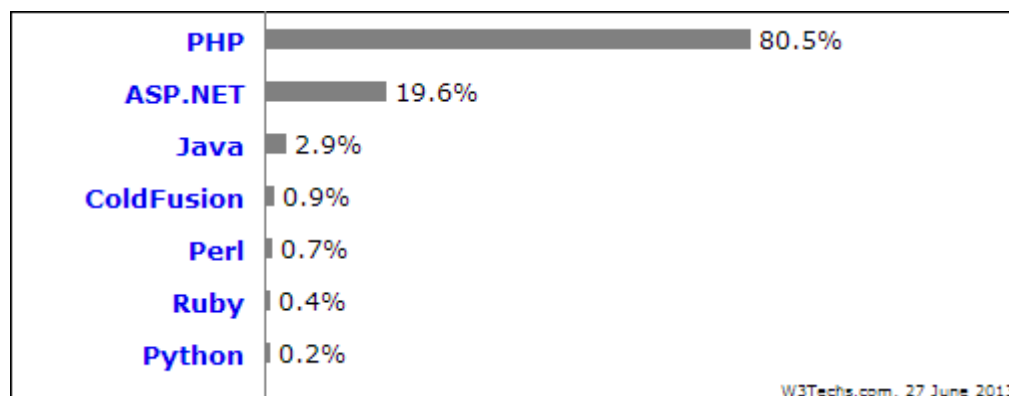
```
<?php
$name = $_POST['name'];
$con = mysqli_connect('localhost','admin','admin_pw');
if(!$con)
{
die('Could not connect ' . mysqli_error($con));
}
mysqli_select_db($con,"customers");
$sql="SELECT * FROM users WHERE user_name = '$name'";

$result = mysqli_query($con, $sql);

echo $result;
?>
```

Kuva 2. PHP-tietokantahaku.

PHP-kieli ei näy HTML-sivuston lähdekoodissa ulkopuoliselle käyttäjälle, joten tämäkin tuo jonkinasteista turvaa web-ympäristössä pyöriville järjestelmille. Kuvassa 3 näkyy selvästi kuinka suosittu PHP on palvelinpuolen käytössä, kuten tietokantojen kanssa.



Kuva 3. Ohjelmointikielten suosio. (w3 2013.)

3.5 JQuery mobile

Jquery Mobile on HTML 5-pohjainen Javascript- ja CSS-kirjasto. Kirjasto on laiteriippumaton, joka on ensisijaisesti kehitetty mobiililaitteille. Se toimii moitteettomasti ja on visuaalisesti näyttävä myös normaaleilla tietokonenäyttöillä. Se on kevyttä koodia, joka kirjoitetaan suoraan HTML koodiin. Kirjaston avulla voidaan toteuttaa lyhyessä ajassa näyttävän ulkoasun. Kirjasto on hyvin monipuolinen ja tarjoaa lähestulkoon rajattomat mahdollisuudet kehittäjille. Kevytyensä vuoksi se on hyvin tehokas ja toimii mallikkaasti vanhemmillakin puhelimilla.

Jquery Mobilella toteutetaan molempia sekä applikaatioita, että Web-sivustoja. JQuery Mobilella on hyvin helppo aloittaa ohjelmointi, riittää vain, että HTML sivun <head> </head>-tagien väliin kirjoitetaan polut CSS- ja Javascript-tiedostoihin. Tiedostot voi ladata paikallisesti itselleen JQuery Mobilen verkkosivuilta tai käyttää heidän ylläpitämiään kirjastoja www-osoitteen kautta. Osoitteen käyttöä suositellaan, koska niihin päivitetään kirjastojen uusimmat versiot aina automaattisesti tai pienet korjaukset, joita julkaisun jälkeen on ilmennyt (tosin kehittäjän pitää muuttaa hieman osoitetta uuden version ilmestyessä). Tällöin JQM-palvelun ylläpitäjän ei tarvitse aina ladata uusimpia kirjastoja omalle palvelimelleen vaan pieni osoitteen muutos riittää päivittämään sivuston kirjastot uusimpaan versioon.

JQM-kirjaston avulla aloittelevakin ohjelmoija pystyy luomaan näyttäviä sivustoja sen yksinkertaisuuden vuoksi. Kirjaston toiminnallisuuksien parhaimpia osia ovat sen kosketustoiminnot. "Kosketustoimintoja on erilaisia, ja ne esittävät pääroolia, kun oletetaan käyttäjän olevan vuorovaikutuksessa web-sovelluksen kanssa" (Bai 2011, 55.). Kosketustoimintoja ovat mm. "valitse" ja "pyyhkäise" toiminnot, jotka toimivat tietokoneella hiiren avulla. Kosketustoiminnoilla on monia variaatioita, esim. pitkään valitse-toiminnon pohjassa pitäminen. Pyyhkäisyt molempiin suuntiin tai napsauttamalla nappia aukeaisi uusi ikkuna tai alaspäin vetotaulukko.

Kokonaisuudessaan JQuery Mobile tarjoaa kattavan paketin erilaisia toimintoja kosketuksista näyttäviin efekteihin, jotka saadaan helposti kommunikoimaan toistensa kanssa. JQuery Mobilen ongelmaksi ei myöskään koidu muut ohjelmointikielet. JQM:n ollessa perus HTML-koodia, jonka sisään on vain upotettu tyyliviittaukset ja toiminnalliset funktiot, jää ainoaksi eroksi normaaliin HTML-sivustoon verrattuna valmiit CSS- ja Javascript-kirjastot. Kirjastojen avulla kehittäjä saa ulkoasun ja tarvittavat toiminnot halutulla tavalla käyttöön ja mobiililaitteille soveltuvaksi kokonaisuudeksi.

3.6 Javascript, HTML ja CSS

Javascript on ohjelmointikieli web-ympäristöön. Se on yksi yleisimmistä web-ympäristössä käytettävistä kielistä, joka ilmenee sillä, että sitä käytetään yli miljardilla eri web-sivustolla. Sen käyttö perustuu käyttäjäpuolen vuorovaikutteiseen kommunikointiin sivuston kanssa. Javascriptillä voidaan tuoda esimerkiksi sivuille erilaisia toimintoja, kuten vahvistuspyyntöjä, eteen pompahtavia ilmoituksia, ja palvelimen kanssa kommunikointia, tuoden vuorovaikutteisuuutta.

HTML (HyperText Markup Language) on pääkuvauskieli www-sivuille. Se kuvaa millä tavoin tieto, kuvat ja sisältö ilmenevät sivulla. HTML on dokumenttimainen kieli ja HTML-sivu alkaa ja päättyy `<html>` `</html>` tagien väliin, joiden sisään kirjoitetaan kaikki sivulla oleva sisältö. Suurimaksi osaksi HTML:ää käytetään vain luomaan sivustolle rungon. Visuaalisen puolen ja toiminnallisuuden hoitavat lähes poikkeuksetta muut kielet. Www- sivusto lukee HTML:ää ylhäältä alaspäin tuoden sisällön näkyville sen mukaan.

CSS (Cascada Style Sheets) on yksinkertainen koodi, joka mahdollistaa Web-dokumentin muotoilua monin eri tavoin. CSS:ää käytetään esimerkiksi antamaan Web-dokumentin muotoiluun, antamaan dokumentin taustalle ja teksteihin värejä, rivin vaihtoja, tai reunuksia määritetyille muodoille. Nykyään lähestulkoon jokainen Web-selain ja applikaatio tukevat CSS-koodia. Pää-tavoite CSS:n käytössä on asettaa sivupohjaan yhtäläinen teema määrittämällä CSS:llä halutunlaiset muotoilut, joiden avulla sivustoa on helpompi hallita ja muokata jälkepäin.

3.7 SQL

SQL (Structured Query Language) on joustava tietokanta-kyselykieli, jota voidaan käyttää erilaisiin tarkoituksiin. Se on kaikkein laajimmin käytetty kieli kommunikointiin relaatiotietokannoissa. SQL:n avulla pystytään relaatiotietokannoissa tekemään muutoksia, lisäyksiä, poistoja sekä tarkennettuja hakuja. Jokainen nykyaikainen relaatiotietokanta tukee SQL-

komentoja. Täytyy kuitenkin huomioida, ettei SQL ole toiminnollinen ohjelmointikieli, jolla voidaan aloittaa prosedureja tai kysyä käyttäjältä vastauksia. SQL-lausekkeella yksinkertaisesti vain kerrotaan, mitä halutaan, eikä kerrota järjestelmälle, kuinka toteuttaa sellaista, mitä kehittäjä itse haluaa proseduurin toteuttavan. Kuvassa 4 on esimerkki SQL-hausta, jossa haetaan EMPLOYEE taulusta yli 40 vuoden ikäisiä henkilöitä joiden palkka on yli 60 000 e.

```
SELECT * FROM EMPLOYEE WHERE Age > 40 OR Salary > 60000 ;
```

Kuva 4. SQL haku. (Taylor 2010, 23-25.)

4 Järjestelmäintegraatio

4.1 Tutkimuksen ERP-järjestelmän tarjoama rajapinta

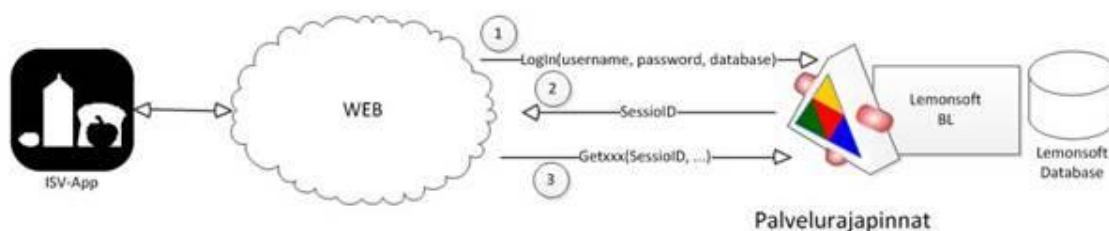
Lemonsoft eli tutkimuksen ERP-järjestelmä tarjoaa mahdollisuuden yrityksille kehittää omia sovelluksia hyödyntäen hyväksi samaa dataa, minkä ohjelmistokokonaisuus pitää sisällään. Lemonsoft tarjoaa myös valmiin kokonaisuuden metodeja, joiden avulla tiedonsiirto rajapinnan avulla onnistuu. Metodeista saa tarkemmat kuvaukset ja tietotyypit rajapinnan asentamisen jälkeen WSDL-sivulta, josta tullaan kertomaan myöhemmin tutkielmassa. Kuvassa 5 on metodi, joka palauttaa asiakasnumerolla kyseenomaisen asiakkaan tiedot.

```
''' <summary>
''' Palauttaa halutun asiakkaan
''' </summary>
''' <param name="iCustomer">Asiakasnumero</param>
''' <returns></returns>
''' <remarks></remarks>
<OperationContract>> _
Function GetCustomer(ByVal iCustomer As Integer) As Proxy.Customer
```

Kuva 5. Asiakashaun metodi. (Lemonsoft 2013.)

Tutkimuksen rajapintana käytetään Palvelurajapintaa (Web Services), koska kehitettävän järjestelmän tulee olla vuorovaikutteinen. Lemonsoft tarjoaisi myös mahdollisuuden siirtää dataa Comma-SeparatedValues (CSV) tai XML formaateissa, mutta nämä formaatit eivät mahdollista reaaliaikaista vuorovaikutteisuutta käyttäjän ja tiedonsiirtämisen välillä. Rajapinta toteutetaan käyttäjälähtöisellä tavalla, koska järjestelmää ei ole tarkoitettu ulkopuolisten henkilöiden käytettäväksi. Tämä tarkoittaa, että käyttäjän täytyy kirjautua ensin taustajärjestelmään, jotta saadaan varmuus henkilön oikeuksista käyttää järjestelmää. Tämän jälkeen

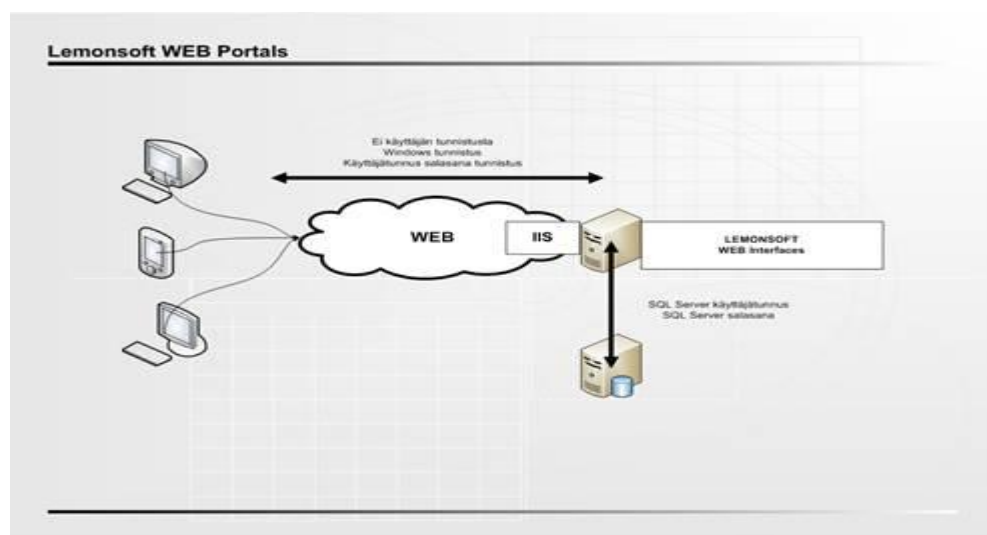
sovellus vasta luo rajapinnan Lemonsoftiin. Kuvassa 6 on esimerkki, siitä miten palvelurajapinta luodaan.



Kuva 6. Palvelurajapinnan luominen. (Lemonssoft 2013.)

4.2 Kehitettävän järjestelmän ja nykyisen ERP:n välinen tietoliikenne

Käyttäjän kirjautuessa järjestelmään avautuu palvelurajapinta. Välittömästi rajapinnan auetessa on käyttäjällä mahdollisuus tietojen hakemiseen ja tallentamiseen. Tietoa asiakkaista, toimittajista, tuotteista, sekä myynti ja asiakastapahtumista liikkuu järjestelmästä toiseen. Järjestelmien tiedonsiirto on toteutettu PHP+SOAP-yhdistelmällä. Yhdistelmä tarjoaa mahdollisuuden vuorovaikutteiseen tiedonsiirtoon käyttäjän ja palvelimen tietokannan välillä. Kuvassa 7 on esimerkki, millä tavoin tieto liikkuu järjestelmästä toiseen.

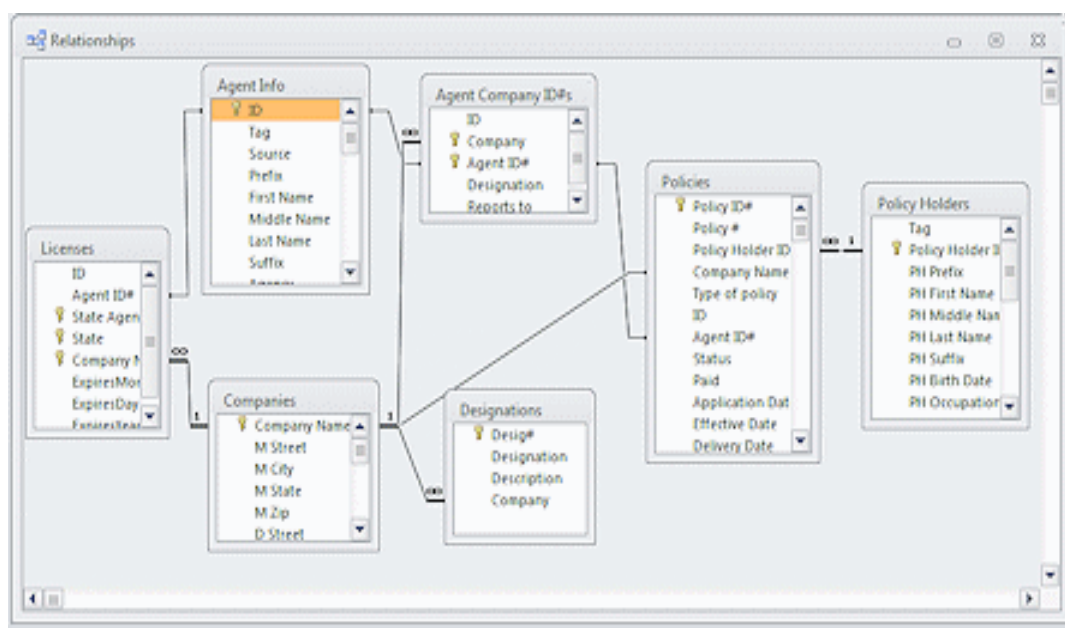


Kuva 7. Web infrakstuuri. (Lemonssoft 2013.)

Kehitettävässä järjestelmässä liikkuva data pohjautuu kokonaan Lemonsoftin tietokantaan. Tietokantaan on tallennettu jo monien vuosien ajan dataa kaikesta mahdollisesta. Tallennettu data on esimerkiksi asiakkaiden ostohistoriaa, laskutusta, myyjien myyntilukuja, ja jopa va-

rastojen hallintaan vaadittavaa kirjanpitoa. Tietokantaan vuosien mittaan tallennettu data on tehnyt siitä jo massiivisen kokoisen. Lemonsoft on kehittänyt tämän takia tietokantaan erinäköisiä taulujen välisiä linkityksiä. Lemonsoftin liiketoimintalogiikka Microsoftin tietokannassa ei mahdollista yhden asiakkaan tietojen tallentamista vain yhteen tauluun, vaan se vaatii monia tauluja yksittäiselle asiakkaalle, kuten ostohistoria, perustiedot ja huoltohistoria. Nämäkin taulut vielä linkittyvät pienemmiksi osiksi alitauluihin. Tietokannassa data-linkitykset perustuvat lähes poikkeuksetta yksilöllisiin: tuote, asiakas tai toimittajanumeroihin. Numerot helpottavat huomattavasti tiedon hakemista.

Tietokannan ison koon takia, on järjestelmien välillä liikkuva data pakotettu toteuttamaan rajapintojen avulla. Lemonsoftin kehittämät rajapinta-metodit mahdollistavat taulujen väliset linkitykset automaattisesti yhdellä ohjelmointikäskyllä. Linkitykset toimivat samalla tavalla kuin Lemonsoft järjestelmässäänkin. Ilman Lemonsoftin kehittämiä rajapinta metodeja, olisivat tallentamiseen vaadittavat ohjelmointirivit tolkkuttoman pitkiä ja vaatisivat monien tuntien ylimääräistä tutkimista tietokannassa oleviin linkityksiin. Kuvassa 8 on esimerkki taulujen välisistä linkityksistä. Kuvassa olevat avaimet pitävät sisällään yksilöllisiä numeroita, jotka mahdollistavat taulujen väliset linkitykset.



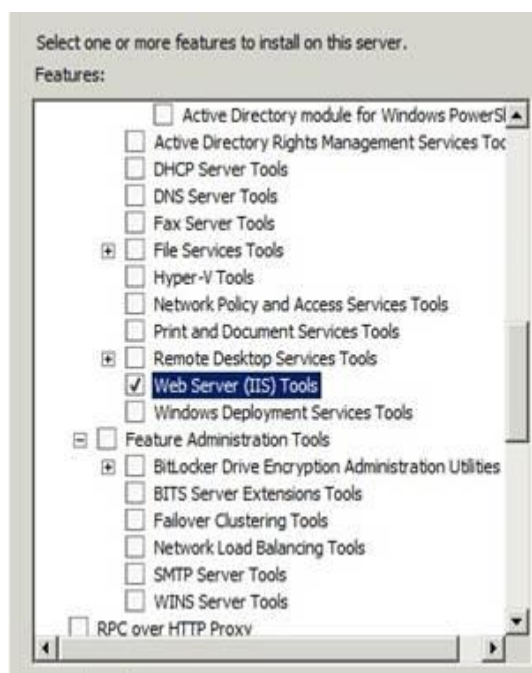
Kuva 8. Taulujen linkitykset. (Smith 2012.)

4.3 Rajapinnan käyttöönottoaminen

Tässä tutkimuksessa rajapinnan käyttöönottoaminen Lemonsoft toiminnanohjausjärjestelmään on hyvin vaivatonta. Rajapintaa ei tarvinnut rakentaa itse, vaan Lemonsoftilla on olemassa

oma asennuspaketti sitä varten. Asennuspaketti mahdollistaa järjestelmän rajapintatoimintojen käyttämisen. Asentaessa pakettia tulee ottaa huomioon, että asennuspaketti vaatii palvelun tarjoajaksi Microsoftin kehittämän Web-palvelinohjelmiston Internet Information Services (IIS). IIS on toiseksi suosituin Web-palvelinohjelmisto 17 %:n markkinaosuudella heti Apachen 53 %:n jälkeen. Netcraft toteutti Web-palvelinohjelmisto tutkimuksen kesäkuussa 2013. IIS tunnetaan etenkin yritysmaailmaan pohjautuvana ohjelmistona.

Palvelinkonetta asentaessa tulee huomioida IIS:n tarpeellisuuden lisäksi se, että Microsoft framework 4.0 ja Lemonsoftin taustajärjestelmä ovat asennettuina. Ilman toimivaa taustajärjestelmää ja frameworkia, IIS yksinään ei riitä siihen, että rajapintapaketti toimisi oikein. Kun tarvittava framework ja taustajärjestelmä on asennettu, aloitetaan IIS asentaminen. On huomioitava, ettei IIS ole oletuksena asennettu Windows käyttöjärjestelmään, vaan sitä varten täytyy määrittää ja asentaa erinäisiä osia Windowsissa. IIS:n asentaminen lähtee liikkeelle Windowsin Lisää Poista Windowsin osia kohdasta. Tämä löytyy Windowsin ohjauspaneelistä. Aloittaessa asennusta tulee ensiksi valita Web Server (IIS) Tools-valikosta kuvassa 9 näkyvä.



Kuva 9. Web Server (IIS) Tools.

Asentaminen jatkuu samasta paikasta, josta valitaan Windows Process Activation Service. Windows Process Activation Servicen asentamisen jälkeen se tulee käynnistää Windowsin Palvelut osiosta kuvan 10 osoittamalla tavalla. Palvelut osio löytyy Ohjauspaneeli -> Hallintatyökalut -> Palvelut polun alta.



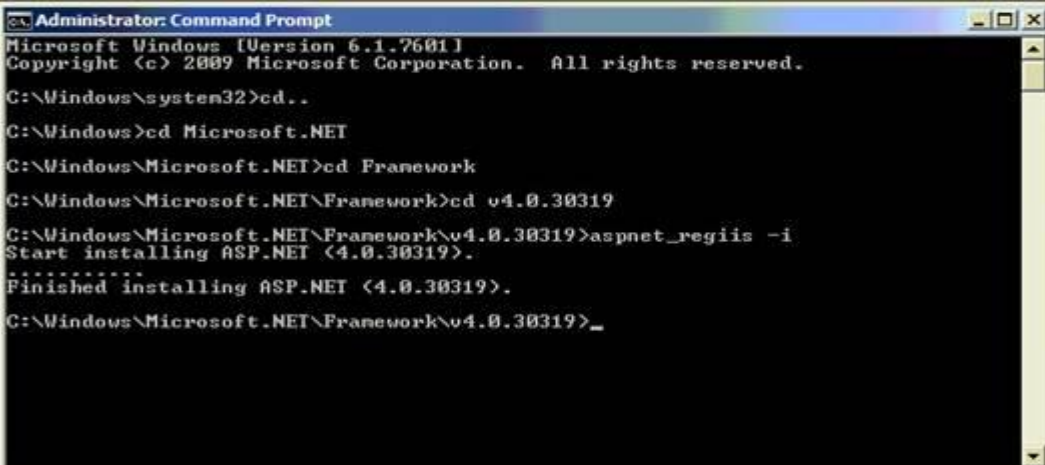
Kuva 10. Windows Process Activation Service.

Käynnistämisen jälkeen lisätään kaikki Windowsin ehdottamat ominaisuudet ja palataan takaisin Lisää Poista Windowsin osia kohtaan. Osioista tulee valita Web Services -> Application Development ja sieltä ASP.NET. ASP.NET asennuksen yhteydessä valitaan ja asennetaan taas kuvan 11 osoittamalla tavalla kaikki Windowsin ehdottamat ominaisuudet.



Kuva 11. Lisäominaisuuksien lisääminen.

Asennusten jälkeen avataan komentorivi joko kirjoittamalla run kohtaan cmd.exe tai valitaan Kaikki ohjelmat -> Apuohjelmat -> Komentorivi. Jotta ASP.NET toimisi kunnolla, tulee asentaa kuvan 12 esittämällä tavalla seuraavat kaksi pakettia.



```

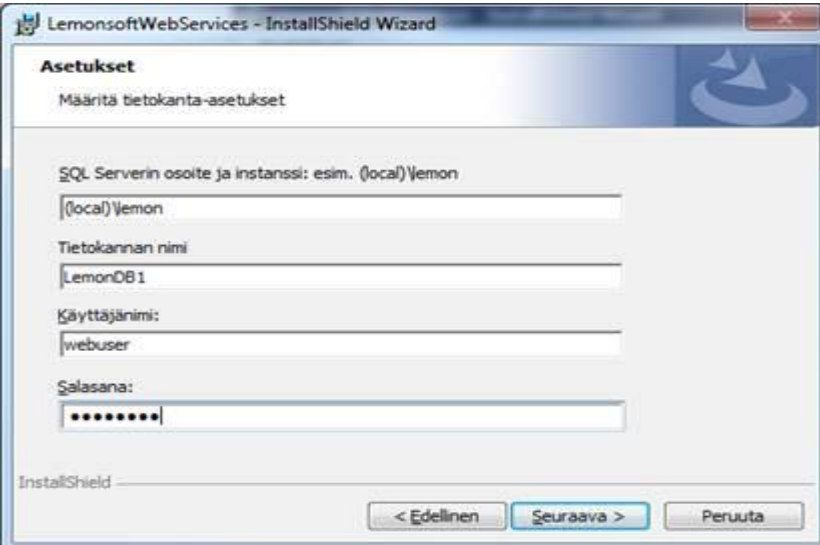
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd..
C:\Windows>cd Microsoft.NET
C:\Windows\Microsoft.NET>cd Framework
C:\Windows\Microsoft.NET\Framework>cd v4.0.30319
C:\Windows\Microsoft.NET\Framework\v4.0.30319>aspnet_regiis -i
Start installing ASP.NET <4.0.30319>.
Finished installing ASP.NET <4.0.30319>.
C:\Windows\Microsoft.NET\Framework\v4.0.30319>_

```

Kuva 12. ASP.NET pakettien asennus (Lemonsoft 2013).

Pakettien asentamisen jälkeen voidaan aloittaa Lemonsoft:n rajapintapaketin asentaminen. Ensiksi tulee määrittää tietokanta, jonka kanssa rajapinta lähettää ja vastaanottaa dataa. Kuvassa 13 on esimerkki, millä tavoin kentät tulee täyttää omien määritysten mukaisesti. Asennukseen tarvitaan Lemonsoftin Admin tunnukset tai vastaavat oikeudet omaavat tunnukset. Admin-tunnukset tarkoittavat sovelluksen tai järjestelmän pääkäyttäjän tunnuksia. Näillä tunnuksilla pääkäyttäjä pystyy hallinnoimaan järjestelmää ja hänellä on rajattomat oikeudet järjestelmään. Pääkäyttäjä voi esimerkiksi luoda uusia käyttäjiä, muokata muiden käyttäjien oikeuksia nähdä tietoja, antaa mahdollisuuden muokata tai tallentaa dataa järjestelmään. Admin tunnukset ovat pakolliset asennusvaiheessa. Tämä mahdollistaa myöhemmässä vaiheessa oikeudet hakea, tallentaa ja muuttaa tietokannassa olevaa dataa.

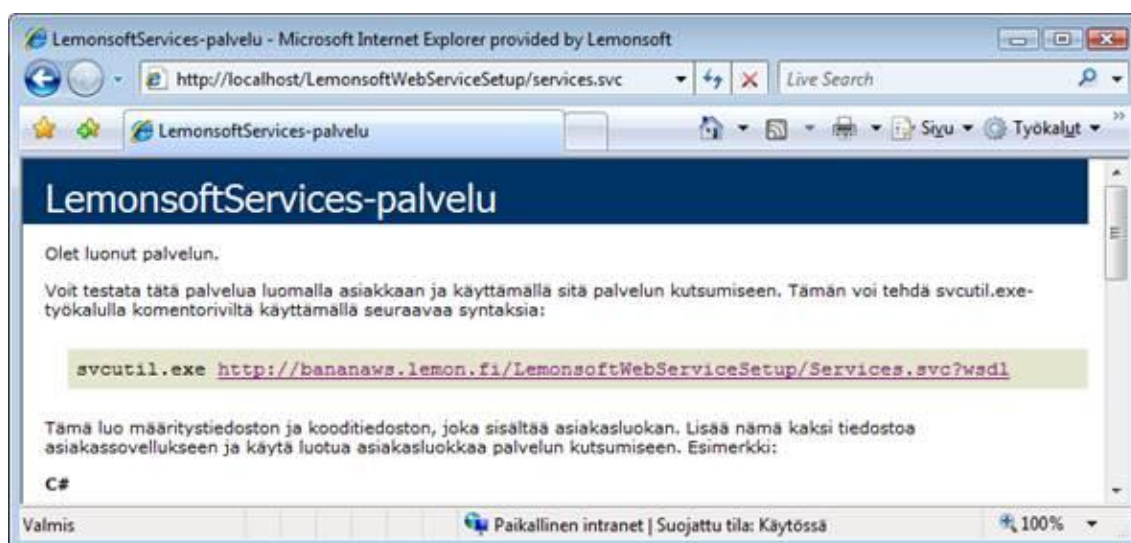


Kuva 13. Lemonsoft tietokannan määrittäminen. (Lemonsoft 2013.)

Seuraavassa vaiheessa määritetään nimi kansiolle, joka tulee IIS kansion alle. Tämä nimi vaikuttaa myös siihen, mikä tulee osaksi lopullista rajanpinnan esittämispolkua verkossa olevaan WSDL tiedostoon. Asennus on tämän jälkeen pääosin eteenpäin menemistä. Lopulta asennusta ja toimivuutta voi testata osoitteessa:

<http://localhost/LemonsoftWebServiceSetup/services.svc> (osoite voi olla eri riippuen määrittelyistä kansion ja tietokannan osoitteiden suhteen aikaisemmissa vaiheissa)

Onnistuneen määrittämisen ja asentamisen jälkeen lopputuloksen tulisi näyttää tältä (Kuva 14).



Kuva 14. LemonsoftServices-palvelu. (Lemonsoft 2013.)

Tarkempaa ohjeistusta Lemonsoftin rajapintapaketin asentukseen löytyy osoitteesta: <http://info.lemonsoft.eu/LemonNethelp/#!Documents/palvelurajapinnatweb.htm>

4.4 Rajapinnan käyttäminen

Onnistuneen rajapinnan asentamisen jälkeen Lemonsoftin toteuttamat rajapintatoiminnot toimivat ja mahdollistavat yhteyden kahden eri sovelluksen välille. Pelkästään jo asiakkuudenhallintaan on olemassa kolmesta erilaista toimintoa eli metodia. Kehitettävässä järjestelmässä suurimmassa käytössä ovat asiakkaiden lisääminen, asiakastietojen hakeminen ja niiden päivittäminen. Kuvassa 15 on esimerkki, miten PHP ja SOAP yhdistelmällä haetaan ennalta määritetyllä asiakasnumerolla sen tietoja Lemonsoftin tietokannasta.

```

// Muodostetaan tietoluokka
$asiakashaku = new stdClass();
$asiakashaku->iCustomer = 2;

// Haetaan asiakasyrityksen tiedot, jos mahdollista
try
{
    $soap = new soapClient( constWSDL, array( "trace" => 1 ) );           // Keskusteluyhteys
    $asiakas = $soap->GetCustomer( $asiakashaku );                       // Haetaan asiakastiedot
} catch (SoapFault $fault) { echo "Tietojenhakuvirhe: KOODI: ".$fault->faultcode."

// Otetaan oliosta nimi muuttujaan
$nimi = htmlentities($asiakas->GetCustomerResult->Customer_name1, ENT_COMPAT, "UTF-8");

```

Kuva 15. Rajapinnan käyttäminen. (Lemonsoft 2013.)

5 Tietoturva

Tietoturva tarkoittaa digitaalisen datan suojaamista tietomurroilta. Tietoturva onkin tekniikan kehittyessä noussut todella suureen rooliin nyky-yhteiskunnassa. Siitä kertovat monet yritykset, jotka ovat siirtyneet viimeistään viimeisen kymmenen vuoden aikana hiljalleen pöydällä notkuvista paperikasoista pilvipalveluiden tarjoamiin mahdollisuuksiin mukaan lukien yrityksen omat palvelinkoneet. Omilla palvelinkoneilla yritykset pystyvät ylläpitämään omia tietokantojaan, joiden hallitsemiseen ja käyttämiseen on olemassa jo monia eri ohjelmistovaihtoehtoja. Laskujen maksaminen on siirtynyt konttoriin menemisen sijasta verkkoon E-laskujen muodossa. Yritysten ja heidän henkilöstönsä osalta yritystietojen digitalisointi ja nopea saatavuus ovat jo välttämättömyyksiä, jos tahdotaan menestyä. Tämä helpottaa työntekijöiden välistä kommunikointia ja nostattaa työn tehoa huomattavasti. Tällöin säästytään turhilta paperien etsimisiltä, postituksilta ja niiden tuomista viiveistä. Suurimassa osassa uuden tekniikan tuomista eduista löytyy kuitenkin haittapuolensa, kuten tästäkin. Nimittäin yritystietojen digitalisointi helpottaa myös tietomurtojen tekoa huomattavasti. Jos paperiaikana haluttiin tehdä tietomurtoja, jouduttiin kirjaimellisesti murtautumaan yrityksen toimitiloihin. Nykyään on mahdollista saada jopa isojenkin yritysten tietoja haltuun kotisohvalta käsin hakeroitumalla heidän järjestelmäänsä heikkojen tietoturvaratkaisuiden takia.

Oikeiden tietoturvaratkaisuiden löytämiseksi piti tässä tutkimuksessa myös perehtyä ja miettiä, mikä olisi turvallisin tapa suojata järjestelmään pääsy ulkopuolisilta. Järjestelmän ollessa käytössä ympärivuorokauden, täytyi siihen rakentaa tunnistautuminen ja suojata se hakuko-neilta. Ilman tunnistautumista voisi kuka tahansa, joka löytäisi kyseisen sivuston vaikka Googlen avulla, käyttää järjestelmää. Tunnistautumisen avulla varmistutaan henkilön oikeuksista päästä tietoihin käsiksi. Tunnistautuminen tapahtuu järjestelmän kirjautumissivulla. Jos hen-

kilö ei ole aktiivinen käyttäjä eli kirjautuneena järjestelmään, hän ei voi päästä käsiksi muihin sivuihin, vaan järjestelmä avaa tällöin herjaussivun, joka ohjaa kirjautumisivulle. Hakukonesuojaus estää sivuston näkymisen esim. Googlen haetuissa listauksissa.

Kirjautumisvaiheessa tulee ottaa huomioon, että siinä tehdään suora kysely tietokantaan. Tämä tarkoittaa sitä, että järjestelmä on haavoittuvainen SQL-kyselyiden eli SQL-injektiohyökkäysten osalta. SQL-injektio voi tapahtua, kun käyttäjä antaa SQL-koodia Websivuston kirjautuminen tai salasana -kenttään ja toteuttaa siten kyselyn suoraan tietokantaan. Näiden hyökkäysten estämiseen oikea tapa on käyttää parametrisia kyselyitä. Tämä tarkoittaa sitä, että SQL-koodi tullaan toteuttamaan ohjelmallisesti lisäämällä saatuihin merkistöihin parametriset arvot. Tämä antaa palvelimelle mahdollisuuden luoda toteutus suunnitelmakyselyn, joka estää SQL Injektion toteuttamisen (Swan 2010). Kuvassa 16 on esimerkki, kuinka PHP koodi tulisi kirjoittaa, jotta SQL injektio hyökkäykset saataisiin torjuttua.

```
<form method="post" action="injection.php" enctype="multipart/form-data" >
    Username:<input type="text" name="Username" id="Username"/></br>
    Password:<input type="text" name="Password" id="Password"/></br>
    <input type="submit" name="submit" value="Submit" />
</form>
<?php
$params = array($_POST['Username'], $_POST['Password']);

$server = "MyServer\sqlexpress";
$options = array("Database"=>"ExampleDB", "UID"=>"MyUID", "PWD"=>"MyPWD");
$conn = sqlsrv_connect($server, $options);
$sql = "SELECT * FROM UserTbl WHERE Username = ? and Password = ?";
$stmt = sqlsrv_query($conn, $sql, $params);
if(sqlsrv_has_rows($stmt))
{
    echo "Welcome.";
}
else
{
    echo "Invalid password.";
}
?>
```

Kuva 16. Tietoturva kirjautumisessa. (Swan 2010.)

Tutkimuksessa käytettyä palvelinkonetta määrittäessä tuli huomioida, että Lemonsoft pyöri yrityksen palomuurin takana, jonne palvelinkonetta ei voinut laittaa turvallisuus syistä. Tähän piti tehdä ratkaisu, joka mahdollisti palvelinkoneen pääsyn sisäverkkoon Lemonsoft-palvelimelle, kyselyiden tekemistä ja vastaanottamista varten niin, ettei ulkopuolelta pääse käsiksi sisäverkkoon. Tätä varten otettiin käyttöön DMZ-kehäverkko, joka on fyysinen tai loo-

ginen aliverkko, joka voi sisältää yrityksen ulkopuolisia palveluita, kuten verkkosivut. Tässä tapauksessa järjestelmä nostettiin DMZ-kytkimen avulla ulkoverkkoon ja kytkimestä on yhteys suoraan sisäverkkoon, joka mahdollistaa tiedon siirtämisen ja hakemisen sisäverkosta Lemonsoftin palvelimelta, mutta estää ulkopuolisilta pääsyn suoraan sisäverkkoon.

6 Johtopäätökset ja Loppusanat

6.1 Johtopäätökset

Tutkimus lähti liikkeelle yrityksen johdon pyynnöstä. Johto oli jo pitkään etsinyt erilaisia ratkaisuja, joilla tyydytettäisiin myyjien tarpeet. Suurimpana toiveena oli saada asiakastietoja kentälle nopeasti laitteesta riippumatta. Myös myyjien heikko raportointi hallintoon oli ollut ongelma yrityksessä. Välillä ei tiedetty, missä kukin myyjä menee ja mitä hän on kauppamassa ja kenelle.

Uuden tulevan järjestelmän tulisi pyöriä web-ympäristössä ja sen täytyisi olla liitoksissa tällä hetkellä pääkonttorissa olevaan ERP-järjestelmään. Kehitettävän järjestelmän ja ERP:n välillä olevan liitoksen päätoiminen tarkoitus on antaa myyjille mahdollisuus hakea, päivittää ja lisätä asiakastietoja Lemonsoftista. Reaaliaikaisten tuotetietojen ja varastosaldojen saaminen kentälle helpottaisi huomattavasti myyjien työskentelyä.

Järjestelmän muut ominaisuudet, kuten karttapalvelut ja asiakastapahtumat kehitetään lisäominaisuuksina hyödyntäen ERP-järjestelmän tietokantaa, mutta ei rajapintoja.

Pienen tutkimuksen jälkeen järjestelmän palvelinpuolen tiedonsiirtoihin päädyttiin käyttämään PHP-ohjelmointikieltä. Rajapinta Lemonsoftiin oli toteutettu PHP+SOAP-yhdistelmällä, joten oli luonnollista toteuttaa myös suorat haut tietokannasta sen avulla.

Ulkoasun kohdalla tuli alussa vastaan pieniä ongelmia. Molemmat, sekä myyjät, että johto, toivoivat webpohjaista järjestelmää, sen tarkemmin tietämättä, millä tavoin tämä toteutettaisiin. Pitkään muutaman erilaisen ulkoasuteeman toteuttamisen jälkeen löytyi se oikea toteutustapa, johon molemmat osapuolet ihastuivat. Ulkoasun toteutukseen valittiin JQuery Mobile (Javascript ja CSS) kirjastot HTML 5-koodille. JQuery Mobile on myös laiteriippumaton, joten se täytti täten kaikki järjestelmälle asetetut pohjakriteerit.

Kunnollisten vaatimusmäärittelyiden jälkeen päästiin itse ohjelmointiin käsiksi. Ohjelmointi vaihe kesti noin 4-5 kuukautta ja eteni lähes ongelmitta järjestelmän tuotantoon nostoon saakka. Kokonaisuudessaan projektiin kului aikaa noin seitsemän kuukautta, tämä sisälsi aiheeseen perehtymisen, muutamien ulkoasujen toteutuksen, vaatimusmäärittelyiden teon ja

itse ohjelmoinnin ja palvelinpuolen määritykset. Tutkimuksen toteuttaja toimi lähes täysipäiväisesti projektin parissa koko toteutuksen ajan.

Toteutettua Web-CRM järjestelmää tullaan varmasti jatkossa kehittämään eteenpäin siitä mi- hin tutkimuksessa jäätiin. Suurimpana syynä on yrityksen toiminnanohjausjärjestelmä, joka ei ole vielä kukaan täysin valmis paketti, vaikka se tuli yrityksen käyttöön jo vuonna 2006. Sitä ke- hittävät vieläkin monet ihmiset entistä paremmaksi, luotettavammaksi ja nopeammaksi. Aina Lemonsoftin kehittyessä, tulee muiden siihen liitoksissa olevien järjestelmien pysyä mukana. Kehitettyyn järjestelmään on jo pienen tuotannossa olon jälkeen tullut muutamia pyyntöjä, joilla sitä voitaisiin viedä eteenpäin.

6.2 Loppusanat

Projektina kokonaisen järjestelmän kehittäminen oli hyvin mielenkiintoinen ja antoisa. Näin ison projektin toteuttaminen oli myös hyvin työlästä. Tutkimusta tehdessä kohtasin monia ongelmia, joihin en ollut koskaan aikaisemmin törmännyt koulussa tai työpaikoissa. Osassa ongelmista en edes tiennyt oliko nämä mahdollisia toteuttaa vai ei, koska yrityksen sisältä ei löytynyt yhtään ohjelmointitaitoista ihmistä, joka olisi voinut auttaa ongelmien kanssa. Osaan ongelmista jouduin parhaillaan etsimään tietoa Internetistä useita päiviä.

Tutkimuksen aikana opin hyvin ymmärtämään isommankin organisaation IT-Infrastruktuuria. Si- säistin myös, kuinka tärkeätä on lopulta, että kaikki yritysten sisällä olevat järjestelmät ovat jotenkin yhteydessä toisiinsa. Tässä projektissa tuli kiinnittää huomiota todella moniin eri seikkoihin. Ollessani ainoa henkilö koko projektissa koko sen prosessin ajan jäi vastuulleni huolehtia tietoturvasta, ottaa selvää kuinka rajapinnat toimivat ja kuinka saada ylipäättänsä rajapinta välittämään ja vastaanottamaan tietoa yrityksen toiminnanohjausjärjestelmästä. Minun tuli huolehtia lisäksi sovelluksen toiminnoista, SQL-näkymien toteuttamisesta, ulkoasus- ta ja verkkoon nostamisesta, sekä sen ohessa http-serverien valinnasta, vertailla ja testilla erilaisia vaihtoehtoja ja valita näistä paras.

Kaikista mutkista ja ongelmista huolimatta olen todella tyytyväinen lopputulokseen. Kehittä- mäni järjestelmä on tällä hetkellä käytössä yrityksen jokapäiväisessä liiketoiminnassa monien eri henkilöiden osalta. Opin todella paljon asioita järjestelmien kehittämisestä ja ylipäättänsä sovelluskehityksestä. Opin määrittämään palvelinkoneen ja tekemän siitä virtuaalipalvelimen, jotta sitä olisi helpompi hallinnoida. Opin ymmärtämään rajapintoja, miten ne toimivat ja miten niitä voidaan lukea WSDL-verkkosivulta. Ohjelmoinnin yhteydessä ohjelmointitaitokin kehittyi huomattavasti, etenkin JQueryn ja Javascriptin osalta. Kaikkein tärkeimpänä oppi-

mastani pidän sitä, että nyt osaan ja hallitsen sovelluskehityksen jokaisen vaiheen alusta loppuun ja nyt tiedän, miten koko kokonaisuus toimii ja mitä se vaatii toimiakseen.

Lähteet

Kirjalliset Lähteet

Bai, G. 2011. JQuery Mobile First Look. Packt Publishing.

Bonnet, M. & MacIntosh, R. 2007. International Perspectives on Validity in Action Research. Emerald Group Publishing Ltd.

Davidson, J. & Di Gregorio, S. 2008. Qualitative Research Design for Software Users. McGraw-Hill Education.

Evjen, B., Sharkey, K., Thangarathinam, T., Kay, M., Vernet, A. & Ferguson, S. 2007. Professional XML. Wiley.

Freebody, P. 2006. Qualitative Research in Education. SAGE Publications. London, GBR.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2009. Tutki ja kirjoita. 15. uud. p. Tammi. Helsinki.

Lengstorf, J. 2010. PHP for Absolute beginners. Springer.

Lalmas, M. 2009. XML Retrieval. Morgan & Claypool Publishers.

Snell, J., Tidwell, D. & Kulchenko, P. 2007. Programming Web Services with SOAP. O'reilly Media Inc.

Taylor, A.G. 2010. SQL For Dummies. Wiley.

Sähköiset Lähteet

w3. 2013. PHP 5 Tutorial. Viitattu 26.6.2013.

<http://www.w3schools.com/php>

PHP. 2013. Viitattu 26.6.2013.

<http://www.php.net>

w3. 2013. Extensible Markup Language (XML). Viitattu 28.6.2013.

<http://www.w3.org/xml>

Helander, T. 1999. XML:n käyttäminen sovellusten välisessä tiedonvaihdossa. Viitattu 29.6.2013.

https://www.tml.tkk.fi/Studies/Tik-110.300/1998/Essays/xml_2.html

w3. 2013. SOAP Introduction. Viitattu 1.7.2013.

http://www.w3schools.com/webservices/ws_soap_intro.asp

w3. 2001. Web Services Description Language (WSDL) 1.1. Viitattu 3.7.2013.

<http://www.w3.org/TR/wsdl>

Jquery Mobile. 2013. Viitattu 29.6.2013.

<http://jquerymobile.com/>

w3. 2013. Javascript Introduction. Viitattu 29.6.2013.

http://www.w3schools.com/js/js_intro.asp

w3. 2013. Cascading Style Sheets. Viitattu 4.7.2013.

<http://www.w3.org/Style/CSS/>

w3. 2013. Introduction to SQL. Viitattu 4.7.2013.

http://www.w3schools.com/sql/sql_intro.asp

w3. 2013. Usage of server-side programming languages for websites. Viitattu 4.7.2013.

http://w3techs.com/technologies/overview/programming_language/all

Lemonsoft. 2013. Viitattu 6.7.2013.

<http://lemonsoft.fi/etusivu>

Lemonsoft. 2013. Palvelurajapinnat (Web Services). Viitattu 3.8.2013.

<http://info.lemonsoft.eu/LemonNethelp/#!Documents/palvelurajapinnatweb.htm>

Smith, J. 2013. Jan's Computer Basics: Applications: Database. Viitattu 4.8.2013.

<http://www.jegsworks.com/lessons/lesson1-2/lesson2-4database.htm>

Dieta. 2013. Yritysesittely. Viitattu 12.8.2013.

<https://www.dieta.fi/fi/hankemyynti/Yritysesittely>

Lemonsoft. 2013. Palvelurajapintojen käyttöönotto. Viitattu 14.8.2013.

<http://info.lemonsoft.eu/LemonNethelp/#!Documents/palvelurajapintojenk.htm>

Netcraft. 2013. Web Server Survey. Viitattu 14.8.2013.

<http://news.netcraft.com/archives/2013/06/06/june-2013-web-server-survey-3.html>

Lemonsoft. 2013. IIS ja Lemonsoft WebService asennus. Viitattu 14.8.2013.

<http://info.lemonsoft.eu/LemonNethelp/#!Documents/iisjalemonsoftwebser.htm>

Saaranen-Kauppinen, A. & Puusniekka, A. 2006. KvaliMOTV - Menetelmäopetuksen tietovaranto. Tampere. Viitattu 30.9.2013.

http://www.fsd.uta.fi/menetelmaopetus/kvali/L3_3_1.html

Swan, B. 2010. What's the Right Way to Prevent SQL Injection in PHP Scripts.

Viitattu 1.10.2013.

http://blogs.msdn.com/b/brian_swan/archive/2010/03/04/what_2700_s-the-right-way-to-avoid-sql-injection-in-php-scripts_3f00_.aspx

Saukkonen, P. 2013. Tutkimusongelma ja tutkimuskysymys. Viitattu 14.11.2013.

<http://www.mv.helsinki.fi/home/psaukkon/tutkielma/Tutkimusongelma.html>

Dominic, J. R. 2010. How to: The Comma Separated Value (CSV) File Format create or parse data in this popular pseudo-standard format. Viitattu 15.12.2013

<http://creativyst.com/Doc/Articles/CSV/CSV01.htm>

Microsoft. 2013. Introduction to IIS. Viitattu 15.12.2013

<http://www.iis.net/learn/get-started/introduction-to-iis>

Kuvat

Kuva 1. XML esimerkki.	16
Kuva 2. PHP tietokantahaku.	17
Kuva 3. Ohjelmointikielten suosio.....	18
Kuva 4. SQL haku.	20
Kuva 5. Asiakashaun metodi.	20
Kuva 6. Palvelurajapinnan luominen.....	21
Kuva 7. Web infrakstuuri.	21
Kuva 8. Taulujen linkitykset.	22
Kuva 9. Web Server (IIS) Tools.	23
Kuva 10. Windows Process Activation Service.	24
Kuva 11. Lisäominaisuuksien lisääminen.	24
Kuva 12. ASP.NET pakettien asennus.	25
Kuva 13. Lemonsoft tietokannan määrittäminen.	25
Kuva 14. LemonsoftServices-palvelu.	26
Kuva 15. Rajapinnan käyttäminen.....	27
Kuva 16. Tietoturva kirjautumisessa.	28