

Petri Leinonen

MYNNIN JOHTAMISEN INTERNETPALVELU

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Tradenomikoulutus
Tietojenkäsittelyn koulutusohjelma
Syksy 2007



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittely
Tekijä(t) Petri Leinonen	
Työn nimi Myynnin johtamisen internetpalvelu	
Vaihtoehtoiset ammattiopinnot Ohjelmointi	Ohjaaja(t) Veli-Pekka Piirainen Toimeksiantaja Arffman Consulting Oy (Kimmo Nikkanen)
Aika Syksy 2007	Sivumäärä ja liitteet 51 + 4
<p>Tämä opinnäytetyö tehtiin Arffman Consulting Oy:n toimeksiannosta. Toimeksiantona oli suunnitella ja toteuttaa internetpalvelu myynnin johtamisen tueksi. Internetpalvelun avulla asiakasyritykset voivat seurata myyjien luomia ja ylläpitämiä myyntiprosesseja, eli tietoja, joita asiakkaista kerätään asiakassuhteen aikana. Palvelun kohderyhmänä ovat suomalaiset yritykset, jotka harjoittavat tuotteiden tai muiden palveluiden myyntiä.</p> <p>Internetpalvelu toteutettiin PHP-ohjelmointikielillä käyttäen MySQL-tietokantoja. Itse ohjelmointi tehtiin Notepad++-tekstieditorilla ja tietokantojen hallinta puolestaan phpMyAdminilla. Testaaminen suoritettiin PHP:n versiolla 5.2.1.</p> <p>Koska internetpalvelusta ei ollut valmiina mitään aikaisempaa versiota, eteni sen kehittäminen erilaisten suunnittelupalavereiden muodossa piirustuksiin ja vaadittavien ominaisuuksien listaan. Näiden pohjalta varsinainen ohjelmointi voitiin aloittaa.</p> <p>Ohjelmointi eteni sykleissä, joiden alussa määriteltiin, mitä uudelta versiolta vaadittiin. Tämän jälkeen muutokset toteutettiin ja testattiin. Lopuksi syklin muutokset arvioitiin, ja päätettiin, tarvitseeko internetpalveluun tehdä jotain uusia muutoksia. Tällä tekniikalla saavutettiin mm. se, että internetpalvelusta oli aina saatavilla toimiva versio, jota olisi voinut esitellä siitä kiinnostuneille henkilöille.</p> <p>Opinnäytetyön aikarajoissa internetpalvelusta ei saatu vielä valmiiksi tarpeeksi laadukasta versiota. Näin ollen sitä ei kannata alkaa vielä laajasti tarjoamaan yrityksiensä käyttöön. Tämä johtui projektin suuruudesta, koska kokonaisen internetpalvelun toteuttaminen on suuri kokonaisuus. Internetpalvelusta kuitenkin saatiin valmiiksi ensimmäinen toimiva versio, aivan kuten opinnäytetyön suunnitteluvaiheessa oli määritelty tehtävän. Valmistunutta versiota voisi jo tarjota jollekin yritykselle testikäyttöön.</p>	
Kieli	Suomi
Asiasanat	php, MySQL, internetohjelmointi, web-sovellus, tietokanta, asiakkuudenhallinta, CRM
Säilytyspaikka	<input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Business	Degree Programme Data Processing
Author(s) Petri Leinonen	
Title Sales Management Internet Service	
Optional Professional Studies Programming	Instructor(s) Veli-Pekka Piirainen
	Commissioned by Arffman Consulting Oy (Kimmo Nikkanen)
Date Autumn 2007	Total Number of Pages and Appendices 51 + 4
<p>The purpose of this thesis commissioned by Arffman Consulting Oy was to build an internet program to help with sales management. The customer companies can use the program to track their sales processes, created and updated by their sales persons. The program was designed to suit for most Finnish companies, that sell products or other services.</p> <p>The program was built and tested by using PHP programming language version 5.2.1. The programming was done by using Notepad++ text editor. The Storing of all the information was accomplished by using MySQL-database and the database was managed with phpMyAdmin.</p> <p>Since there was no earlier version of the program available, the production began with planning meetings and blueprints, where the needed features were described. After the blueprints were made, the actual programming started.</p> <p>The programming was made in cycles. Before each cycle, the needed features were defined, then implemented and tested. After each cycle, the changes were evaluated if there were any changes or new features needed. With this technique, there was always a working version of the program that could have been presented to someone who was interested about it.</p> <p>Within the thesis time limit, the final version of the program was not completed. This, however, was noted before the thesis was started, because developing an entire internet service is a large entity. Since the program does not represent the final quality yet, it is not recommended to be offered for paying customers. However, the complete working version of the program could be offered to companies for testing purposes.</p>	
Language of Thesis	Finnish
Keywords	php, MySQL, internet programming, database, sales management, CRM
Deposited at	<input checked="" type="checkbox"/> Kaktus Database at Kajaani University of Applied Sciences <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

ALKUSANAT

Valitsin tämän opinnäytetyön aiheeksi siksi, koska käytännön osuutena tehty ohjelmistoprojekti kuulosti mielenkiintoiselta toteuttaa ja se myös tuntui sopivan laajalta ja opettavaiselta kokonaisuudelta. Tämän tyyppiset projektit ovat myös hyvä lisä ansioluetteloon.

Opinnäytetyön aihe löytyi työharjoittelun pohjalta muodostuneiden henkilökontaktien kautta. Suoritin työharjoittelun vuoden 2006 kesän ja talven aikana Movie Game Designen palveluksessa, jolloin tutustuin Kimmo Nikkaseen. Tuolloin hän oli vielä CR-Keskuksen palveluksessa yhteys- ja kehityspäällikkönä. Hän toimi työharjoittelussa teknisenä neuvonantaja ja ohjaajana, johon pystyin olemaan yhteydessä ongelmien ilmaantuessa. Työharjoittelun loppupuolella Kimmo siirtyi Arffman Consulting Oy:lle toimitusjohtajaksi.

Työharjoittelun jälkeen olen työskennellyt opiskelun ohessa Arffman Consulting Oy:llä erilaisten ohjelmistoprojektien parissa. Tämän vuoksi opinnäytetyön käytännön osuutena luotu ohjelmistoprojekti eteni jo rutiinilla. Aikaisemmista ohjelmistoprojekteista voisi mainita mm. CV-palvelun, mikä on käytössä yrityksen kotisivuilla. Tämän lisäksi olen valmistanut Arffman Consulting Oy:lle erilaisia ohjelmia yrityksen sisäiseen käyttöön.

Kiitokset Arffman Consulting Oy:n toimitusjohtaja Kimmo Nikkaselle, että sain tehdä tämän projektin opinnäytetyönä, ja lisäksi kiitokset kaikin puolin hyvästä yhteistyöstä. Kiitokset myös Arffman Consulting Oy:n markkinointivastaava Aleksi Partaselle, koska sain hyödyntää hänen valmiiksi suunnittelemaa sivun ulkoasua. Tämä säästi minulta paljon aikaa kun pystyin lähes täysin keskittymään opinnäytetyön käytännön osuutena tehdyn internetpalvelun toimintalogiikan tekemiseen.

19.11.2007 Kajaani

Petri Leinonen

SYMBOLILUETTELO

Apache	Avoimeen lähdekoodiin perustuva palvelinohjelma.
ASP.NET	Active Server Pages .NET
Blogi	Verkossa julkisesti nähtävä päiväkirja, johon joko yksi tai useampi henkilö kirjoittaa mietteitään. Aiheet vaihtelevat laajasti, mutta esimerkkeinä mainittakoon henkilökohtaiset päiväkirjat ja ohjelmistoprojektit.
Blue Book	Suomen johtava erikoishakemistojen julkaisija ja kustantaja. (Blue Book)
FTP	File Transfer Protocol Tiedostojen siirtokäytäntö, jota käytetään tiedostojen siirtämiseen.
HTML	HyperText Markup Language
IDE	Integrated Development Environment Kehitysympäristö, joka tyypillisesti sisältää editorin, kääntäjän ja virheenkäsittelijän. (Webopedia.)
IIS	Internet Information Services Microsoftin kehittämä www-palvelinohjelmisto.
IMDb	The Internet Movie Database Erittäin suosittu internetsivusto, josta löytyy tietoa useimmista elokuvista ja tv-sarjoista. http://www.imdb.com/
IRC-galleria	Suomalainen ihmisten verkottumispalvelu, johon kukin käyttäjä voi tehdä oman yksilöllisen sivun, missä mainostaa itseään. http://irc-galleria.net/

JavaScript	NetScape Communicationsin Corporationin kehittämä komentosarjakieli.
JSP	JavaServer Pages
Magic_quotes	PHP:n asetus, joka päälle kytkettynä automaattisesti lisää sivulta toiselle välitettäviin parametreihin kenoviiva-merkkejä, jos muuttujan sisällössä esiintyy heittomerkkejä. Tämän ominaisuuden tarkoitus on tehdä SQL-lauseista turvallisempia aloittelijoille vaikeuttamalla SQL-injektion tekemistä. (PHP: Hypertext Preprocessor a.)
MySpace	Kansainvälinen ihmisten verkottumispalvelu, johon kukin käyttäjä voi tehdä oman yksilöllisen sivun, missä mainostaa itseään. Monet yritykset, yhteisöt, artistit ja yksityiset henkilöt ovat rekisteröityneet näille sivuille. http://www.myspace.com/
MySQL	Ilmainen SQL-tietokantojen hallintajärjestelmä.
phpMyAdmin	Ilmainen internetselaimessa käytettävä MySQL-tietokannan hallintatyökalu.
SQL	Structured Query Language IBM:n kehittämä standardoitu kyselykieli, jolla relaatiotietokaan voidaan tehdä erilaisia hakuja, muutoksia, lisäyksiä ja poistoja.
XML	Extensible Markup Language Tiedon merkkauškieli, jonka avulla mitä tahansa tekstimuotoista dataa voidaan tallentaa ennalta määriteltyjen sääntöjen mukaan.
YouTube	Videopalvelu, johon kuka tahansa palveluun rekisteröitynyt voi lähettää omia videotiedostoja muiden nähtäville. http://www.youtube.com/

SISÄLLYS

1 JOHDANTO	1
2 DYNAAMISET INTERNETSIVUT	3
2.1 Sovelluslogiikka	4
2.2 Ohjelmointikielet	5
2.2.1 PHP	6
2.2.2 ASP.NET	9
2.3 Tietokannat	11
2.3.1 Relaatiotietokannat	12
2.3.2 Vaihtoehtoiset tietokannat	14
2.4 Autentikointi	15
2.5 Dynaamisten sivujen hyödyt	17
2.6 Dynaamisten sivujen riskit	18
2.7 Yhteisöt	19
3 YRITYKSEN SÄHKÖINEN ASIAKUUDENHALLINTA	20
3.1 Myyntiprosessi	21
3.2 Asiakkuudenhallintajärjestelmän kehittämisen lähtökohdat	22
4 MYYNNIN PIPELINE -INTERNETPALVELU	25
4.1 Taustaa	26
4.2 Suunnittelu	26
4.3 Kohderyhmä	27
4.4 Ohjelmistoarkkitehtuuri	28
4.4.1 Käyttötapauskaaviot	28
4.4.2 Tietokanta	31
4.4.3 Käyttöliittymä	33
4.5 Tietoturva	41
4.6 Kehitystyökalut	42
4.7 Tulevaisuus	43
5 POHDINTA	44
LÄHTEET	46
LIITTEET	52

1 JOHDANTO

Sanaa dynaaminen käytetään usein ilmaisemaan muutosta tai muutettavuutta, kiinteän eli staattisen vastakohtaa. (Korpela 2006.)

Internetohjelmoinnissa dynaaminen toiminnallisuus on koko ajan kasvanut, sillä verkkosivujen tekeminen onnistuu koko ajan helpommin entistä parempien kehitystyökalujen avulla. Näiden kehitystyökalujen ansiosta verkkosivujen kehittäjien ei välttämättä tarvitse enää ohjelmoida sivuja pelkästään tekstipohjaisesti, käyttäen esimerkiksi Windowsin mukana tulevaa muistiota. Sen sijaan he voivat suurelta osin tehdä dynaamisia verkkosivuja graafisen käyttöliittymän avulla. Tämä tapahtuu vetämällä ja pudottamalla eri komponentteja, kuten tietokantayhteyksiä tai erilaisia käyttöliittymän komponentteja, kuten lomakkeita, kuvia, tekstiä ja nappeja. Tietenkin näiden komponenttien taakse täytyy vielä itse ohjelmoida toimintaa, mutta graafiset käyttöliittymät ovat helpottaneet ja nopeuttaneet työskentelyä silti huomattavasti.

Dynaamisten verkkosivujen kasvuun ovat vaikuttaneet yritykset, jotka tarjoajat palvelintilaa verkkosivuille, eli ns. webhotellit. Webhotellit tarjoavat nykyään todella halpaa tai joissain tapauksissa jopa ilmaista palvelintilaa verkkosivuille, joissa on tietokantatuki ja tuki muillekin kuin staattisille HTML-tiedostoille, kuten PHP-tiedostoille. Jotkut internetpalveluntarjoajat tarjoavat laajakaistayhteyksien kylkiäisenä palvelintilaa. Lisäksi verkkosivujen teettäminen ammattilaisilla on nykyään melko edullista myös pienille yrityksille. (Louhi.)

Dynaamisten verkkosivujen avulla käyttäjän selauselämystä voidaan parantaa, tekemällä sovelluksista joustavia ja mukavia käyttää. Etenkin dynaaminen sisällönhallinta on monesti käyttäjien mieleen, sillä sen avulla sivuille voidaan luoda uutta sisältöä ilman, että päivitystiedostoja täytyy siirtää FTP-ohjelmalla palvelimelle. Tämän ansiosta sivuston päivittämisestä vastaavalta henkilöltä ei vaadita mitään erityistä osaamista.

Ilman dynaamista toimintaa verkkosivuilla, mm. sellaisten erittäin suosittujen internetsivustojen, kuten YouTube, IMDb, MySpace ja IRC-Gallerian toiminta olisi täysin mahdotonta, koska sivustojen toiminta perustuu suuresti käyttäjien itse luomaan sisältöön.

Opinnäytetyön toimeksiantaja oli kajaanilainen Arffman Consulting Oy, joka tarjoaa koulutuspalveluita useilla eri koulutusaloilla julkiselle ja yksityiselle sektorille. Yrityksen yhteyshenkilönä toimi koko opinnäytetyön ajan toimitusjohtaja Kimmo Nikkanen. (Arfcon.)

Opinnäytetyön käytännön osiona oli Myynnin pipeline -internetpalvelun suunnittelu ja toteutus. Myynnin pipeline -internetpalvelu on mynninhallintaan tarkoitettu internetsovellus, jota käytetään internetselaimessa. Sovelluksen avulla yritykset voivat seurata omien myyjien luomia ja ylläpitämiä myyntiprosesseja eli tietoja, joita asiakkaista kerätään asiakassuhteen aikana. Sovellus mahdollistaa myös erilaisia raportteja, joita voidaan luoda asiakastietojen pohjalta.

Opinnäytetyön käytännön osuus toteutettiin kokonaisuudessaan käyttäen PHP-ohjelmointikieltä ja tietojen varastointiin käytettiin MySQL-tietokantoja. Nämä työkalut valittiin sen vuoksi, koska molemmille löytyi valmiiksi tuki Arffman Consulting Oy:n palvelimelta ja molemmista oli jo ennestään hyviä kokemuksia.

Opinnäytetyön teoriaosuuden on tarkoitus kertoa lukijalle, mitä dynaamiset internetsivut ovat, miten ne toimivat, mistä eri osista ne koostuvat ja mitä kaikkea ne mahdollistavat. Työssä käsitellään hieman myös yrityksen sähköistä asiakkuuksien hallintaa, jotta lukija ymmärtäisi paremmin, mikä päättötyön käytännön osion, eli Myynnin pipeline -internetpalvelun tarkoitus on.

Opinnäytetyön ei ole kuitenkaan tarkoitus opettaa lukijaa ohjelmoimaan dynaamisia internet-sivuja, vaan paremminkin ohjata lukijan mielenkiinto opettelemaan asiaa itse, selvittämällä muutamia oleellisia asioita, joita usein tarvitaan, kuten asiakas- ja palvelin -käsitteet, dynaamisten internetsivujen toimintalogiikka ja käyttäjien autentikointi.

2 DYNAAMISET INTERNETSIVUT

Internetsivujen staattista luonnetta on vuosien varrella parannettu useilla eri tekniikoilla, joiden avulla sivujen toimintaa on voitu muokata dynaamiseksi. Nämä tekniikat voidaan ryhmitellä asiakas- ja palvelintekniikoihin. Asiakas- ja palvelin -sanat viittaavat siihen, missä web-sovelluksen toiminta suoritetaan. (Rantala 2005, s. 5 - 6.)

Loppukäyttäjille, eli internetsivujen käyttäjille dynaamisuus voi ilmetä esimerkiksi siinä, jos eri käyttäjät tai käyttäjäryhmät voivat nähdä saman sivun erilaisena, tai sivut ovat jollain tavalla vuorovaikutteisia. Yleisiä dynaamisia ominaisuuksia ovat mm. keskustelupalstat, sähköpostijärjestelmät, internetsivuilla olevat kalenterit, nettikaupat, sanahakukentät, vieraskirjat, blogit ja monet muut ominaisuudet. Myös useimpien yritysten, yhteisöjen ja yksityisten henkilöiden verkkosivut, joiden sisältö vaihtuu usein, päivitetään jonkinlaisen dynaamisen sisällönhallintatyökalun kautta, yksinkertaisesti kirjautumalla sisään internetsivun kautta ja kirjoittamalla uutta sisältöä, ilman, että missään vaiheessa päivitystiedostoja täytyy lähettää palvelimelle erillisellä FTP-ohjelmalla. (Heinisuo & Rauta 2007, s. 11.)

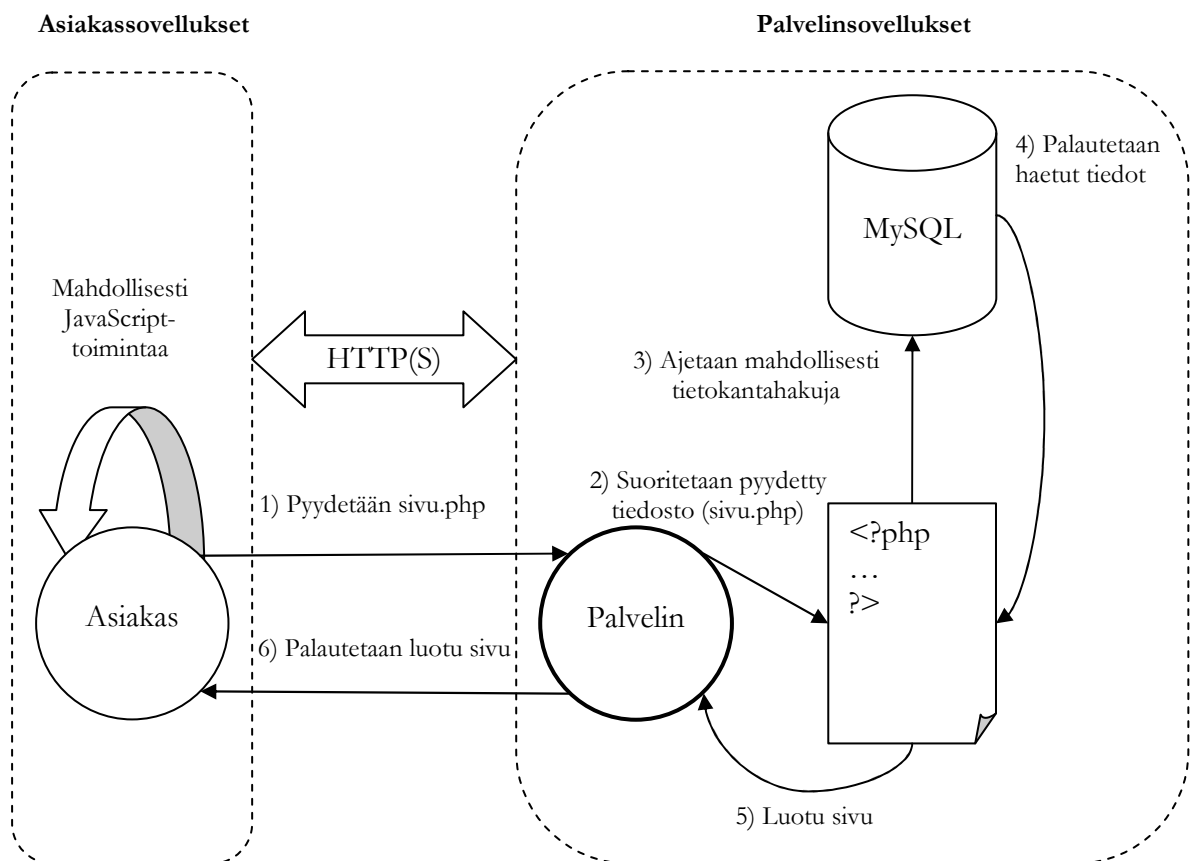
- a) **Asiakastekniikoilla** (JavaScript, Flash, Java applet) toteutettujen sovellusten toiminta suoritetaan käyttäjän koneella, ts. käyttäjän internetselaimessa. Tämä on hyvä asia internetsivun käyttäjän kannalta, sillä sivun käyttäminen voi olla nopeampaa ja esteettisempää kuin puhtaasti palvelintekniikalla tehtyjen sovellusten. Lisäksi tietoa ei välttämättä tarvitse liikutella internetin yli kahden eri tietokoneen välillä niin usein. (Rantala 2005, s. 5 - 6.)
- b) **Palvelintekniikoilla** (ASP.NET, JSP, PHP) toteutettujen sovellusten toiminta suoritetaan palvelimella, eli sillä koneella, mistä käyttäjä lataa internetsivun omalle koneelleen. (Rantala 2005, s. 7 - 8.)

Käytännössä yllämainittuja tekniikoita kuitenkin käytetään useimmiten yhdessä, sillä sovellusten toiminta yleensä vaatii molempien käyttöä. Asiakastekniikalla voidaan tehdä esimerkiksi käyttöliittymästä parempi, lisäämällä vaikka lähetettävien tietojen esitarkistuksia JavaScriptillä ennen kuin niitä yritetään lähettää internetin yli palvelimelle käsiteltäväksi. Sovelluksen toimintalogiikka ja tiedon käsittely puolestaan voidaan hoitaa palvelintekniikoilla kuten PHP:llä. (Rantala 2005, s. 6 - 8.)

2.1 Sovelluslogiikka

Dynaamisten internetsivujen sovelluslogiikka suoritetaan lähes aina palvelintekniikoilla. Syy tähän on se, että sovelluksen toiminnan kannalta oleelliset asiat, kuten tietokannasta haetut tiedot sijaitsevat erillisessä järjestelmässä palvelimella. Tämän tiedon lähetyksen kokonaisuudessaan asiakkaan selaimen ei olisi kovin käytännöllistä tiedonsiirron hitauden ja mahdollisesti tietokannan tietojen arkaluonteisuuden vuoksi. (Rantala 2005, s. 7.)

Alla (Kuvio 1.) on kuvattu tyypillinen dynaamisen verkkodokumentin toimintaperiaate PHP-dokumentin avulla. Asiakas pyytää palvelimelta tiedostoa sivu.php (1). Palvelin huomaa, että tiedostossa on .php-pääte, joten tiedosto ajetaan PHP-tulkkin läpi (2). Jos tiedostossa on tietokantahakuja, ne suoritetaan (3) ja tietojen pohjalta luodaan HTML:ää sisältävä tiedosto, johon ohjelmoitu tieto kirjoitetaan (4). Kun PHP-tulkki on käsitellyt tiedoston (5), lähetetään se internetin yli asiakkaan selaimen (6). (Rantala 2005, s. 4 - 7.)



Kuvio 1. Asiakas ja palvelinsovelluksien toiminnan yleiskuvaus. (Rantala 2005, s. 4.)

Kuten aiemmin (Kuvio 1.) kuvatussa prosessissa ilmenee, asiakkaan pyytämä dokumentti luodaan jokaisen sivun latauksen yhteydessä uudestaan. Eli palvelimella ei ole valmista dokumenttia, joka lähetettäisiin asiakkaalle sellaisenaan, kuten puhtaan HTML-tiedoston tapauksessa. Tämä on suuri ero ja etu perinteisiin staattisiin verkkodokumentteihin verrattuna, jotka ovat valmiiksi luotuna palvelimella, koska dokumenttiin voidaan aina hakea suoritusvaiheessa esimerkiksi ajan tasalla olevat tekstit tietokannasta. (Rantala 2005, s. 4.)

Asiakkaalle palautettu dokumentti voi sisältää ihan mitä tahansa tietoa, mitä staattinen HTML-dokumenttikin voi sisältää, kuten esimerkiksi tekstiä, kuvia, lomake-elementtejä ja JavaScripteja.

Palvelintekniikoilla toteutetun sovelluslogiikan tekemiseen on saatavilla lukuisia eri vaihtoehtoja, joista on alla lueteltu muutamia suosittuja.

- a) ASP.NET (Active Server Pages) on Microsoftin kehittämän ASP:n uudempi versio, joka perustuu Microsoftin .NET –arkkitehtuuriin. ASP.NET-sivujen ohjelmointi muistuttaa enemmän Windows-ohjelmien tekemistä, sillä ne ovat olio- ja tapahtumapohjaisia. ASP.NET-sivuja voidaan ohjelmoida usealla eri kielellä, mutta tyypillisimpiä ovat C# ja Visual Basic.NET. (Rantala 2005, s. 8; Yank 2002.)
- b) JSP (JavaServer Pages) on Sun Microsystemsin kehittämä tekniikka, joka toimii kuten PHP. JSP-dokumentit käännetään Java-servleteiksi ennen suoritusta. (Kurkinen 1998; Rantala 2005, s. 7.)
- c) PHP (PHP: Hypertext Preprocessor) on useilla alustoilla toimiva vapaaseen lähdekoodiin perustuva tekniikka. Ohjelmointikielen syntaksi muistuttaa C-kieltä. (Rantala 2005, s. 9.)

2.2 Ohjelmointikielien

Dynaamisten internetsivujen ohjelmointiin on mahdollista käyttää useita eri ohjelmointikieliä ja tekniikoita, kuten ASP.NET, JSP ja PHP. Valintaan vaikuttavat monesti lähinnä ohjelmoinnin mieltymys jonkin ohjelmointikielen kehitystyökaluihin, työn tilaajan laitteistossa valmiiksi tuetut ohjelmointikielien sekä budjetti, ja mahdolliset kytkökset aikaisemmin tehtyihin järjestelmiin. (Web.Blazonry.)

Microsoftin kehittämä Visual Studio (ASP.NET) ja Sun Microsystemsin NetBeans (JSP) kehitystyökalut esimerkiksi ovat olleet todella käyttäjäystävällisiä ohjelmoijan kannalta ja tarjoavat lisäksi hyvät debug, eli virheenjäljitystoiminnot. Myös PHP:n kehitystyökalut ovat hyviä, mutta ne eivät vielä aivan yllä käytettävyydeltään Visual Studion tai NetBeansin tasolle. Varsinkin virheenjäljitys-työkalut kaipaavat parannusta. Kehitystä on kuitenkin tapahtunut koko ajan tämän asian hyväksi. (Hull.)

Valitsipa sitten minkä ohjelmointikielen tahansa, ei kielen valinta aseta juuri rajoituksia sisällön luontiin. Kaikki yllämainitut ohjelmointikielät ovat vielä vahvasti tuettuja ja kaikille löytyy aktiivisia yhteisöjä, eikä niiden kehitystä todennäköisesti tulla lopettamaan ihan lähiaikoina.

Seuraavassa on kuvattu lyhyesti kaksi hyvin suosittua ohjelmointikieltä. Avoimeen lähdekoodiin perustuva PHP ja Microsoftin kehittämä ASP-ohjelmoinnin seuraaja ASP.NET. Molemmista löytyy todella paljon materiaalia internetistä, kuten aloittelijan oppaita ja esimerkkisovelluksia, jos kiinnostusta vain riittää kielten opetteluun.

2.2.1 PHP

PHP on avoimeen lähdekoodiin perustuva HTML-dokumenttien sisään upotettava ohjelmointikieli. Kehitystyön aloitti Rasmus Lerdorf vuonna 1994 ja ensimmäinen versio oli nimeltään ”Personal Home Page Tools”. Kielen lyhenne PHP on säilynyt vuosien saatossa, vaikka lyhenteen merkitys onkin vaihtunut useamman kerran. Nykyään kuitenkin PHP tulee sanoista ”PHP: Hypertext Preprocessor”. (Meloni 2005, s. Introduction xvi; Rantala 2005, s. 9.)

Ohjelmointikielenä PHP on erinomainen niin aloittelijoille, kuin kokeneemmillekin ohjelmoijille. Aloittelijoille se tarjoaa matalan aloituskynnyksen ohjelmoinnin harjoitteluun, koska se on ns. heikosti tyypitetty kieli. Sen kokeileminen ja yleensäkin sen käyttäminen on täysin ilmaista, sillä tiedostoja voidaan ajaa Apache-verkkopalvelimella, joka perustuu avoimeen lähdekoodiin ja on näin ollen täysin ilmainen. PHP:llä on aktiivinen yhteisö, joten internetistä varmasti löytyy kullekin jokin keskustelupalsta, mistä voi kysyä neuvoa jos tulee ongelmia. PHP on tasaiseen tahtiin saanut uusia versioita, joissa siihen on lisätty uusia ominaisuuksia, kuten parannettu tukea olio-ohjelmoinnille, lisätty kiinteä tuki XML:lle ja parannettu suorituskykyä. (Meloni 2005, s. Introduction xviii; Rantala 2005, s. 9 - 10.)

Kokeneemmat ohjelmoijat varmasti arvostavat PHP:n version 5 myötä tulleita uudistuksia, kuten versioon 4.x nähden parempaa tukea olio-ohjelmoinnille. Lisäksi kielen syntaksi on erittäin helppo oppia, jos on ennestään ohjelmoinut C:llä tai jollain muulla ohjelmointikielillä. (Meloni 2005, s. Introduction xx; Trachtenberg 2004.)

Alla on esitetty (Kuvio 2.) kuinka PHP:llä voidaan tulostaa ruudulle tekstiä ja kuinka ohjelmakoodia voidaan upottaa HTML-kielen sekaan. Samassa esimerkissä näkyy myös PHP:lle tyypillinen syntaksi, jossa ohjelmakoodi kirjoitetaan `<?php` ja `?>` -tagien väliin. Koska PHP on ns. heikosti tyypitetty kieli, muuttujia ei tarvitse erikseen määritellä ja kertoa mitä tietoa niihin tullaan tallentamaan, vaan PHP osaa miettiä tietotyypit itse. Muuttujien edessä on aina dollarimerkki ja lauseet loppuvat aina puolipisteeseen. (Rantala 2005, s. 9 - 10.)

```
<html>
<head>
<title>HTML-sivu, jonka sisään on upotettu PHP-kieltä</title>
</head>
<body>
<?php
// Tallennetaan string-tyyppinen tieto muuttujaan
$muuttuja = "Terveppä terve!";
// Tulostetaan muuttujan sisältö ruudulle
echo $muuttuja;
?>
</body>
</html>
```

Kuvio 2. Muuttuja ja muuttujan tulostaminen ruudulle PHP-sovelluksessa.

Yllä listattu ohjelmakoodi (Kuvio 2.) näyttäisi ruudulla vain tekstin: ”Terveppä terve!”. Kahdella kauttaviiva-merkillä alkavat rivit ovat ns. kommenttirivejä, joihin ohjelmoija voi kirjoittaa vaikka muistiinpanoja. Nämä rivit eivät näy ohjelmaa suoritettaessa millään tavalla. Yhdessä PHP-tiedostossa voi myös käyttää useita koodiblokkeja, joiden välissä voi myös olla normaalia HTML:ää, kunhan PHP-koodi vain on aina ympäröity niiden osoittamilla tageilla, jotka esiteltiin aiemmin. (2K mediat a.)

PHP:llä voidaan pienellä koodimäärällä saada aikaiseksi dynaamista toimintaa internetsivuille, eikä ohjelmointiin välttämättä tarvita edes muuta ohjelmaa kuin tekstieditori. Jo pelkällä Windowsin muistiolla voi ohjelmoida, mutta ohjelmakoodia värikoodaava sovellus, kuten avoimeen lähdekoodiin perustuva Notepad++, helpottaa työskentelyä huomattavasti. Saatavilla on myös kokonaisia integroituja kehitysympäristöjä (IDE), jotka sisältävät ainakin koodieditorin, kääntäjän ja virheenjäljitystoiminnot. Suosituimmat kehitysympäristöt, kuten Adobe Dreamweaver, PHPeD ja phpDesigner ovat kuitenkin maksullisia. PHP:n heikkoudeksi voisikin ehkä sanoa sen, ettei siihen ole tarjolla ilmaiseksi aivan niin hyvää kehitysympäristöä, kuin esimerkiksi ASP.NET:lle tai JSP:lle. (Notepad++; php-editors.)

PHP-tiedostojen ajamiseen vaaditaan palvelinohjelma, jossa on tuki .php-tiedostoille. Suosituin tällainen palvelinohjelmisto on Apache, jonka voi kuka tahansa ladata omalle koneelle Apachen kotisivuilta <http://www.apache.org/>. Apache toimii kaikilla yleisimmillä käyttöjärjestelmillä, kuten Windows ja Unix. Apachessa ei kuitenkaan ole suoraan asennuksen jälkeen tukea PHP:lle, joten siihen täytyy asentaa erikseen PHP. Sen voi ladata PHP:n virallisilta kotisivuilta osoitteesta <http://www.php.net/>. (Heinisuo & Rauta 2007, s. 26.)

Toinen vaihtoehto on ajaa ohjelmia jollakin valmiilla palvelimella. Tämä on aloittelijalle aavistuksen helpompi vaihtoehto. Internetistä löytyy lukuisia webhotellien tarjoajia, joita voi vertailla esimerkiksi osoitteessa <http://www.webhotellit.com/>. (2K mediat b; Heinisuo & Rauta 2007, s. 14; MVnet.)

Aikaisemmin esitellyn koodiesimerkin (Ohjelmalistaus 1.) ajamiseen ei tarvitse muuta, kuin tallentaa ohjelmakoodi tiedostoon, nimetä se .php-tiedostopäätteellä ja lähettää se jollekin palvelimelle, mikä tukee .php-tiedostoja. Ohjelma suoritetaan yksinkertaisesti lataamalla palvelimelle lähetetty tiedosto internetselaimella, jolloin palvelin lähettää rakennetun tiedoston asiakkaan selaimeen. Jos katsot sivun lähdekoodin, huomaat, että siinä ei näy ollenkaan ohjelmakoodia, mikä kirjoitettiin ohjelman tiedostoon, vaan näet ainoastaan PHP-tulkkin tuottaman HTML-tiedoston koodin. (2K mediat a; 2K mediat b.)

Yhteenvetona voidaan sanoa, että PHP soveltuu hyvin, niin harrasteprojekteihinkin kuin kaupallisiin sovelluksiin. Se on laajasti tuettu, tehokas, yksinkertainen mutta silti monipuolinen ja helposti opeteltavissa oleva kieli riippumatta siitä, onko aikaisempaa kokemusta ohjelmoinnista. (Meloni 2005, s. Introduction xx.)

PHP:n uusin versio on kirjoitushetkellä 5.2.4, mutta siitä on kehitteillä myös versio 6, tosin sen tuomista muutoksista ei ole vielä tarkkaa tietoa. Muutamia merkittävimpiä kaavailtuja uudistuksia ovat: siirtyminen unicode-merkistöön, olio-ohjelmoinnin parantaminen lisäämällä nimiavaruudet ja ns. magic_quotesien poistaminen, mitkä ovat aiheuttaneet ongelmia siirrettäessä PHP:llä tehtyjä sovelluksia ympäristöihin, joissa on erilaiset palvelinasetukset. (Heinisuo & Rauta 2007, s. 396 - 398; Jeroen; PHP: Hypertext Preprocessor a; PHP: Hypertext Preprocessor b.)

2.2.2 ASP.NET

ASP.NET on Microsoftin kehittämä tekniikka, mikä mahdollistaa dynaamisten internetsivujen luonnin millä tahansa .NET-ympäristön ohjelmointikielellä, kuten C#:lla ja Visual Basic.NET:llä. (Inkinen 2003, s. 4.)

ASP.NET-sivut toimivat pääpiirteittäin hyvin pitkälti samalla tavalla kuten PHP-sivutkin. Asiakkaan pyytäessä jotain sivua palvelimelta, palvelin (IIS) tutkii tiedostossa olevan tiedostopäätteen. Jos se on jokin ASP.NET:n käyttämä tiedostopäätte, ajetaan tiedosto, ja palautetaan lopputulos käyttäjälle. (Inkinen 2003, s. 59 - 60; Lee 2005 a.)

ASP.NET:n yleisimmin käytetty kehitysympäristö Microsoft Visual Studio tarjoaa helppokäyttöisen, mutta silti monipuolisen kehitysympäristön, jossa on valmiiksi ohjelmoituja komponentteja. Nämä valmiit komponentit säästävät paljon aikaa, jos vertaa siihen, että itse kirjoittaisi komponentit tekstipohjaisesti tyhjästä joka kerta uudestaan. Komponenttien käyttöön ei tarvitse muuta kuin vetää ja pudottaa niitä sivuille ja kirjoittaa niiden taakse jotain toimintaa. Visual Studioon voi myös tehdä itse omia komponentteja, mitkä voivat olla käteviä siinä tapauksessa, jos niille on tarvetta usein. Koska vetämällä ja pudottamalla lopputulos on nähtävillä välittömästi. Tämä voi toimia aloittelijoille pienenä lisämotivaationa, kun saa nopeasti toimintaa aikaiseksi. (Lee 2005 b.)

ASP.NET tarjoaa mm. tuen yli 25 eri ohjelmointikielelle, melko matalan aloituskynnyksen ohjelmoinnin aloittamiseen, monipuolisen verkkosivuille upotettavan tekniikan, täyden tuen oliopohjaiselle ohjelmoinnille, laajan luokkakirjaston hyvällä dokumentaatiolla ja vaikkei kehitysympäristö varsinaisesti ohjelmointikieleen liitykään, tarjoaa Microsoft myös Visual Studio Express ohjelmointiympäristön maksutta. (Brilliance Web Design 2003; W3Schools.)

ASP.NET on helppo oppia, varsinkin jos on ennestään kokemusta jostain .NET-ohjelmointikielestä, kuten C#:sta tai Visual Basic.NET:stä. Tämä siksi, koska ASP.NET-sivujen ohjelmointi muistuttaa enemmän perinteisten Windows-ohjelmien tekemistä. (Software Business Competence a.)

Toisin kuin PHP, ASP.NET:ssä yleisimmin käytetyt ohjelmointikielät C# ja VB.NET ovat molemmat vahvasti tyyppitettyjä kieliä, eli muuttujille täytyy kertoa esittelyvaiheessa millaista tietoa niihin tullaan tallentamaan. (Inkinen 2003, s. 39.)

Alla on kuvattu (Kuvio 3.) sama ohjelmaesimerkki C#:lla, joka aiemmin (Kuvio 2.) kuvattiin PHP:n esittelyluvussa. Ohjelmakoodi kirjoitetaan `<% ja %>` -tagien väliin. Aivan kuten PHP:llä, voidaan tämä ohjelmakoodi lisätä mihin tahansa HTML-tiedostoon, kunhan tiedoston pääte muutetaan ASP.NET:n tiedostopäätteeksi `.aspx`. (W3Schools.)

```
<html>
<head>
<title>HTML-sivu, jonka sisään on upotettu ASP.NET:ä</title>
</head>
<body>
<%
// Tallennetaan string-tyyppinen tieto muuttujaan
String muuttuja = "Terveppä terve!";
// Tulostetaan muuttujan sisältö ruudulle
Response.Write(muuttuja);
%>
</body>
</html>
```

Kuvio 3. Muuttuja ja muuttujan tulostaminen ruudulle ASP.NET-sovelluksessa.

ASP.NET:n heikkous on se, että jos ohjelmoija haluaa julkaista tekemänsä tuotoksen internetissä, täytyy hänen usein maksaa enemmän webhotellista kuin PHP:n tapauksessa. Tämä siksi, koska ASP.NET sivut vaativat IIS-serverin, joka puolestaan vaatii maksullisen Windowsin lisenssin. Varsinkin yksityisille harrastelijoille tämä voi olla merkittävä eroavaisuus, jonka seurauksena PHP on houkuttelevampi vaihtoehto. Yrityksille kuitenkin muutaman euron hintaero kuukaudessa tuskin vaikuttaa valintaan. (Eaton 2007.)

Yhteenvetona voidaan sanoa ASP.NET:stä, että se soveltuu PHP:n tapaan hyvin sekä yksityisiin harrasteprojekteihin kuin kaupallisiinkin sovelluksiin. Jollekin ohjelmoijalle jo pelkkä Visual Studiolla ohjelmointi voi olla valintaperuste PHP:n ylitse, jos sovelluksen voisi tehdä kummalla kielellä vain. Koska Microsoft tarjoaa Visual Studiosta ns. Express Editioneita täysin veloittamatta, ei PHP:n valintaa voi perustella pelkällä hinnalla. (msdn.)

ASP.NET:n uusin versio on kirjoitushetkellä 2.0 mutta siitä on saatavilla myös 3.5 Beta-versio. Uusinta 3.5 Beta-versiota tosin ei vielä suositella tuotantokäyttöön, koska muutokset ovat mahdollisia. Merkittävimpiä muutoksia, joita Visual Studio 2008 yhdessä .NET 3.5:n kanssa tulevat tarjoamaan ovat mm. integroitu tuki CSS-tyylitiedostoille, parempi tuki AJAX-tekniikalle ja useita eri parannuksia ohjelmointikieleen, joilla pyritään yksinkertaistamaan ohjelmointia. (Guthrie 2007.)

2.3 Tietokannat

Tietokannoilla tarkoitetaan jotain tapaa varastoida tietoa, mutta tietotekniikasta puhuttaessa sillä yleensä tarkoitetaan, joko normaalia tekstitiedostoa, johon on tallennettu tietoa jossain ennalta määritellyssä formaatissa, tai tiedon varastointiin erityisesti kehitettyä järjestelmää. (Rantala 2005, s. 251 - 252; Wikipedia a.)

Koska tietokantojen koot voivat vaihdella muutaman rivin mittaisista tiedostoista miljoonien tietorivien varastoiksi, jotka vievät jopa teratavuittain tietoa, on useimmissa tapauksissa järkevää käyttää ns. relaatiotietokantoja. Niiden avulla erityyppiset tiedot voidaan tallentaa kompaktimpaan tilaan ja niitä voidaan käsitellä huomattavasti helpommin kuin esimerkiksi puhtaita tekstitiedostoja. (Business Intelligence Lowdown 2007; Heinisuo & Rauta 2007, s. 130 - 131; Wikipedia a.)

2.3.1 Relaatietietokannat

Kaikki suosituimmat tietokannat perustuvat standardoituun relaatiomalliin, joiden määrittely ja käsittely tapahtuvat useimmiten standardoidulla SQL-kyselykielellä. (Rantala 2005, s. 252.)

Relaatietietokannoissa eri asiakokonaisuudet jaotellaan tauluihin, joilla voi olla yhteyksiä toisiinsa. Taulut koostuvat tietoriveistä (tietue), jotka on jaettu erityyppisiin tietosarakkeisiin (kenttä). Tietosarakkeille on määritelty erilaisia tietotyyppisiä tauluja luotaessa, jotka määrittävät, minkä tyyppistä tietoa kuhunkin sarakkeeseen voidaan tallentaa. (Rantala 2005, s. 252 - 253.)

Seuraavassa (Taulukko 1.) on kuvattu tietotyypeistä muutamia kaikkein yleisimmin käytettyjä ja selitetty lyhyesti niiden toimintaa. (Meloni 2005, s. 200 - 201.)

Taulukko 1. Muutamia yleisimpiä MySQL-tietotyyppisiä. (Meloni 2005, s. 200 - 201.)

TIETOTYYPPI	SELITYS
VARCHAR(255)	Merkkijono. Taulua esitellessä sulkuihin voidaan määritellä, kuinka monta merkkiä tähän sarakkeeseen voidaan tallentaa. Merkkijonon maksimipituus on 255 merkkiä.
DECIMAL(8,2)	Numeerinen arvo. Suluissa ole ensimmäinen numero kertoo numeron maksimipituuden ja jälkimmäinen numero desimaaliosuuden lukumäärän. Suluissa oleva ensimmäinen numero voi olla 1-65 ja jälkimmäisen numero voi olla 0-30, muttei kuitenkaan koskaan suurempi kuin ensimmäinen numero.
INTEGER(3)	Kokonaisluku. Numero suluissa kertoo luvun maksimipituuden.
BLOB	Pitkä merkkijono, johon voidaan tallentaa jopa 65535 merkkiä.
DATETIME	Päivämäärän ja kellonajan yhdistelmä.

Lähes mitä tahansa arkimaailman asioita voidaan kuvata tietokannan tauluina. Esimerkiksi Henkilö voisi olla taulu, jolla on kentät: henkilötunnus, ikä, pituus ja paino. Yllä esiteltyjen (Taulukko 1.) tietotyyppien avulla taulun ”Henkilo” luonti tapahtuisi seuraavalla (Kuvio 4.) SQL-komennolla, jonka voisi syöttää jonkin SQL-tietokannan hallintajärjestelmän, kuten phpMyAdminin SQL-komentokonsoliin. (Rantala 2005, s. 253.)

```
CREATE TABLE Henkilo(
henkilotunnus VARCHAR(11) NOT NULL,
ika INT(3),
pituus DECIMAL(3,2),
paino DECIMAL(5,2),
PRIMARY KEY(henkilotunnus));
```

Kuvio 4. Taulun Henkilo luonti SQL-kyselykielellä. (Kiukas.)

Yllä (Kuvio 4.) luotiin taulu nimeltä ”Henkilo”, jonka pääavain, eli uniikki tunnistetieto on henkilotunnus. Taulun tietueeseen

- vaaditaan henkilötunnus, jonka pituus on maksimissaan 11 merkkiä
- saa kirjoittaa kolminumeroisen luvun iäksi
- saa kirjoittaa pituuden metreinä kahden desimaalin tarkkuudella
- saa kirjoittaa painon kolmen kokonaisluvun ja kahden desimaalin tarkkuudella.

Relaatiotietokanta voi sisältää useita tauluja, jotka muodostavat yhdessä tulosjoukkoja. Tulosjoukot vaativat aina jonkin uniikin tunnistetiedon, joka voisi henkilön tapauksessa olla henkilötunnus. Tämän uniikin tiedon avulla, jota kutsutaan myös pääavaimeksi, voidaan käsitellä taulun tietoja esimerkiksi ohjelmakoodissa. (Rantala 2005, s. 253.)

Relaatiotietokantoja käsitellään nykyisin yleensä standardoidulla SQL-kyselykielellä (Structured Query Language), jonka avulla pystytään komentamaan lähes minkä tahansa valmistajan tekemää tietokantajärjestelmää. Kielessä voi esiintyä pieniä valmistajakohtaisia eroja, mutta ne ovat yleensä hyvin samantyyppisiä. (Hillyer; Rantala 2005, s. 252.)

Alla (Kuvio 5.) on esimerkin vuoksi kuvattu, miltä aikaisemmin luotu (Kuvio 4.) taulu näyttäisi phpMyAdminin kautta katsottuna. Tauluun on syötetty kaksi henkilöä.

henkilotunnus	ika	pituus	paino
xxxxxx-xxxx	20	1.75	72.00
yyyyyy-yyyy	32	1.68	65.00

Kuvio 5. Kaksi henkilöä tietokannan taulussa.

2.3.2 Vaihtoehtoiset tietokannat

Relaatiotietokantojen sijaan dynaamisilla internetsivuilla olevaa tietoa voidaan varastoida useilla muillakin tavoilla. Muita tapoja ovat mm. TSV (Tab Separated Values) ja CSV (Comma Separated Values). Tiedot varastoidaan samaan tapaan sarakkeisiin, kuten relaatiotietokannan tauluissa, mutta sarakkeille ei ole määriteltyjä tietotyyppejä. Kaikki tiedot tallennetaan vain tekstimuodossa johonkin tiedostoon palvelimella. (Rantala 2005, s. 147.)

Alla on kuvattu esimerkki TSV-tietokannasta (Kuvio 6.), jossa tietosarakkeet on eroteltu toisistaan tabulaattori-merkein (\t). Tietorivit puolestaan on eroteltu rivinvaihdolla, jotka voidaan tunnistaa ohjelmakoodissa tiedostoa luettaessa. (Rantala 2005, s. 147.)

```
sakari\tkauppakatu 3\t0441231234
ville\tmannerheimintie 2\t0443214321
```

Kuvio 6. TSV-muodossa tallennettu tekstimuotoinen tietokanta.

Vastaavia tietokantoja on useita muitakin, mutta ne toimivat kaikki samalla periaatteella. Tiedot erotetaan jollain ennalta määrätyllä merkillä ja tietorivit useimmiten erotetaan rivinvaihdoin. Joskus tämän tyyppisissä tietokannoissa on myös sarakkeiden nimet ylimmällä rivillä, jotka voisivat esimerkiksi (Kuvio 6.) olla nimi, katuosoite ja puhelinnumero.

Tekstimuotoiset tietokannat ovat kätevä tapa varastoida tietoa internetsovelluksissa, mutta ne eivät ole kovin joustavia. Varsinkin jos internetsovelluksen täytyy usein sekä kirjoittaa, että lukea tietoa tietokantaan, ratkaisu on MySQL tai jokin muu relaatiotietokanta, sillä relaatiotietokannoissa etenkin tietojen lisäys, muokkaus ja poisto ovat helpompia ja joustavampia toteuttaa. (Rantala 2005, s. 177.)

Tekstitiedostomuotoisen tietokannan yksi suuri rajoitus on se, että tietokantaan voi kirjoittaa vain yksi käyttäjä kerrallaan, koska muuten tiedostot voisivat korruptoitua ja muuttua käyttökelvottomiksi. (Software Business Competence b.)

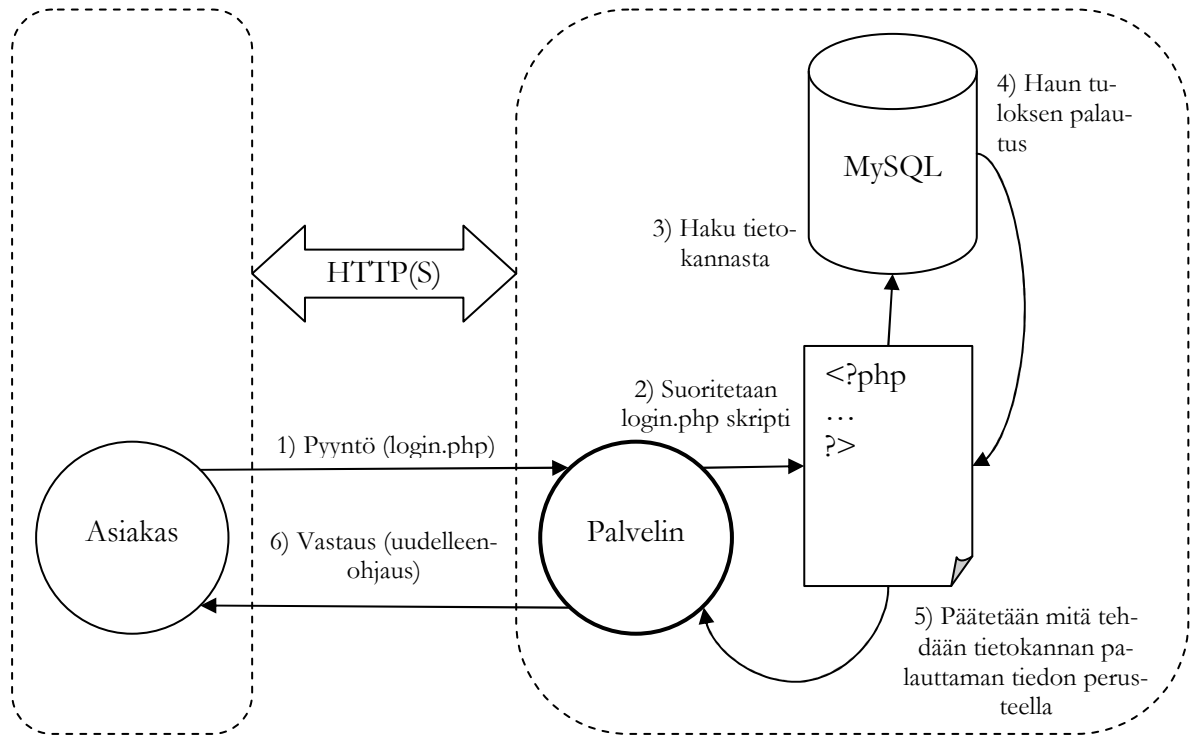
2.4 Autentikointi

Autentikoinnilla tarkoitetaan kaikkia niitä menetelmiä, joilla tiettyihin resursseihin oikeutetut käyttäjät voidaan tunnistaa. (Rantala 2005, s. 230.)

Yleisin verkkosivuilla käytetty autentikointimenetelmä on pyytää käyttäjää kirjoittamaan käyttäjätunnus ja sitä vastaava salasana lomakkeelle, ja verrata näitä tietoja tietokannan tietoihin. Jos vastaavat tiedot löytyvät, niin käyttäjälle näytetään hänelle määritelty sisältö. (Rantala 2005, s. 230.)

Koska HTTP-protokolla on ns. yhteydetön protokolla, täytyy käyttäjän kertoa palvelimelle jokaisen sivunlatauksen yhteydessä omat autentikointitietonsa ja palvelimen täytyy vastaavasti tarkistaa ne aina uudelleen. On kuitenkin huomioitava, että tiedot liikkuvat oletuksena salaamattomana HTTP-protokollalla internetin yli asiakkaan ja palvelimen välillä. Siksi on usein järkevää salata tiedot siirron ajaksi HTTPS-protokollalla, ettei niitä voi kukaan salakuunnella. (Rantala 2005, s. 230.)

Seuraavassa on kuvattu tyypillinen verkkosivuilla käytettävä autentikoinnin menettely kuvion (Kuvio 7.) avulla. Autentikoinnin prosessointi lähtee liikkeelle siitä, kun käyttäjä pyytää (1) palvelimelta jotain skriptiä, joka esimerkin tapauksessa on login.php. Skriptin kutsu voisi tapahtua esimerkiksi klikkaamalla verkkosivun lomakkeella olevaa ”Kirjaudu”-nappia. Samassa käyttäjän kirjautumislomakkeelle syöttämät tiedot: käyttäjätunnus ja salasana, lähetetään palvelimelle käsiteltäväksi. Tämän jälkeen palvelin suorittaa (2) pyydetyn skriptin, jossa tehdään tietokantahaku (3) asiakkaan lähettämällä tiedoilla. Jos tietokannasta löytyi tulos, eli tietue, jossa oli sama käyttäjätunnus ja salasana, palauttaa (4) kysely kyseisen rivin tietokannasta. Tämän kyselyn tuloksen perusteella voidaan käsitellä kirjautuminen (5). Jos tuloksia löytyi (6), niin tunnukset olivat oikeat ja käyttäjä voidaan ohjata määritetylle sivulle. Jos tuloksia puolestaan ei löytynyt (6), käyttäjä voidaan ohjata vaikka takaisin kirjautumisruutuun. (Rantala 2005, s. 232 - 237.)



Kuvio 7. Asiakkaan ja palvelimen välinen tunnuksien tarkistamisen menettely.

Kuten aiemmin jo todettiin, on HTTP-protokolla yhteydetön ja tunnuksien tarkistus täytyy tehdä jokaisen sivunlatauksen yhteydessä. Koska tietojen uudelleenkysely käyttäjältä olisi erittäin epäkäytännöllistä, on ongelmaan kehitetty muutamia ratkaisuja. Esimerkiksi evästeet yhdessä istuntojen kanssa tarjoavat verkkosivujen kehittäjille ratkaisun asiakkaan tilatietojen ylläpitämiseen asiakkaan pyyntöjen ja palvelimen vastausten välille. (Rantala 2005, s. 218; Wikipedia b.)

Evästeet (cookie) ovat palvelimen asiakkaan selaimelle lähettämiä merkkijonoja, jotka tallennetaan tiedostona asiakkaan koneelle. Niitä voidaan hyödyntää mm. selaajan tilatietojen ylläpitoon sivupyyntöjen välillä. Tilatiedoilla tarkoitetaan esimerkiksi verkkokaupan ostoskorin sisältöä tai edellä mainittuja kirjautumisvaiheessa syötettyjä kirjautumistietoja, jotka pidetään muistissa asiakkaan selatessa verkkokaupan tuotteita. (Rantala 2005, s. 218; Wikipedia b.)

Eväste koostuu aina vähintään nimestä ja sen arvosta. Vaihtoehtoisiksi parametreiksi evästeelle voidaan antaa verkkoalue, hakupolku, voimassaoloaika ja turvallisuusparametri. Näiden parametrien avulla eväste voidaan esimerkiksi laittaa toimimaan ainoastaan HTTPS-yhteyden kanssa tai asettaa eväste vanhenemaan vaikka 24 tunnin kuluttua sen luonnista. (Meloni 2005, s. 265; Rantala 2005, s. 218; Software Business Competence c.)

Istunto (session) on puolestaan palvelimella toimiva tekniikka, jonka avulla käyttäjien tilatietoja voidaan tallentaa muistiin palvelinkoneella. Tiedot yhdistetään käyttäjiin ns. istuntotunnuksen avulla, joka nykyään useimmiten tallennetaan asiakkaan koneella evästeeseen kirjautumisvaiheessa. Istunnot ovat siltä kannalta tietoturvasempi keino evästeisiin nähden, että tiedot tallennetaan palvelimen puolesta. Näin esimerkiksi yleisillä tietokoneilla, joissa on useita käyttäjiä, muut käyttäjät eivät voi löytää tiedostoja tutkimalla muiden tietokoneen käyttäjien tietoja, joita on mahdollisesti tallennettu evästeisiin. (Meloni 2005, s. 278 - 279; Software Business Competence c; Wikipedia b.)

Muutamia käytännön kohteita, joissa käytetään istuntoja ja evästeitä ovat mm. verkkokaupat. Niissä käyttäjä yleensä kirjautuu järjestelmään henkilökohtaisilla käyttäjätunnuksilla, jolloin palvelin luo istuntotunnuksen ja lähettää sen evästeenä käyttäjälle. Jatkossa kun asiakas laittaa ostoskoriin tavaroita, tallennetaan ostokset palvelimen istuntoon. Käyttäjä puolestaan lähettää aina palvelimelle sivunpyynnön yhteydessä oman istuntotunnuksen evästeestä. Näin palvelin voi tunnistaa eri käyttäjät, jotka käyttävät palveluita. Kun käyttäjä kirjautuu ulos kaupasta, tuhotaan eväste, johon oli tallennettu istuntotunnus. (Ohjelmointiputka; Wikipedia b.)

Vaikka evästeet yhdistettynä istuntoihin ovatkin tehokas ja kohtuullisen helppo ratkaisu toteuttaa, liittyy niiden käyttöön ongelmia. Suurin niistä on se, että niihin ei voida täysin luottaa, koska osa internetin käyttäjistä kytkee evästeet internetselaimesta pois päältä, jolloin nämä käyttäjät eivät voi käyttää evästeisiin perustuvaa sivua. Evästeet ovat kuitenkin nykyään kohtuullisen hyväksytty tapa autentikoinnin apuna, joten niiden käyttöä ei varsinaisesti voi suositella vältettävän. (Rantala 2005, s. 218.)

2.5 Dynaamisten sivujen hyödyt

Dynaamisten internetsivujen kaikkein suurin ja ilmeisin hyöty on varmasti se, että sivuston käyttäjät voivat itse luoda sisältöä internetiin ilman, että heiltä vaaditaan mitään teknistä tietotaitoa. Riittää, että osaa kirjoittaa ja selata internetiä.

Monet internetsivut olisivat lähes mahdottomia toteuttaa staattisilla sivuilla. Hyvänä esimerkkinä toimivat verkkokaupat. Ilman dynaamista toimintaa uusien tuotteiden lisäys, vanhojen muokkaus tai poisto olisi todella työläs prosessi sivuston omistajalle. Sivut pitäisi ensin paikallisesti muokata uudelleenlaisiksi ja sitten julkaista uudestaan palvelimelle käyttäen jotain

FTP-ohjelmaa. Myös tuotteiden tilausprosessi olisi asiakkaan kannalta kömpelö, sillä tuotteita ei voisi poimia ostoskoriin, eikä tilausta voisi mitenkään järkevästi seurata tai peruuttaa kuten nykyaikaisissa verkkokaupoissa. Ainut keino tilauksen tekemiseen täysin staattisesti rakennetussa verkkokaupassa olisi käyttää, joko sähköpostiin perustuvaa järjestelmää, puhelinta, tai perinteistä postia, jossa verkkokauppa toimisi paremminkin tuoteluettelona, jota vain luetaan.

2.6 Dynaamisten sivujen riskit

Dynaamisten internetsivujen riskit keskittyvät lähinnä itse dynaamisten sivujen kehittämissivuihin, sillä joskus voi olla vaikea määrittää käyttäjille rajoja, että mitä kaikkea käyttäjien annetaan tehdä. Jos sivut ovat erittäin dynaamiset, voivat käyttäjät tehdä jotain aivan odottamatonta, jota ei ole otettu huomioon sivuja suunniteltaessa. Tästä voi seurata tietoturvariskejä. Yleensäkin dynaamiset sivut vaativat enemmän kehittäjiltä kuin staattiset sivut, jotta niistä saadaan tietoturvallisia.

SQL-injektiot ovat suuri riski dynaamisia verkkosivuja kehitettäessä. SQL-injektio tarkoittaa sitä, että esimerkiksi verkkosivun lomakkeilla olevien kenttien kautta kehittäjien valmiiksi luomiin SQL-lauseisiin yritetään syöttää jotain järjestelmään tunkeutujan omia SQL-lauseen osia. Tunkeutuja voi yrittää selvittää tietokannan rakenteen ja sitä kautta vaikuttaa tietokannan sisältöön, esimerkiksi tuhoamalla tietokannan tiedot tai varastamalla jotain arkaluonteista tietoa. SQL-injektiot voidaan kuitenkin estää oikein rakennetuilla SQL-lauseilla, mutta varsinkin aloittelevat ohjelmoijat yleensä huomaavat tämän vasta sitten, kun vahinko on jo tapahtunut. (Heinisuo & Rauta 2007, s. 400; Mureahuha.)

Tuore esimerkki SQL-injektioiden hyödyntämisestä oli 13.10.2007 julkaistu lista, johon oli listattu lähes 80 000 käyttäjän käyttäjätunnukset ja salasanat eri internetsivustoille. Useimmat tunnukset olivat suomalaisten käyttäjien. Tällaiset tapaukset ovat huomattava tietoturvariski koko internetin tietoturvan kannalta, sillä moni käyttää samoja tunnuksia eri internetsivustoilla. (Poropudas 2007; Pynnönen, 2007.)

Kaikkeen asiakkaalta tulevaan syötteeseen pitäisi suhtautua kriittisesti, ja se tulisi aina tarkistaa, että se varmasti on odotetun tyyppistä tietoa. Esimerkiksi jos käyttäjän odotetaan syöttävän numero johonkin lomakkeen kenttään, tarkistetaan palvelimella, että asiakkaan lähettämä

tieto on numeerinen. SQL-injektioiden ehkäisemiseksi varsinkin lomakkeille asiakkaiden syöttämät tiedot tulisi ajaa aina jonkin suodatuksen läpi, ettei niihin voisi kirjoittaa SQL-lauseita. Myös tietokantojen salaaminen on suositeltavaa, sillä se vaikeuttaa tietojen käyttöä jos ne jotenkin päätyvät väärin käsiin. (Heinisuo & Rauta 2007, s. 400; Mureakuha.)

Riskeiksi voi määritellä myös sen, että jos käyttäjille annetaan kirjoitusoikeus johonkin järjestelmään, täytyy sivuston kehittäjän olla tarkkana, mitä heidän annetaan tehdä. Tämän vuoksi dynaamiset sivut vaativat joskus jopa jatkuvaa valvontaa ja ylläpitoa. Jos vaikka ollaan kehittämässä jotain keskustelupalstan tapaista järjestelmää, on valvottava, ettei sivuilla julkaista mitään asiattomuuksia.

2.7 Yhteisöt

Ohjelmoipa sitten millä ohjelmointikielellä tahansa, tulee ohjelmoidessa usein vastaan ongelmia tai puutteita omissa tiedoissa. Tähän auttaa monesti, jos voi kysyä joltain asiaan perehtyneeltä neuvoa. Mutta mistä tällaisia asiantuntijoita löytää? Vastaus kysymykseen ovat erilaiset yhteisöt, joita löytyy internetistä lukuisia.

Suosittuja kotimaisia ohjelmointiyhteisöjä ovat esimerkiksi Ohjelmointiputka ja Mureakuha, joista löytyy apua moneen pulmaan ja useimpiin eri ohjelmointikieliin. Alla mainituilta sivuilta löytyy myös lukuisia esimerkkiohjelmiä, joiden avulla voi itsenäisesti opetella tekemään dynaamisia verkkosivuja. Muita hyviä ulkomaalaisia sivustoja ovat Microsoft Developer Networkin (MSDN) ja PHP:n viralliset kotisivut.

Ohjelmointiputka <http://www.ohjelmointiputka.net/>

Mureakuha <http://mureakuha.com/>

MSDN <http://msdn2.microsoft.com/>

PHP <http://fi.php.net/>

3 YRITYKSEN SÄHKÖINEN ASIAKKUUDENHALLINTA

Asiakkuudenhallinnasta käytetään useita eri termejä, joista suosituin on varmasti CRM (Customer Relationship Management). Sillä tarkoitetaan asiakkuudenhallintaa, ja sitä tukevia tietojärjestelmiä. Tämä määrittely kuitenkin vaihtelee hieman kysyttäessä eri henkilöiltä, sillä sitä voidaan ajatella eri näkökulmista: teknisesti tai asiakaspainotteisesti. Voidaan kuitenkin sanoa, että se on ideologia, jossa yhdistyvät yritysکوhtainen liiketoiminta-ajattelu ja sitä tukeva tietojärjestelmä. (Fujitsu; Koskinen 2005, s. 34; Mäntyneva 2001, s. 7; Tieke a.)

Asiakkuudenhallinta käsittelee asiakasta ja asiakkaaseen liittyviä toimia tiiviinä vuorovaikutussuhteena, jonka avulla asiakassuhteet voidaan pitää hallinnassa koko asiakassuhteen ajan. (Douglas; Fujitsu; Koskinen 2005, s. 36.)

Yrityksillä on usein monia tietokantoja joilla pidetään yllä tietoa asiakkaista, kuten Excelin dokumentteja, Accessin tietokantoja ja paperimuotoisia muistioita. Ongelmaksi tällä tavalla toteutetussa asiakkuudenhallinnassa kuitenkin muodostuu tiedon hajanaisuus. Useissa eri lähteissä sijaitsevista ja jatkuvasti päivittyvistä tietokannoista on hankala tai jossain tapauksissa jopa mahdoton saada kohtuullisessa ajassa muodostettua kokonaiskuvaa asiakkaan tilanteesta päätöksenteon tueksi. (Ala-Mutka & Talvela 2004, s. 141; Fujitsu.)

Asiakkuudenhallinnan tavoitteena on, parantaa yrityksen kannattavuutta lisäämällä asiakkaiden lukumäärää, tuottoja, asiakastyytyväisyyttä ja löytämällä yrityksen avainasiakkaat. Tavoitteena on myös helpottaa johtoryhmän, myyjien, markkinoijien ja tukipalveluiden toimintaa, hyödyntäen yhteistä tietokantaa, jota kaikki voivat käyttää paikasta tai ajasta riippumatta. Koska kaikki käyttävät samoja tietoja, pysyvät ne samalla ajan tasalla. Sähköisellä palvelulla tieto voidaan kuitenkin palauttaa tietokannasta eri muodossa, ja vaikka rajoittaen joidenkin käyttäjäryhmien pääsyä haluttuihin tietoihin. Sähköisen tietokannan avulla asiakkaista voidaan myös luoda erilaisia raportteja ja ennusteita tarpeen mukaan. (Ala-Mutka & Talvela 2004, s. 142; Douglas; Fujitsu; Koskinen 2005, s. 36; Tieke b.)

Jotta asiakkuudenhallinta voisi toimia elektronisissa järjestelmissä, täytyy yrityksen myyntiprosessit voida pilkkoa selkeisiin vaiheisiin, joille on ennakkoon määritelty tavoitteet, milloin mikäkin vaihe on valmis. Ilman vaiheisiin jakamista myyntiprosesseja ei voida hyödyntää erilaisten mittareiden kanssa, joilla seurataan, kuinka hyvin asiakassuhteet etenevät. (Douglas.)

3.1 Myyntiprosessi

Myyntiprosessi, eli myyntiputki, jaetaan vaiheisiin. Prosessissa edetään vaiheittain alusta loppuun. Läpikäynnin nopeus voi vaihdella myytävän tuotteen mukaan ja yleensä myyntiprosessi etenee sitä hitaammin, mitä kalliimmasta tuotteesta on kyse. (Douglas; Wolfram.)

Vaiheisiin jaetulla myyntiprosessilla saavutetaan mm. myyntien parempi hallinta ja tarkempi ennustettavuus. Molemmat ovat yrityksen liiketoiminnan kannalta erittäin tärkeitä. (Douglas.)

Alla on kuvattu kuvitteellinen ohjelmistoyrityksen kuusivaiheinen myyntiprosessi, ja selitetty lyhyesti, mitä asioita missäkin vaiheessa voitaisiin tehdä. Vaiheiden nimet ja lukumäärät voivat vaihdella yrityksittäin ja toimialoittain, mutta myyntiprosessin vaiheiden sisältö on useimmiten enemmän tai vähemmän samankaltainen eri yrityksissä. (Douglas; Wolfram.)

Kartoitussoitto-vaiheessa soitellaan eri yrityksille ja kaikin tavoin yritetään kerätä mahdollisimman pitkä lista potentiaalisia asiakkaita. Samalla selvitetään potentiaalisten asiakkaiden nykytilanne kyselemällä esimerkiksi, mikä järjestelmä asiakkaalla on nykyään käytössä ja kuinka tyytyväisiä he ovat siihen. Tässä vaiheessa myös selvitetään potentiaalisten asiakkaiden yhteyshenkilöt joihin mahdollisesti ollaan jatkossa yhteydessä. (Douglas; Wolfram.)

Asiakkaaseen perehtyminen -vaiheessa hankitaan lisätietoa asiakkaista, kuten kasvuhistoria ja taloudellinen tilanne. Tietolähteinä voivat toimia mm. yritysten kotisivut ja erilaiset palvelut jotka tarjoavat yrityksistä tietoa, kuten Blue Book. (Blue Book; Douglas; Wolfram.)

Asiakkaan yhteyshenkilöön otetaan yhteyttä ja esitellään tuotetta. Tällöin pyritään selvittämään ostovaikuttajat ja asiakkaan liiketoimintahaasteet, joihin myytävä tuote voisi vaikuttaa positiivisesti. Jos asiakkaan yhteyshenkilö myöntää, että tuote voi tarjota hyötyä yritykselle ratkaisemalla vaikka jonkin ongelman ja hän on edelleen kiinnostunut siitä, edetään myyntiprosessissa seuraavaan vaiheeseen. (Douglas; Wolfram.)

Ostokriteerien selvittäminen -vaiheessa tavataan asiakkaan ostovaikuttajat. Selvitetään heidän tarpeet, ostokriteerit, myytävän tuotteen hinta ja riskit, sekä sovitaan, että miten edetään. Tarvitaanko referenssikäyntejä, demotilaisuuksia tai lisäkartoituksia. (Douglas; Wolfram.)

Kunhan kaikki asiakkaan ostovaikuttajat (päättäjä, tekninen vaikuttaja ja käyttäjät) on tavattu ja ostovaikuttajilla on näkemys ratkaisusta, voidaan sopia etenemistavoista. (Douglas; Wolfram.)

Ehdotus-vaiheessa käydään asiakkaan kanssa läpi, millaisia lisenssisopimusvaihtoehtoja myytävällä tuotteella on, ja mitä kaikkea niihin sisältyy (lisäarvotuotteet, kuten ylläpito, tuki tai ohjelmistopäivitykset) ja ehdotetaan asiakkaalle jotain sopimusta. (Douglas; Wolfram.)

Vaiheen lopuksi kerrataan asiakkaan saamat hyödyt ja ehdotetaan jatkoetenemistä. Jos asiakas hyväksyy kaiken tämän, niin siirrytään seuraavaan vaiheeseen. (Douglas; Wolfram.)

Neuvottelu-vaiheessa käydään läpi ratkaisut, hinnat, ehdot ja muut avoimet asiat, mitä on asiakkaan kanssa käyty läpi tähän mennessä. Jos asiakas hyväksyy kaiken, niin on päästy yhteisymmärrykseen sopimuksesta. Eli saatiin kaupat tehtyä ja voidaan siirtyä asiakassuhteen arviointivaiheeseen. (Douglas; Wolfram.)

Arviointi-vaiheessa mietitään, miksi sopimus saatiin aikaiseksi tai miksi sopimus menetettiin? (Douglas; Wolfram.)

3.2 Asiakkuudenhallintajärjestelmän kehittämisen lähtökohdat

Asiakkuudenhallintajärjestelmät voidaan jakaa viiteen erilaiseen lähestymistapaan ja niiden yhdistelmiin, jotka ovat:

1. pakettisovellus
2. parametroitava sovellus
3. konfiguroitava sovellus
4. mukautettava sovellus
5. ohjelmoitu sovellus. (Ala-Mutka & Talvela 2004, s. 143.)

Pakettisovellus sisältää valmiit parhaisiin käytäntöihin perustuvat prosessit ja tietosisällön, joten sitä ei mukauteta toiminnallisesti. Myös sen yhteensovittaminen muiden tietojärjestelmien kanssa voi olla heikkoa, varsinkin jos tietojärjestelmät ovat eri tuoteperhettä. Parametroitavassa sovelluksessa voidaan muuttaa tiedon sisältöä (kenttiä) ja konfiguroitavassa sovelluksessa voidaan kenttien lisäksi muuttaa myös prosesseja. Mukautettava sovellus on näistä kaikkein joustavin. Sillä kyetään tekemään parhaiten asiakaskohtaisia sovelluksia sisäänrakennetun sovelluskehittimen avulla. (Ala-Mutka & Talvela 2004, s. 144.)

Sovelluskehitin on ohjelma, jolla voidaan tehdä uusia käsitelmalleja ja prosesseja ilman teknistä räätälöintiä. Ohjelman mukauttaminen tehdään muuttamalla symboleja ja käsitteitä, loogisia yhteyksiä ja operaatioita ilman, että joudutaan varsinaisesti ohjelmoimaan. (Ala-Mutka & Talvela 2004, s. 144.)

Ohjelmoitu sovellus tarkoittaa ohjelmiston arkkitehtuurin suunnittelua ja ohjelmoimista asiakaskohtaisesti ohjelman elinkaaren alusta loppuun. Näin ohjelmat vastaavat parhaiten yrityksen tarpeita, koska ne tehdään juuri jotain tiettyä yritystä varten, jolloin ne voivat tarjota jopa kilpailuetuja verrattuna valmiisiin yleisesti myytäviin vaihtoehtoihin nähden tehokkaamman prosessin hallinnan muodossa. Toisaalta tämäntyyppisten järjestelmien ylläpito ja jatkokehitys on yleensä kallista ja aikaa vievää, joten se ei sovellu kaikille. (Ala-Mutka & Talvela 2004, s. 144.)

Yrityksien tuleekin arvioida tarkasti, sopiiko esimerkiksi valmis pakettisovellus asiakkuudenhallinnan tietotekniseksi ratkaisuksi, jos sovelluksen täytyy olla hyvin integroitavissa yrityksen vanhoihin järjestelmiin. Tässä tapauksessa erikseen yritykselle ohjelmoitu sovellus voi olla perusteltu vaihtoehto, vaikka se todennäköisesti maksakin enemmän. Toinen vaihtoehto olisi tutkia, että löytyykö samasta tuoteperheestä yhteensopivaa ohjelmaa, johon asiakkuudenhallintasovelluksen tulee olla integroitavissa. (Ala-Mutka & Talvela 2004, s. 143.)

Yhteenvetona voidaan sanoa, että eri lähestymistavat sopivat erilaisille yrityksille. Valintaan voivat vaikuttaa mm. yrityksen koko, liiketoiminnan monipuolisuus ja strategian ainutlaatuisuus. (Ala-Mutka & Talvela 2004, s. 147.)

Seuraavassa (Taulukko 2.) on kuvattu vielä lyhyet määritelmät ja mahdolliset käyttökohteet, johon eri lähestymistavat soveltuvat.

Taulukko 2. Tietotekniset lähestymistavat asiakashallintajärjestelmiin ja niiden käyttökohteet. (Ala-Mutka & Talvela 2004, s. 147.)

LÄHESTYMISTAPA	KUVAUS	KÄYTTÖKOHTEET
1. Pakettisovellus	Valmiit prosessit, sisältö (kentät) ja näytöt, joita on vaikea muuttaa.	Pienet yritykset (hinta merkitsevä tekijä), yhtenäinen asiakaskanta, kontaktien operatiivinen hallinta, parhaiden käytäntöjen mukaiset prosessit ja vähän palvelukanavia.
2. Parametroitava sovellus	Valmiit prosessit, joiden sisältöä voidaan muuttaa.	Suuret ja pienet yritykset, yhtenäinen asiakaskanta, parhaiden käytäntöjen mukaiset prosessit, standardit toimintamallit ja vain vähän erilaisia palvelukanavia.
3. Konfiguroitava sovellus	Prosesseja voidaan muuttaa ja samoin kenttien sisältöä.	Suuret yritykset, yhtenäinen tai joitakin eroja toimintamalleissa. Melko laaja palvelukanavastrategia.
4. Mukautettava sovellus	Prosesseja ja tietosisältöä voidaan muuttaa paljon.	Monimuotoinen liiketoiminta. Laaja kirjo tuoteperheitä ja palveluja. Erilaisia asiakasryhmiä (pienet ja isot organisaatiot sekä kuluttaja-asiakkaat). Laaja palvelukanavastrategia
5. Ohjelmoitu sovellus	Prosessit ja sisältö sekä näytöt tehdään asiakaskohtaisesti.	Erikoiskohteet, joissa toiminnan erityispiirteet vaativat tarpeeseen tehtyjä näyttöjä, toimintoja ja prosesseja.

4 MYNNIN PIPELINE -INTERNETPALVELU

Päättötyön käytännön osuudeksi valittiin Myynnin pipeline -internetpalvelun suunnittelu ja toteutus. Työn toimeksiantaja oli Arffman Consulting Oy ja yrityksen yhteyshenkilönä toimi toimitusjohtaja Kimmo Nikkanen.

Myynnin pipeline -internetpalvelu on erittäin merkittävä osa Arffman Consulting Oy:n uuden liiketoiminta-avausten sarjaa, sillä yritys toimii tällä hetkellä lähes yksinomaan koulutusliiketoiminnassa. Tietokoneohjelmistojen valmistaminen ei kuulu yrityksen liiketoimintaan, mutta yritys on erittäin vahva osaja konseptoinnissa ja volyymipohjaisessa massaräätälöinnissä. Tällaisella internetpalvelulla on runsaasti kysyntää yritysmaailmassa, joten sillä on potentiaalia menestyä, ja palvelu on myös hyvin ajankohtainen yrityksen nykytilanteessa.

Myynnin pipeline -internetpalvelu on täysin internetselaimessa toimiva sovellus, jonka on tarkoitus olla yrityksissä myynnin johtamisen tukena ja helpottaa asiakkuudenhallintaa, tarjoamalla yrityksille tarvittavat työkalut myyntitapausten kirjaamiseen, seuraamiseen ja raportointiin.

Työ sovittiin tehtäväksi vaiheittain vuoden 2007 kesän ja syksyn aikana. Välillä kuitenkin pidettiin pieniä taukoja, koska samaan aikaan oli työn alla muitakin projekteja kuin opinnäytetyö. Katsottiin myös parhaaksi, että työ tehdään opiskelijan kotona, koska ei ollut mitään pakottavaa syytä siihen, että työ pitäisi tehdä Arffman Consulting Oy:n toimitiloissa.

Työn tekemiseen vaadittiin kaikkia aikaisemmissa kappaleissa käsiteltyjä asioita, kuten dynaamisen sisällön hallintaa PHP-ohjelmointikielellä, MySQL-tietokantoja, käyttäjien autentikoinnin hallintaa ja perusteet yritysten asiakkuuksien hallinnan teoriasta. Työn tekeminen aloitettiin aivan alusta, eli ohjelmasta ei ollut mitään keskeneräistä versiota valmiina, jota olisi voinut jatkaa.

Palvelun nimi ”Myynnin pipeline -internetpalvelu” ei todennäköisesti tule olemaan lopullisen tuotteen nimi, mutta tätä nimitystä käytettiin toistaiseksi opinnäytetyön ajan.

4.1 Taustaa

Nykyään moni yritys, joka harjoittaa jollain tavalla tuotteiden tai palveluiden myyntiä, kärsii epätietoisuudesta omien myyjien asiakasprosesseista. Tämä johtuu siitä, koska myyjien luomista asiakasprosesseista ei välttämättä saa muuten välitietoja, kuin ottamalla yhteyttä myyjään ja kysymällä häneltä, missä vaiheessa hän on jonkin asiakkaan kanssa. Epätietoisuus on suuri ongelma, sillä jos yrityksissä ei tiedetä, milloin kauppoja saadaan aikaiseksi, ei tiedetä myöskään milloin yritykseen saadaan rahaa. Tämä tekee tulevaisuuden ennustamisesta hankalaa ja pitkän tähtäimen suunnitelmien teosta jopa täysin mahdotonta.

Tämän opinnäytetyön käytännön osuutena luotu internetpalvelu yrittää ratkaista tämän ongelman, luomalla linkin yrityksen johtoryhmän ja myyjien välille, hyödyntäen internetse-laimessa toimivaa internetsovellusta.

Elektronisen myynnin seurannan avulla yritykset voivat tarkistaa reaaliajassa myyjien asiakkaita juuri silloin, kun on tarvetta. Tämän ansiosta yrityksen myynnin seuranta helpottuu huomattavasti, sillä tietoja ei tarvitse erikseen tiedustella myyjiltä, joiden työpisteet voivat sijaita ympäri Suomea. Riittää, kun myyjät vain syöttävät asiakkaansa tiedot aina järjestelmään ja päivittävät niitä myyntiprosessin etenemisen myötä.

4.2 Suunnittelu

Myynnin pipeline -internetpalvelun tekeminen lähti liikkeelle siitä kysymyksestä, että pitäisikö siitä tehdä itsenäinen ohjelma, Microsoft Excelin dokumentti, internetsovellus tai jopa näiden kaikkien yhdistelmä? Lopulta asiasta kuitenkin päätettiin, että siitä tulee puhdas internetsovellus, koska se ei vaadi loppukäyttäjältä mitään muuta kuin internetyhteyden, internetse-laimen ja käyttäjätunnukset järjestelmään. Tämän ratkaisun katsottiin tavoittavan suurimman käyttäjäkunnan, ja kaiken lisäksi, se on kaikkein helpoin ratkaisu, sillä se ei vaadi loppukäyttäjiltä mitään ohjelmistojen asentamista.

Suunnitteluvaiheessa erityistä huomiota pidettiin sovelluksen toimivuuteen mahdollisimman monessa ympäristössä. Se ei saanut vaatia mitään ”ylimääräisiä” liitännäisiä ja sen tuli toimia yleisimmin käytetyissä internetse-laimissa, kuten Internet Explorer, Mozilla Firefox ja Opera.

Sovelluksen tuli myös olla erittäin helppokäyttöinen ja vaivaton käyttää, koska yritysten myyjillä voi olla hyvin vaihtelevan tasoisia taitoja tietotekniikassa. Silti sen täytyi olla myös monipuolisesti muokattavissa, jotta se soveltuisi useaan eri ympäristöön, sillä yrityksissä käytettävä myyntiprosessi ja myytävät tuotteet voivat vaihdella hyvinkin kovasti eri yritysten välillä.

Suunnitelma eteni erilaisten tapaamisten, puhelinpalavereiden ja lukuisien sähköpostien pohjalta piirustuksiin, joissa kuvattiin sovelluksen eri osiot, niiden käyttöliittymät ja niiltä vaadittavat toiminnallisuudet. Itse ohjelmakoodia ei suunniteltu etukäteen lainkaan, eli sen tekemiseen ohjelmoijalle annettiin täysi vapaus, kunhan lopputulos vain sisälsi suunnitellun toiminnallisuuden.

Varsinaisen suunnitteluvaiheen päätyttyä sovellus jaettiin kolmeksi erilliseksi internetsivuksi: hallinta, ylläpito ja myyjä. Kaikista kolmesta internetsivusta on tarkemmat esittelyt luvussa ”Ohjelmistoarkkitehtuuri”, jossa on kuvattu suunnittelupalaverissa päätetyt ominaisuudet käyttötapauskaavioiden ja käyttöliittymästä otettujen kuvankaappausten avulla.

Ohjelmointikieleksi valittiin PHP ja tietokannaksi MySQL. Valitsemisperusteena olivat aikaisemmat hyvät kokemukset molemmista järjestelmistä ja se, että Arffman Consulting Oy:n internetpalvelimella oli valmiiksi tuki PHP:lle ja MySQL:lle. Yksi lisäperuste oli myös se, että tämä yhdistelmä ei tuottanut mitään lisäkustannuksia, vaikkei se ratkaisevasti vaikuttanutkaan päätökseen.

4.3 Kohderyhmä

Koska kohderyhmän luonnollisesti täytyy olla mahdollisimman suuri suunniteltaessa tämän tyyppistä sovellusta, siitä yritettiin tehdä mahdollisimman dynaaminen, että se soveltuisi useimmille yrityksille, jotka harjoittavat myyntiä. Toistaiseksi sovellus kuitenkin tukee vain suomenkieltä, joten pääkohderyhmänä ovat suomalaiset yritykset.

Useimmat sovelluksen osa-alueet voidaan muokata vastaamaan yritysten tarpeita. Kaksi keskeisintä osiota, joilla sovellusta voidaan muokata eri yritykselle sopivaksi, ovat myyntiprosessin ja tuotevalikoiman luonti. Nämä siksi, koska molemmat ovat sellaisia, jotka varmasti poikkeavat jossain määrin eri yrityksissä.

4.4 Ohjelmistoarkkitehtuuri

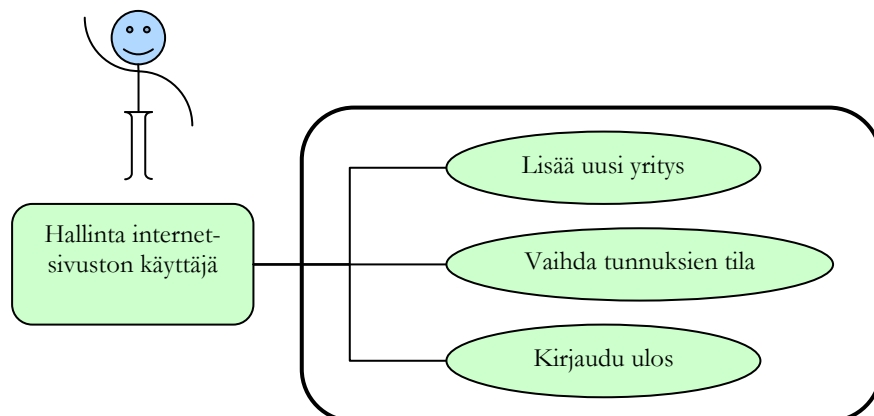
Koska Myynnin pipeline -internetpalvelun ohjelmistoarkkitehtuuria ei suunniteltu kovin tarkasti teknisestä näkökulmasta ennen ohjelmoinnin aloittamista, eteni projekti erilaisten prototyypin pohjalta nykyiseen muotoonsa, joka on kuvattu seuraavissa alaluvuissa.

4.4.1 Käyttötapauskaaviot

Seuraavassa on kuvattu koko sovelluksen toiminta kolmella käyttötapauskaaviolla: hallinta (Kuvio 8.), ylläpito (Kuvio 9.) ja myyjä (Kuvio 10.). Näissä käyttötapauskaavioissa ilmenevät kaikki Myynnin pipeline -internetpalveluun opinnäytetyön aikana tehdyt toiminnot. Kaikille kolmelle eri sivustoille vaaditaan erilliset käyttäjätunnukset. Näin ollen esimerkiksi yrityksen pääkäyttäjän tunnuksilla ei voida kirjautua saman yrityksen myyjien sivuille.

Hallinta

Hallinnan internetsivustolle (Kuvio 8.) kirjautumalla Arffman Consulting Oy:n henkilökunta voi lisätä järjestelmään uusia käyttäjätunnuksia asiakasyrityksille. Tunnuksia voidaan myös kytkeä päälle tai pois päältä, jos vaikka halutaan estää järjestelmän käyttö joltain asiakasyritykseltä.



Kuvio 8. Hallinta internetsivuston käyttötapauskaavio.

Ylläpito

Ylläpidon internetsivustolle (Kuvio 9.) kirjautunut yrityksen pääkäyttäjä voi lisätä oman yrityksen myyjille käyttäjätunnuksia, sekä muokata ja poistaa niitä. Myyjän tunnuksia muokkaamalla voidaan esimerkiksi asettaa jonkin myyjän tunnukset pois käytöstä, jolloin myyjä ei voi enää kirjautua sisään. Myös myyjän salasana voidaan käydä vaihtamassa muokkausvalikosta, jos myyjä vaikka unohtaa omat tunnuksensa.

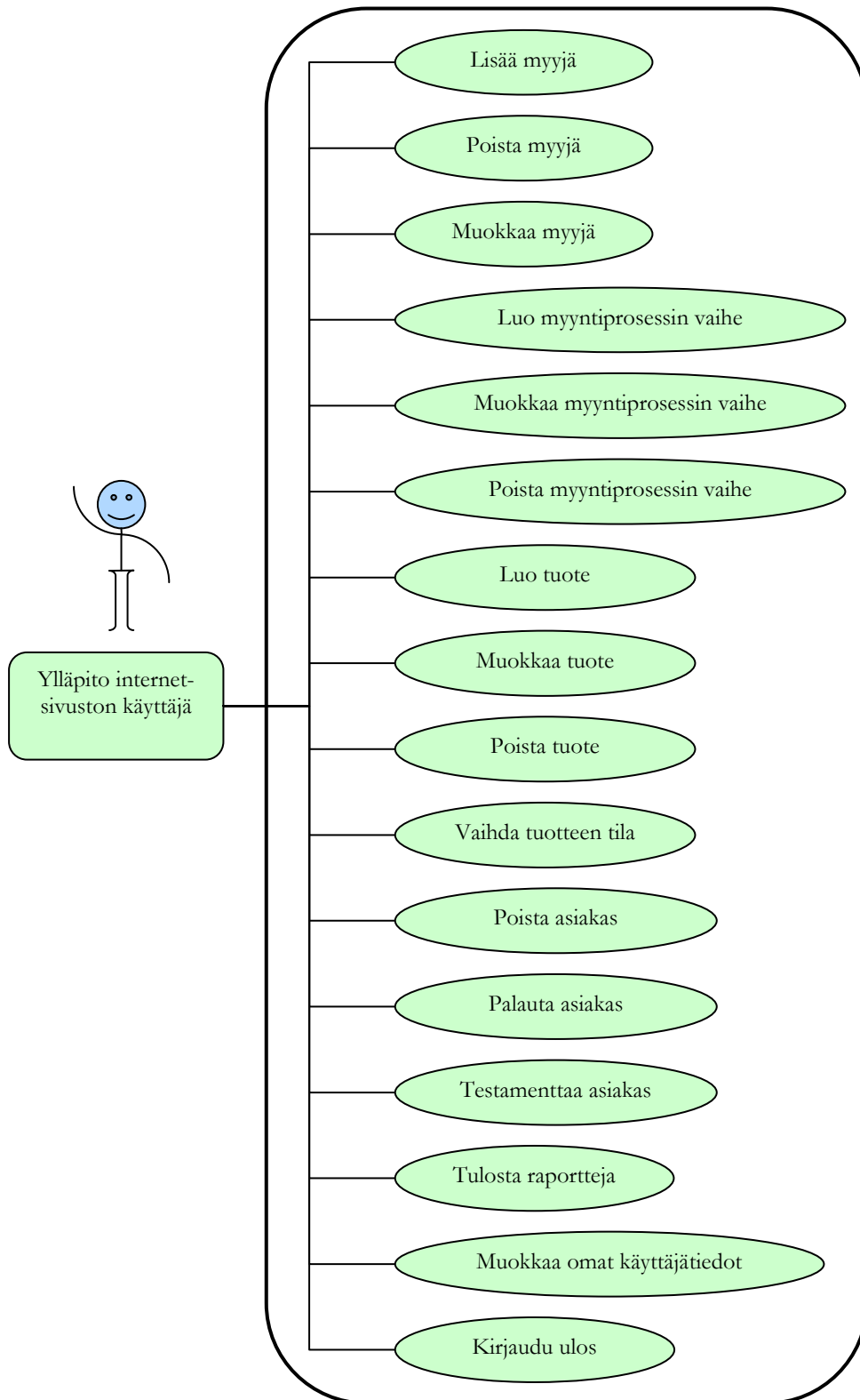
Ylläpidon internetsivujen kautta voidaan myös luoda yrityksen käyttämä myyntiprosessi. Myyntiprosessin luonnin jälkeen sen vaiheita voidaan vielä vapaasti muokata tai jopa poistaa niin kauan, kuin yritys ei ole vielä lisännyt asiakkaita. Ensimmäisen asiakkaan luonnin jälkeen myyntiprosessi lukittuu automaattisesti ja myyntiprosessin vaiheiden lisäys ja poisto kytetään pois päältä. Vastaavasti myyntiprosessin muokaus kytkeytyy takaisin päälle, jos kaikki asiakkaat poistetaan.

Tuotteita voidaan lisätä, muokata ja poistaa, aivan kuten myyntiprosessin vaiheitakin. Poikkeuksena on kuitenkin, että tuotteita voidaan lisätä ja poistaa myös silloin, kun järjestelmään on lisätty asiakkaita. Tuotteiden muokkaustoiminnolla tuotteiden nimiä voidaan korjata tai asettaa esimerkiksi jokin tuote pois käytöstä. Tällöin myyjät eivät voi enää valita niitä asiakkaalle.

Myyjien poistamia asiakkaita voidaan palauttaa takaisin niitä poistaneille myyjille. Asiakkaita voidaan myös siirtää myyjältä toiselle tai vaihtoehtoisesti poistaa kokonaan tietokannasta.

Yrityksen myyjien lisäämien asiakkaiden pohjalta voidaan myös luoda raportteja, joita ei opinnäytetyön aikana ennätetty tekemään kuin yksi, jossa listataan valituista yrityksen asiakkaista oleellisimpia tietoja.

Näiden toimintojen lisäksi ylläpitäjän tunnuksilla kirjautunut käyttäjä voi myös vaihtaa oman käyttäjätunnuksen salasanan.

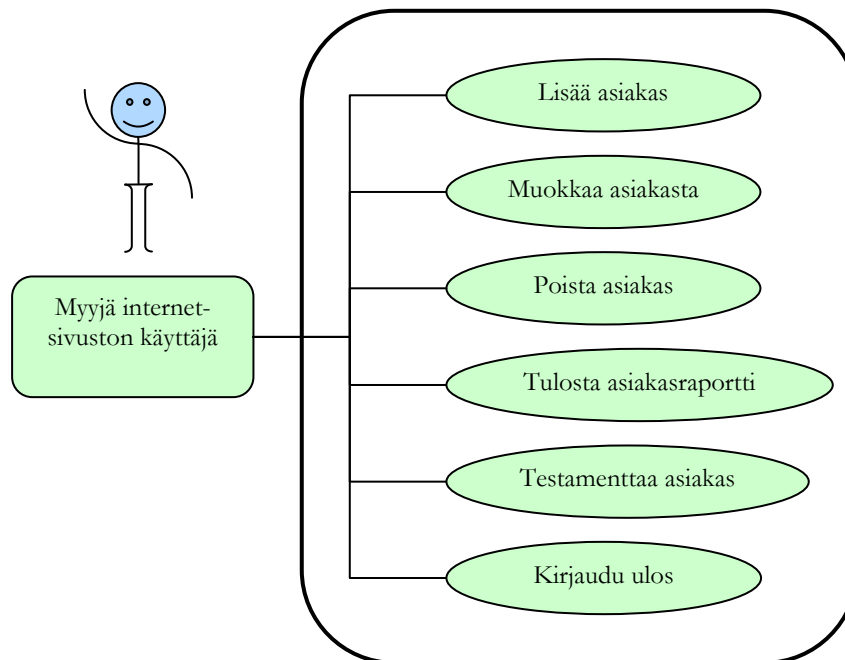


Kuvio 9. Ylläpito internet-sivuston käyttötapa-
skaavio.

Myyjä

Myyjän internetsivustolla (Kuvio 10.) voidaan lisätä, muokata ja poistaa asiakkaita. Myyjän poistaessa asiakasta, tietoja ei kuitenkaan oikeasti poisteta tietokannasta. Asiakkaan tietoihin vain määritetään, ettei sitä enää tulosteta myyjän näkymään. Lopullinen asiakkaiden tietokannasta poistaminen täytyy suorittaa aina yrityksen ylläpidon toimesta.

Asiakkaiden käsittelyn perustoimintojen lisäksi asiakkaista voidaan tulostaa tiivistelmäraportti, jossa näkyvät asiakkaista kaikkein oleellisimmat tiedot. Myyjät voivat myös testamentata, eli siirtää omia asiakkaita toisille yrityksen myyjille.

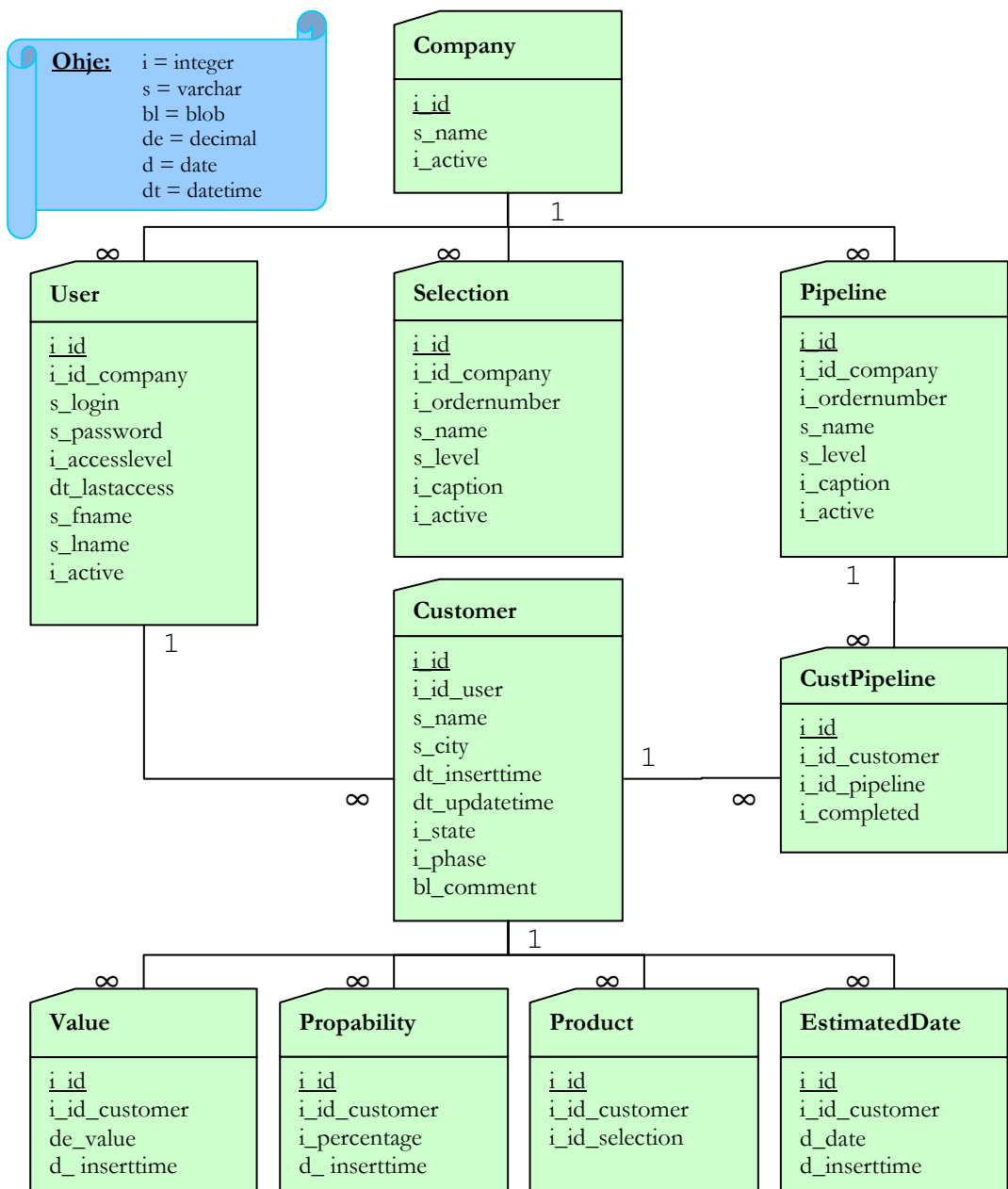


Kuvio 10. Myyjän internetsivuston käyttötapauskaavio.

4.4.2 Tietokanta

Tietokannan suunnitteluun käytettiin runsaasti aikaa, jotta sitä ei tarvitsisi enää suuresti muokata varsinaisessa ohjelmointivaiheessa. Yleensä tietokannan muutokset aiheuttavat monia muutoksia myös ohjelmakoodiin. Sovelluksessa käytetty tietokanta kehittyi muutaman prototyypin perusteella nykyiseen muotoonsa, mikä on kokonaisuudessaan kuvattu seuraavassa kuviossa (Kuvio 11.).

Tietokanta (Kuvio 11.) toimii kokonaisuudessaan seuraavalla tavalla: yhdellä yrityksellä voi olla useita myyjiä, tuotteita valikoimassa ja myyntiprosessin vaiheita. Yhdellä myyjällä puolestaan voi olla useita asiakkaita ja asiakkaalla voi olla useita kaupan arvoja, kaupan todennäköisyysprosentteja, kaupan ennustettuja päivämääriä sekä tuotteita. Kaupan arvolle, todennäköisyysprosentille ja kaupan ennustetulle päivämäärälle luotiin omat taulut, koska niistä täytyy kerätä myös historiatietoja. Historiatietoja tarvitaan siksi, että esimerkiksi myyjän muuntaessa kaupan todennäköisyysprosenttia, on oleellista tietää, oliko edellinen prosentti suurempi vai pienempi kuin uusi.

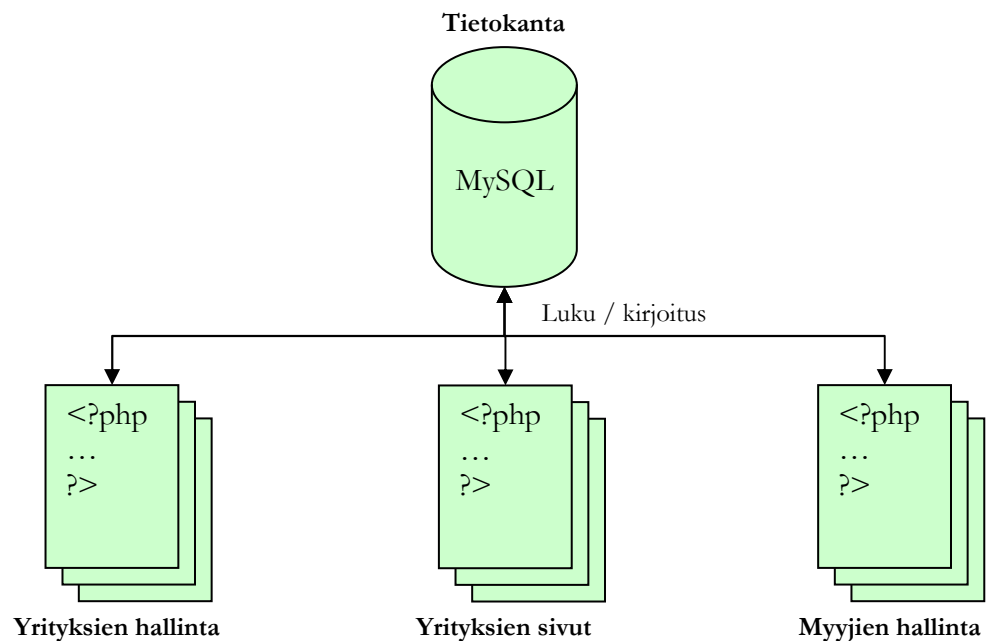


Kuvio 11. Myynnin pipeline -internetpalvelun tietokannan rakenne.

Tietokannan (Kuvio 11.) suunnittelussa suurimman haasteen muodosti dynaaminen myyntiprosessi. Myyntiprosessin tallentaminen kuitenkin saatiin ratkaistua siten, että tallennettaessa asiakasta tietokantaan, kopioidaan kaikista yrityksen myyntiprosessin vaiheista (Pipeline-tili) id:t CustPipeline-tiliin ja asetetaan niiden suoritettu-arvoksi ”ei-suoritettu”. Tämän lisäksi myyntiprosessin muokkaus lukitaan ensimmäisen asiakkaan luonnin jälkeen, jotta asiakasta kerätty tieto olisi yhdenmukaista.

4.4.3 Käyttöliittymä

Myynnin pipeline -internetpalvelu koostuu kolmesta (Kuvio 12.) erillisestä internetsivustosta: yritysten hallinta, yrityksen ylläpito ja yrityksen myyjien omat sivut, jotka ovat kaikki yhteydessä samaan MySQL-tietokantaan. Kaikkien osioiden käyttöliittymät pyrittiin tekemään yksinkertaisiksi, jotta palvelun käyttöönotto ei olisi ylivoimaisen vaikeaa.



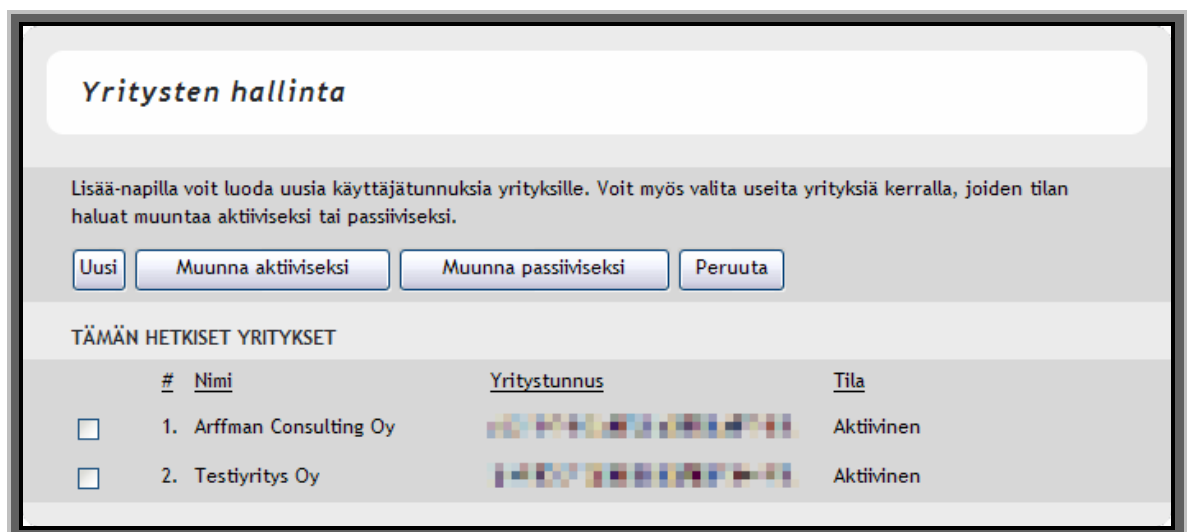
Kuvio 12: Myynnin pipeline internetsivujen kokonaiskuvaus.

Jos järjestelmään halutaan lisätä käyttöoikeus uudelle yritykselle, täytyy uusi yritys lisätä Arffman Consulting Oy:n toimesta yritysten hallintasivujen kautta. Siellä uudelle yritykselle luodaan yhden pääkäyttäjän tunnukset, jotka voidaan toimittaa asiakkaalle esimerkiksi sähköpostilla.

Tämän jälkeen asiakasyritys voi lisätä myyjien tunnuksia, jotka asiakasyritys puolestaan toimittaa myyjilleen. Samalla asiakasyritys voi lisätä omia tuotteita järjestelmään, joita myyjät voivat valita asiakkaille. Ennen uusien asiakkaiden lisäystä järjestelmään täytyy yrityksen pääkäyttäjän kuitenkin luoda omalle yritykselleen myyntiprosessi, jonka mukaan myyjien luomien asiakassuhteiden edistymistä voidaan mitata. Myyjien käyttämä asiakkaiden lisäystoiminto on lukossa siihen asti, kunnes myyntiprosessi on luotu.

Hallinta

Hallintasivut (Kuvio 13.) pitävät sisällään toiminnallisuuden uusien asiakasyrityksien lisäämiseen, poistoon, käyttöoikeuksien kytkemiseen ja pois kytkemiseen. Sivuille ei ole muilla käyttäjillä oikeuksia kuin Arffman Consulting Oy:n henkilökunnalla. Mikäli jonkin asiakasyrityksen tila vaihdetaan aktiivisesta ei-aktiiviseksi, ei millään asiakasyrityksen tunnuksilla voida enää kirjautua järjestelmään. Hallintasivuista on kuvattu täydellinen sivukartta liitteessä (Liite 1.).



Kuvio 13. Yrityksien hallinta-valikon käyttöliittymä.

Ylläpito

Ylläpitosivut on se kokonaisuus, jonka kautta asiakasyritykset voivat tutkia omien myyjiensä tekemisiä ja muokata palvelun toimintaa vastaamaan paremmin omia tarpeitaan. Ylläpitosivuista on kuvattu täydellinen sivukartta liitteessä (Liite 2.).

Ylläpitosivut tarjoavat toiminnallisuuden mm. myyjien tunnuksien lisäykseen, poistoon ja tunnuksien kytkemiseen pois päältä tai takaisin päälle. Näin myyjiä ei välttämättä tarvitse kokonaan poistaa, jos jonkin myyjän tunnukset vaikka haluttaisiin palauttaa toimintaan jonkin ajan kuluttua.

Ylläpitosivut osoittautuivat kokonaisuutena kaikkein vaativimmaksi osuudeksi toteuttaa, koska siellä asiakasyritys luo oman myyntiprosessin ja tuotevalikoiman. Lopulta kuitenkin löytyi tarpeeksi dynaaminen tietokannan rakenne ja käyttöliittymä, joiden avulla voidaan luoda melko monipuolisia myyntiprosesseja ja tuotevalikoimia.

Myyntiprosessi koostuu jostain määrästä vaiheita, joilla voi olla myös alivaiheita. Kuvassa (Kuvio 14.) esitetyllä Eteen-napilla voidaan liikkua valitun myyntiprosessin vaiheen alivaiheeseen ja vastaavasti Taakse-napilla päästään takaisin ylemmän tason vaiheeseen.

Myyntiprosessi (Lukossa)

Alla voit luoda yrityksen myyntiprosessin vaiheita. Myyntiprosessia voidaan muokata niin pitkään vapaasti, kuin yritykseen ei ole lisätty yhtään asiakasta. Jos yritykselle on lisätty asiakkaita, siirtyy myyntiprosessin luonti muokkaustilaan, jolloin uusia prosessin vaiheita ei voida enää lisätä. Vanhoja myyntiprosessin vaiheiden voidaan kuitenkin edelleen uudelleennimetä.

Juuritaso

Otsikkotaso

#	Nimike	Valinta
1	<input style="width: 90%;" type="text" value="Ensimmäinen päätaso"/>	<input type="checkbox"/>
2	<input style="width: 90%;" type="text" value="Toinen päätaso"/>	<input type="checkbox"/>
3	<input style="width: 90%;" type="text" value="Kolmas päätaso"/>	<input type="checkbox"/>

Kuvio 14. Myyntiprosessin luonti / muokkaus.

Myyntiprosessi laitetaan automaattisesti aina lukkoon sen jälkeen, kun asiakasyritykselle on lisätty vähintään yksi asiakas. Tämä täytyy tehdä siksi, että jos myyntiprosessia muunnettaisiin vielä asiakkaiden lisäyksen jälkeen, eivät asiakastiedot olisi enää yhtenäisiä, eikä niistä voisi enää luoda esimerkiksi raportteja. Myyntiprosessin vaiheiden nimiä voidaan kuitenkin edelleen uudelleen nimetä, vaikka tuoteprosessi olisi lukittu. Myyntiprosessi aukeaa automaattisesti muokattavaksi, jos kaikki asiakasyrityksen asiakkaat poistetaan.

Tuotteiden lisäys (Kuvio 15.) toteutettiin täysin samalla tekniikalla, kuten myyntiprosessin luominen. Eli asiakasyritykset voivat luoda tuotekategorioita, joiden alla on esimerkiksi alikategorioita, joista varsinaiset tuotteet löytyvät. Ainoana erona se, että tuotteiden muokkausta ei lukita missään vaiheessa, sillä tuotevalikoima voi muuttua usein asiakasyrityksissä.

Tuotteet

Juuritaso

Otsikkotaso

#	Nimike	Aktiivinen	Valinta
1	Koulutuspalvelut	Kyllä	<input type="checkbox"/>
2	Ohjelmistotuotteet	Kyllä	<input type="checkbox"/>

Lisää kenttiä Vähennä kenttiä

<< Taakse Poista valitut Muuta tila Poistu Eteen >>

Kuvio 15. Tuotevalikoiman luonti / muokkaus.

Ylläpitosivuilla voidaan myös selata asiakkaita (Kuvio 16.), esimerkiksi suodattaen niitä myyjäkohtaisesti tai asiakkaan tilan mukaan tai näyttäen vaikka kaikki samassa listassa. Samassa näkymässä asiakkaita voidaan testamentata myyjältä toiselle. Tätä ominaisuutta voisi käyttää esimerkiksi silloin, jos jonkin myyjän käyttäjätunnukset täytyy poistaa. Ylläpitosivuille tehtiin myös palautustoiminto, jolla myyjien poistamia asiakkaita voidaan palauttaa.

Asiakkaiden hallinta

Alla voit valita suodatuksen jos haluat esimerkiksi hakea tietyn myyjän poistamia asiakkaita. Huomaa myös, että jos poistat tästä valikosta asiakkaita, ne katoavat lopullisesti. Palauta-toiminto palauttaa myyjän poistaman asiakkaan ja asettaa sen tilaksi aktiivinen.

Suodatus

Myyjä: Asiakkaan tila:

Suodatetut asiakkaat

Arffman Consulting Oy, Kajaani
Yritys Oy, Kajaani (Poistettu)

Yrityksen myyjät

Kinnunen, Sakari
Leinonen, Petri

Kuvio 16. Asiakkaiden selaus ja hallinta.

Ylläpitopuolen tärkeimpiä ominaisuuksia ovat kuitenkin erilaiset raportit, joita voidaan tulostaa myyjien syöttämien asiakastietojen perusteella. Alla on kuvattu (Kuvio 17.) tiivistelmäraportti, johon on kerätty asiakasyrityksen myyjien lisäämien asiakkaiden perustiedot, kuten kauppasumma, kaupan todennäköisyysprosentti, todennäköinen eurosumma sekä myyntiprosessin vaihe.

Tiivistelmä asiakkaista

#	Asiakkaan nimi	Paikkakunta	Vastuuhenkilö	Tarjottu eur	Todnäk-%	Todnäk-eur	
1	Arffman Consulting Oy	Kajaani	Petri Leinonen	100 000,00 €	75	75 000,00 €	2/2 0/4 0/2

Kuvio 17. Yrityksen asiakasraportti.

Myyjä

Myyjien sivut ovat asiakasyrityksen myyjille tarkoitettu sivusto, jossa myyjät kirjaavat ylös asiakkaitaan ja niiden myyntiprosessien vaiheita. Myyjän lisätessä asiakasta, asiakkaasta kirjaataan ylös nimi, paikkakunta, myyntiprosessin vaihe, myytävät tuotteet, kauppasumma, kaupan todennäköisyysprosentti, kaupan ennustettu toteutumispäivämäärä ja asiakkaan tila. Myyjän sivuista on kuvattu täydellinen sivukartta liitteessä (Liite 3.).

Myyjät voivat myös tulostaa raportteja omista asiakkaistaan ja testamentata asiakkaita asiakasyrityksen toisille myyjille. Alla (Kuvio 18.) on kuva myyjän asiakasnäkymästä, jossa asiakkaita voidaan lisätä, muokata, poistaa ja tulostaa raporttiin.

Valitse asiakas

Suodatus

Tuotteet: Asiakkaan tila:

Lajitteluperuste: Lajitteluun: Nouseva Laskeva

SUODATETUT ASIAKKAAT

Arffman Consulting Oy, Kajaani

Kuvio 18. Myyjän asiakasnäkymä.

Seuraavassa on kuvattu (Kuvio 19.), kuinka myyjät päivittävät asiakkaiden myyntiprosessin vaihteita. Tämä näkymä luodaan dynaamisesti sen perusteella, millaiseksi se on määritetty ylläpitopuolella myyntiprosessin luontivalikossa.

Myyntiprosessin yläpuolella (Kuvio 19.) näkyy värikoodattu tiivistelmä, miten myyntiprosessi on edennyt. Tiivistelmän vaiheiden määrä riippuu siitä, kuinka monta päävaihetta myyntiprosessiin luotiin ylläpitopuolella (Kuvio 14.). Jos jonkin päävaiheen kaikki alitasot on jäänyt merkkäämatta suoritetuksi ja on siirrytty seuraavaan vaiheeseen, muuttuu vaihe punaiseksi. Jos kuitenkin vain osa alivaiheista on merkattu suoritetuksi, näkyy vaihe keltaisena. Jos kaikki päävaiheen alivaiheet on suoritettu, näkyy vaihe vihreänä.

MYyntIPROSESSI	
Vaiheen nimi:	Suoritettu?
<div style="display: flex; justify-content: center; gap: 10px;"> <div style="background-color: red; color: white; padding: 2px 5px;">0/2</div> <div style="background-color: yellow; color: black; padding: 2px 5px;">2/4</div> <div style="background-color: green; color: white; padding: 2px 5px;">2/2</div> </div>	
1. Ensimmäinen päätaso	
1.1. Ensimmäinen alitaso	<input type="checkbox"/>
1.2. Toinen alitaso	<input type="checkbox"/>
2. Toinen päätaso	
2.1. Ensimmäinen	<input checked="" type="checkbox"/>
2.2. Toinen alitaso	<input checked="" type="checkbox"/>
2.3. Kolmas alitaso	<input type="checkbox"/>
2.4. Neljäs alitaso	<input type="checkbox"/>
3. Kolmas päätaso	
3.1. Ensimmäinen alitaso	<input checked="" type="checkbox"/>
3.2. Kolmas alitaso	<input checked="" type="checkbox"/>

Kuvio 19. Asiakkaan myyntiprosessin päivitys.

Myyjän asiakkaalle valitsemien tuotteiden päivitys tapahtuu täysin samalla periaatteella kuin myyntiprosessin päivitys. Myytävien tuotteiden kohdalla vain merkitään halutut tuotteet aktiiviseksi. Esimerkiksi seuraavassa esitetyssä kuviossa (Kuvio 20.) asiakkaalle on valittu myytäväksi tuotteeksi Myynnin pipeline -ohjelmistotuote. Huomaa, että aikaisemmin esitellyssä kuviossa (Kuvio 15.) määritellyt tuotteiden päätasot näkyvät pelkkinä otsikoina, koska se taso oli määritelty otsikkotasoksi.

TUOTTEET	
Tuotteen nimi:	Valinta?
Koulutuspalvelut	
Hoiva	<input type="checkbox"/>
Luonnonvaratalous	<input type="checkbox"/>
Maahanmuuttaja	<input type="checkbox"/>
Ohjaava	<input type="checkbox"/>
Rakentaminen ja puu	<input type="checkbox"/>
Tietotekniikka	<input type="checkbox"/>
Yrittäjä ja talous	<input type="checkbox"/>
Ohjelmistotuotteet	
CV-Palvelu	<input type="checkbox"/>
Myynnin pipeline	<input checked="" type="checkbox"/>

Kuvio 20. Asiakkaalle myytävien tuotteiden valitseminen.

Kauppan tietojen päivitys tapahtuu yksinkertaisesti syöttämällä kauppasumma, kaupan toteutumisen todennäköisyysprosentti, kaupan todennäköinen toteutumispäivämäärä ja asiakkaan tila oheisessa kuvassa (Kuvio 21.) esiteltyihin kenttiin. Kenttien alapuolelle tulostetaan myös myyjän aikaisemmin syöttämät arvot, joiden avulla voidaan seurata, kuinka myyntiprosessi on edennyt. Näiden tietojen lisäksi myyjä voi kirjoittaa asiakkaan tietoihin jotain omia kommentteja, jos on tarpeen.

Kaikki käyttäjien syöttämät tiedot luonnollisesti ajetaan tarkistuksien läpi ja vääryyppisistä tiedoista myös ilmoitetaan virheilmoituksilla. Tietojen poistojen yhteydessä myös pyydetään aina vahvistus. Väärän tyyppisten tietojen lähetyksen ennaltaehkäisemiseksi myyjälle on myös kirjoitettu lyhyet ohjeet (Kuvio 21.) kenttien viereen.

KAUPAN TIEDOT	
Summa euroissa:	<input type="text"/> (esim. "55000") 2.10.2007 ---> 100 000,00e
Todennäköisyysprosentti:	<input type="text"/> (esim. "75" tai "100") 2.10.2007 ---> 75 %
Kauppa saadaan:	<input type="text"/> (esim. "15.1.2007" tai "1.10.2007") 2.10.2007 ---> 24.12.2007
Asiakkaan tila:	<input type="text" value="Aktiivinen"/>
KOMMENTIT	
<input type="text"/>	

Kuvio 21. Asiakkaan kaupan tietojen päivitys.

4.5 Tietoturva

Tietoturva on aina tärkeä osa minkä tahansa verkkosovelluksen kehitystä. Myynnin pipeline - internetpalvelussa tietoturvaa on ajateltu useammastakin näkökulmasta, koska etenkin julkisessa verkossa olevien verkkosovellusten tietoturvan kanssa täytyy olla erityisen huolellinen.

Myyjät esimerkiksi eivät voi poistaa lisäämiään asiakkaita kokonaan järjestelmästä. Poisto tapahtuu käytännössä niin, että jos myyjä poistaa asiakkaan, se häviää myyjän näkyvistä, mutta se voidaan edelleen palauttaa ylläpidon toimesta takaisin myyjälle tai vaikka siirtää se toiselle myyjälle. Tämän toiminnon tarkoitus on toimia palautustoimintona, jos myyjä poistaa vahingossa jotain asiakkaita tai jos myyjä vaikka ilkeästi poistaisi kaikki asiakkaansa, niin tiedot eivät katoa yritykseltä. Toki tietoja voidaan myös oikeasti poistaa tietokannasta, mutta se tapahtuu aina yrityksen ylläpidon toimesta.

Itse tietokannan tietoturva toteutettiin salaamalla käyttäjätiedot. Näin tietoja ei voi lukea selväkielisenä edes silloin, jos niitä luettaisiin suoraan tietokannasta. Tämän avulla voidaan ennaltaehkäistä käyttäjätietojen väärinkäyttöä, jos ne jostain syystä päätyisivät väärin käsiin.

Erityistä huomiota pidettiin SQL-injektioiden ennaltaehkäisemiseen. Tämä toteutettiin erilaisten suodatusten ja tarkistusten avulla, joiden avulla palvelun käyttäjien lähettämät tiedot tarkistetaan aina ennen kuin niitä käytetään tietokantahauissa.

Yleensäkin Arffman Consulting Oy:n palvelimille on annettu käyttöoikeudet vain niitä todella tarvitseville henkilöille, eikä niitä anneta aivan kenelle tahansa. Sovellusta myös tullaan käyttämään mitä suurimmalla todennäköisyydellä HTTPS-yhteyden kautta, kunhan sovellus otetaan oikeasti käyttöön.

4.6 Kehitystyökalut

Myynnin pipeline -internetpalvelu toteutettiin täysin PHP-ohjelmointikielellä, kehittämien ja testaten sitä PHP:n versiolla 5.2.1. Sivuilla ei käytetä edes JavaScriptiä, koska haluttiin poistaa mahdollisuus, että käyttäjän selaimessa voi olla JavaScriptit kytkettynä pois päältä.

Työssä käytettyjen kehitystyökalujen valintaan eniten vaikuttava tekijä oli aiempi kokemus. Sekä sivuston ohjelmoija ja työn toimeksiantaja olivat ennestään tuttuja PHP:llä ja MySQL:llä työskentelystä ja molemmille löytyi jo ennestään tuki Arffman Consulting Oy:n palvelimelta. Toinen kriteeri oli se, että tämä yhdistelmä on hyvin yhteensopiva.

Sivujen varsinainen ohjelmointi toteutettiin täysin Notepad++:lla, mikä on puhtaasti tekstipohjainen editori. Tätä editoria käytettiin lähinnä sen vuoksi Windowsin muistion tilalla, koska se helpottaa ohjelmointia automaattisella värikoodauksella.

Tietokantoja puolestaan hallittiin phpMyAdminilla, mikä on internetselaimen kautta käytettävä SQL-tietokannan hallintatyökalu, joka on myös täysin ilmainen.

4.7 Tulevaisuus

Myynnin pipeline -internetpalvelusta saatiin ensimmäinen toimiva versio valmiiksi, mutta se ei ole vielä tarpeeksi viimeistelty, että sitä kannattaisi vielä alkaa kauppaamaan laajasti asiakkaille. Tämä oli kuitenkin jo aloitusvaiheessa selvää, sillä kyseessä on melko laaja kokonaisuus. Siispä palvelun kehitystä jatketaan vielä uusien versioiden myötä opinnäytetyön ulkopuolisella ajalla.

Opinnäytetyön aikana tehdystä versiosta on helppo jatkaa uusien ominaisuuksien lisäyksiä, sillä palvelun ydintoiminnot ovat jo täysin toimivia. Ydintoiminnoiksi voidaan luokitella mm. hallintapuolella yrityksen lisäyksen, ylläpitopuolella myyntiprosessin, tuotteiden ja myyjien hallinnan sekä myyjien puolella asiakkaiden hallinnan.

Ensimmäiset uudet ominaisuudet, mitä sovellukseen todennäköisesti tullaan toteuttamaan, ovat monipuolisemmat raportointityökalut, parempi graafinen ilme ja yleensäkin käytettävyyden parantaminen muokkaamalla käyttöliittymää.

Erityisesti myyjien käytössä oleva asiakkaiden lisäys- ja muokkausnäkyä tullaan varmasti muokkaamaan, koska se ei nykyisellään sovellu hyvin yrityksille, joilla on esimerkiksi todella laaja tuotevalikoima. Sivun kasvaa epäkäytännöllisen pitkäksi, jos siihen listataan jopa satoja tuotteita allekkain. Lopullisen graafisen ulkoasun tulee todennäköisesti toteuttamaan siihen erikoistunut henkilö.

Uusien asiakasyrityksien lisääminen täytyy myös tehdä Arffman Consulting Oy:lle helpomaksi tulevaisuudessa. Nykyisellään uudelle asiakasyritykselle täytyy tehdä omat kirjautumis-sivut ylläpidon ja myyjien puolelle manuaalisesti, ja lähettää ne Arffman Consulting Oy:n internetpalvelimelle FTP-ohjelmalla. Tämä ei sinänsä ole suuri toimenpide, mutta jos järjestelmä toimisi täysin ilman FTP-ohjelman käyttämistä, se nostaisi käytettävyyttä huomattavasti.

Seuraava merkittävä virstanpylväs olisi saada pilottiasiakas, eli jokin oikea yritys, joka ottaisi järjestelmän testikäyttöön. Tämän pohjalta saataisiin arvokasta palautetta kehitystarpeista ja varmasti tulisi myös sellaista palautetta, mitä ei ole tullut edes mieleen kun ohjelmaa on kehitetty. Tässä vaiheessa ohjelmaan voidaan vielä melko helposti lisätä uusia ominaisuuksia tai tehdä muutoksia vanhoihin, koska se ei ole laajasti käytössä.

5 POHDINTA

Teoriaosuudessa käsitelty dynaaminen toiminnallisuus verkkosivuilla on nykyään jo kovin arkipäiväinen asia. Tulevaisuudessa kuitenkin eri verkkosivut tullaan todennäköisesti tekemään ohjelmointikielellä, kuten PHP:llä tai ASP.NET tekniikalla staattisten HTML-dokumenttien sijaan, koska verkossa olevan tiedon hallinta voidaan näin toteuttaa joustavammin, eikä esimerkiksi dynaamisen sisällönhallinnan toteuttaminen ole enää kovin kallista. Toki kustannukset riippuvat aina toteutettavan järjestelmän koosta ja monimutkaisuudesta, mutta esimerkiksi perinteiset yrityksen kotisivut, joilla yrityksen toimintaa esitellään satunnaisille surffailijoille, eivät ole kovin kalliit toteuttaa. Internetistä löytyy jopa valmiita sivupohjia tosi halvalla, joihin yrityksen tarvitsee vain itse lisätä tekstit ja yrityksen logot, mitkä haluaa julkaista muiden nähtäville.

Teoriaosuus eteni kokonaisuudessaan hyvin ja melko tasaiseen tahtiin, vaikka siihen tuli yllättäviä muutoksia ja jopa kokonaan uusi pääotsikko verrattuna opinnäytetyön suunnitelmiin. Tämä johtui suurelta osin siitä, että opinnäytetyön suunnitelma tehtiin jo alkukesällä ja itse ohjelman suunnitelmat tehtiin loppuun vasta kesälomien alkaessa. Uusi pääotsikko, mikä löytyi, on nimeltään asiakkuudenhallinta. Se osoittautui lopulta todella laajaksi käsitteeksi, joten sen vuoksi sitä ei käsitelty teoriaosuudessa kovin tarkasti. Ainoastaan opinnäytetyön käytännön osuuden kannalta oleelliset asiat käsiteltiin. Teoriaosuudesta jätettiin myös JavaScriptien käsittely lähes kokonaan pois, koska sitä ei lopulta tarvittu käytännön työn tekemiseen ja sen opetteluun olisi näin kulunut melko paljon ylimääräistä aikaa. Toisaalta opinnäytetyössä käsiteltiin hieman myös ASP.NET:ä, mitä ei oltu suunniteltu käsiteltävän.

Käytännön osuus eteni myös tasaiseen tahtiin ilman mitään suurempia ongelmia. Tietenkin ohjelmoidessa tuli aina välillä virheitä. Niitä saattoi miettiä useammankin päivän, että mistä ne johtuivat, mutta kaikki virheet kuitenkin ratkesivat lopulta ilman suurempia ongelmia. Käytännön osuudessa suurin haaste oli sovelluksen dynaamisuus. Tämä siksi, koska ennakkoon ei voida esimerkiksi tietää, millaisia myyntiprosesseja missäkin yrityksessä on käytössä. Eli myyntiprosessi ja tuotevalikoima eivät voineet olla vain kiinteitä sivuja, joiden oletetaan käyvän suunnilleen kaikille, vaan niistä täytyi tehdä muokattavia. Lopulta kuitenkin myyntiprosessin ja tuotteiden editoreista tuli erilaisten prototyyppien seurauksena tarpeeksi monipuolisia.

Käytännön osuuden kulkua parhaiten vastaava elinkaarimalli on evolutiivinen protoilu, koska siinä edettiin vaiheittain aina sen mukaan, mitä parannettavaa tuli mieleen edellisestä toimivasta versiosta. Kyseinen elinkaarimalli on myös toiminut aikaisemmissa Arffman Consulting Oy:lle toteutetuissa ohjelmistoprojekteissa hyvin, joten sen käyttöä päätettiin jatkaa myös tässä projektissa.

Lähimpänä oleva tekninen lähestymistapa, jolla käytännön osuus toteutettiin, on parametroidava sovellus. Etenkin myyntiprosessi ja tuotevalikoima ovat ns. ”parametroitavissa”, koska yrityksen täytyy itse luoda oma myyntiprosessi ja tuotevalikoima, ennen Myynnin pipeline - internetpalvelun käyttöönottoa.

Kaikkineen käytännön osuuteen voi olla tyytyväinen. Siitä ei jouduttu karsimaan mitään muita opinnäytetyösuunnitelmassa kuvattuja ominaisuuksia, kuin eri kielivaihtoehdot. Lisäksi opinnäytetyössä valmistuneeseen versioon ei vielä tehty kovin monipuolisia raportteja. Molemmat kuitenkin voidaan lisätä myöhemmin opinnäytetyön ulkopuolisella ajalla. Erilaisia raportteja esimerkiksi on helppo lisätä tarpeen mukaan, koska se on vain tiedon tulostamista tietokannasta.

Käytännön osuuden tekemiseen kulunutta aikaa ei voi enää jälkikäteen tarkkaan laskea, koska sitä tehtiin pitäen välillä pieniä taukoja. Voisi arvioida siihen kuluneen yhteensä lähes kahden kuukauden kokopäiväinen työmäärä. Teoriaosuus valmistui ohjelmoinnin sivussa kirjoitellen. Loppua kohti teoriaosuuden kirjoittaminen kuitenkin nousi suuremmalle osuudelle päivän työmäärässä.

Kokonaisuutena opinnäytetyö oli haastava ja opettava prosessi. Työ eteni hyvään tahtiin ja toimeksiantajan sekä ohjaavan opettajan kanssa yhteistyö toimi ongelmitta, vaikka työn tekeminen suoritettiin opiskelijan kotona, eikä esimerkiksi toimeksiantajan tiloissa. Opinnäytetyöhön löytyi myös todella hyvin materiaalia internetistä ja Kajaanin eri kirjastoista, kunhan vain oikeat hakusanat löytyivät. Ongelmaksi muodostuivat lähinnä oikeiden, käytännön työn kannalta oleellisten asioiden poimiminen, ettei työ paisunut liian laajaksi kokonaisuudeksi.

LÄHTEET

KIRJALLISUUS

Ala-Mutka, Jukka & Talvela Erkki 2004. Tee asiakassuhteista tuottavia. Helsinki: Talentum.

Heinisuo, Rami & Rauta, Ilkka 2007. PHP ja MySQL. Helsinki: Talentum.

Inkinen, Ville 2003. ASP.NET Toolkit. Jyväskylä: Docendo.

Meloni, Julie 2004. PHP 5 fast & easy web development. Boston: Thomson Course Technology.

Mäntyneva, Mikko 2001. Asiakkuudenhallinta. Helsinki: WSOY.

Rantala, Ari 2005. Web-ohjelmointi. Jyväskylä: Docendo.

INTERNET

2K mediat a. Johdatus PHP-ohjelmointiin.

<http://www.2kmediat.com/php/johdanto.asp> (Luettu 5.10.2007).

2K mediat b. Miksi PHP?

<http://www.2kmediat.com/php/johdanto2.asp> (Luettu 5.10.2007).

Arfcon. Yritysesittely.

<http://www.arfcon.fi/> (Luettu 29.10.2007).

Blue Book. Blue Book.

<http://www.bluebook.fi/fi/bluebook/> (Luettu 30.9.2007).

Brilliance Web Design 2003. Better Web Design.

http://www.brillianceweb.com/betterwebdesign/tips_52.aspx (Luettu 7.10.2007).

Business Intelligence Lowdown 2007. Top 10 Largest Databases in the World.

http://www.businessintelligencelowdown.com/2007/02/top_10_largest_.html (Luettu 7.10.2007).

Douglas, Brad. Manage your sales pipeline effectively.

<http://office.microsoft.com/en-us/help/HA011424061033.aspx> (Luettu 30.9.2007).

Eaton, Jess 2007. Why not ASP.Net?

<http://www.lullabot.com/blog/why-not-asp-net> (Luettu 7.10.2007).

Fujitsu. Asiakkuuden hallinta.

<http://www.fujitsu.com/fi/services/businessapps/crm/> (Luettu 9.10.2007).

Guthrie, Scott 2007. VS 2008 and .NET 3.5 Beta2 Released.

<http://weblogs.asp.net/scottgu/archive/2007/07/26/vs-2008-and-net-3-5-beta-2-released.aspx> (Luettu 23.10.2007).

Hillyer, Mike. Migration from Microsoft SQL Server and Access to MySQL.

<http://dev.mysql.com/tech-resources/articles/migrating-from-microsoft.html> (Luettu 22.9.2007).

Hull, Sean. PHP vs. ASP.NET Redux.

http://www.oracle.com/technology/pub/columns/hull_php2.html (Luettu 16.9.2007).

Kiukas, Juha. Tietotyypit.

http://www.frodos.fi/kurssi/sql/tab_tietyy.htm (Luettu 23.9.2007).

Korpela, Jukka 2006. Dynaaminen HTML.

<http://www.cs.tut.fi/~jkorpela/webjulk/3.6.html> (Luettu 19.5.2007).

Koskinen, Katja 2005. Asiakkuus ja sen hallinta CRM-ohjelmiston avulla Case mySAP. Tampereen yliopisto. Johtamistieteiden laitos. Pro gradu -tutkielma.

<http://tutkielmat.uta.fi/pdf/gradu00739.pdf> (Luettu 19.10.2007).

Kurkinen, Ville 1998. Java Servletit.

http://www.tml.tkk.fi/Studies/Tik-110.300/1998/Essays/java_servletit.html (Luettu 22.9.2007).

Lahtonen, Tommi 2004. Relaatiotietokannat.

<http://appro.mit.jyu.fi/doc/tiedonhallinta/tietokannat/index1.html> (Luettu 12.10.2007).

Lee, Wei-Meng 2005 a. How ASP.NET works.

<http://www.ondotnet.com/pub/a/dotnet/2005/09/19/what-is-asp-net.html> (Luettu 5.10.2007).

Lee, Wei-Meng 2005 b. Visual Studio and ASP.NET.

<http://www.ondotnet.com/pub/a/dotnet/2005/09/19/what-is-asp-net.html?page=2> (Luettu 5.10.2007).

Louhi. Louhi Kotisivupalvelu.

<http://www.louhi.fi/?path=louhiwebdesigner> (Luettu 6.10.2007).

msdn. Microsoft Visual Studio Express Edition.

<http://msdn2.microsoft.com/en-us/express/default.aspx> (Luettu 6.10.2007).

Mureakuha. SQL-injektio.

<http://wiki.mureakuha.com/wiki/SQL-injektio> (Luettu 17.10.2007).

MVnet. PHP-ohjelmointi.

http://www.mvnet.fi/index.php?osio=Kotisivun_teko&sivu=PHP-ohjelmointi (Luettu 22.9.2007).

Notepad++. About Notepad++.

<http://notepad-plus.sourceforge.net/> (Luettu 22.9.2007).

Ohjelmointiputka. Osa 7 – Evästeet ja istunnot.

<http://www.ohjelmointiputka.net/opas.php?tunnus=phpj7> (Luettu 15.10.2007).

php-editors. Reviewed PHP Editors.

<http://www.php-editors.com/review/> (Luettu 6.10.2007).

PHP: Hypertext Preprocessor a. Chapter 31. Magic Quotes.

<http://fi2.php.net/manual/en/security.magicquotes.php#security.magicquotes.what> (Luettu 2.10.2007).

PHP: Hypertext Preprocessor b. PHP 5.2.4 Released.

<http://www.php.net/> (Luettu 2.10.2007).

Poropudas, Timo 2007. Poliisi tavoitti 80 000 salasanan levityksestä epäillyn.

http://www.itviikko.fi/page.php?page_id=46&news_id=200725995 (Luettu 22.10.2007).

Pynnönen, Jouko 2007. Sherlockin päiväkirja.

<http://www.omasana.fi/jouko-pynnonen/110330> (Luettu 22.10.2007.)

Software Business Competence a. Web-sovellusten ohjelmointi: Web-sovellusten toteutus-tekniikoita.

<http://www.oamk.fi/sbc/www/websovellus.php> (Luettu 12.10.2007).

Software Business Competence b. Web-sovellusten ohjelmointi: Tiedostojen käsittely.

<http://www.oamk.fi/sbc/www/tiedosto.php> (Luettu 22.9.2007).

Software Business Competence c. Web-sovellusten ohjelmointi: Käyttäjän tunnistaminen ja seuranta web-sovelluksissa.

<http://www.oamk.fi/sbc/www/tunnistaminenseuranta.php> (Luettu 15.10.2007).

Tieke a. Asiakkuuden hallinta.

http://www.tieke.fi/verkkokaveri/teemat/asiakkuuden_hallinta/ (Luettu 28.9.2007).

Tieke b. Usein kysytyt kysymykset.

http://www.tieke.fi/verkkokaveri/teemat/asiakkuuden_hallinta/usein_kysytyt_kysymykset/ (Luettu 17.10.2007).

Trachtenberg, Adam 2004. Why PHP 5 Rocks!

<http://www.onlamp.com/pub/a/php/2004/07/15/UpgradePHP5.html> (Luettu 19.9.2007).

Van der Meer, Jeroen. Taking a look at PHP 6.

<http://jero.net/articles/php6> (Luettu 2.10.2007).

W3Schools. ASP.NET Introduction.

http://www.w3schools.com/aspnet/aspnet_intro.asp (Luettu 6.10.2007).

Web.Blazonry 2002. PHP vs. ASP vs. JSP Part 1.

<http://blazonry.com/devnotes/phpasp1.php> (Luettu 16.9.2007).

Webopedia. Integrated Development Environment.

http://www.webopedia.com/TERM/i/integrated_development_environment.htm (Luettu 6.10.2007).

Wikipedia a. Tietokanta.

<http://fi.wikipedia.org/wiki/Tietokanta> (Luettu 16.9.2007).

Wikipedia b. Eväste.

<http://fi.wikipedia.org/wiki/Evaste> (Luettu 10.12.2007).

Wolfram, Eric. Sales Lead Tracing — Your Pipeline is the key to selling your pants off.

http://wolfram.org/writing/howto/sell/sales_pipeline.html (Luettu 29.9.2007).

Yank, Kevin 2002. Event Driven ASP.NET Development with C#.

<http://www.sitepoint.com/article/driven-asp-net-development-c> (Luettu 24.9.2007).

LIITTEET

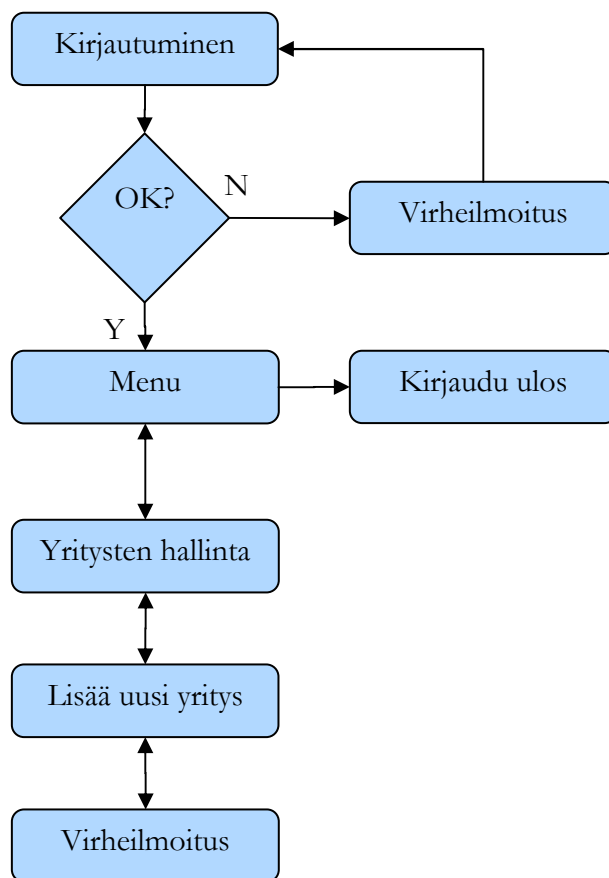
Myynnin pipelineen käyttöliittymäkartat

Liite 1: Hallinnan käyttöliittymä

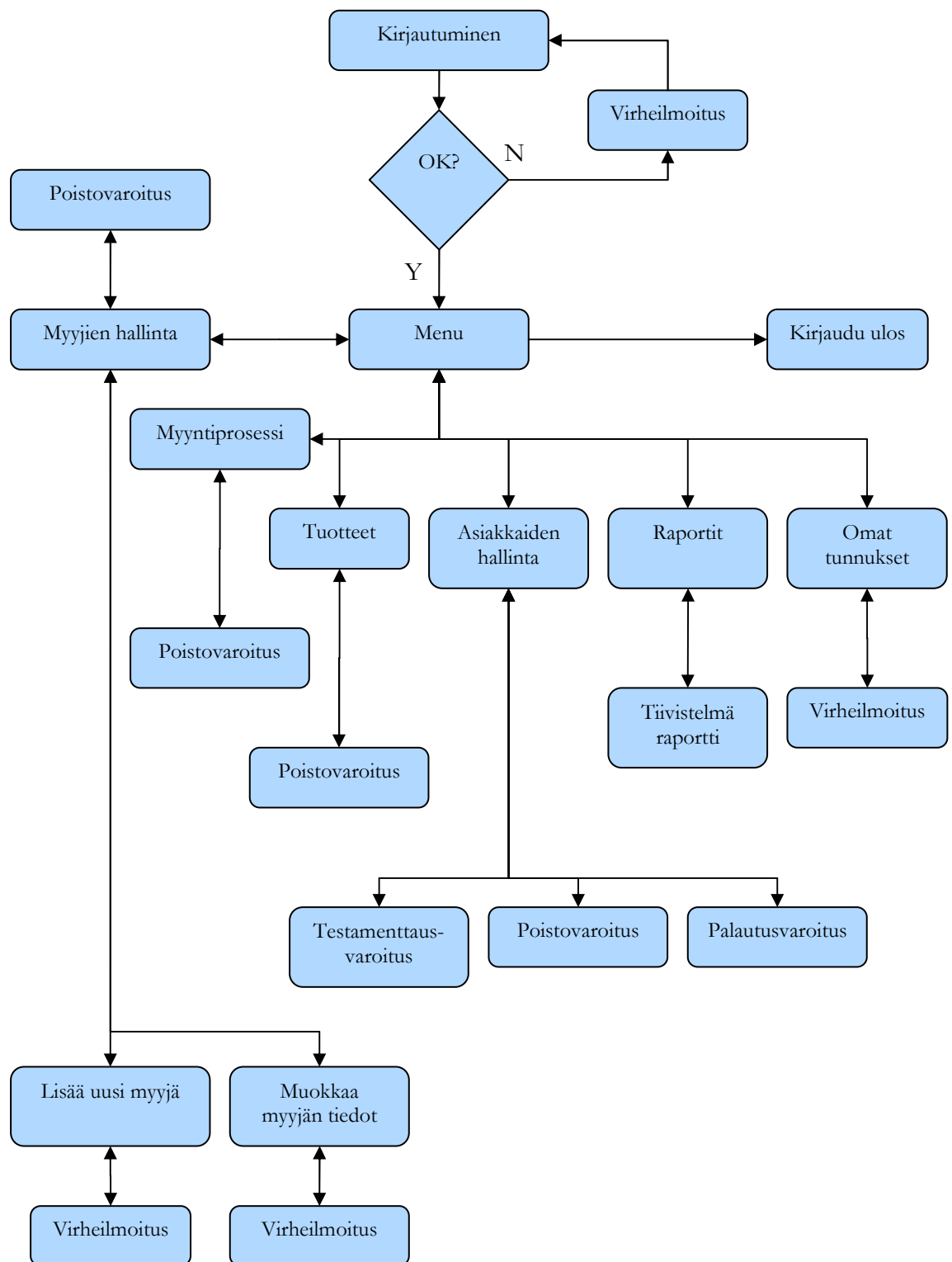
Liite 2: Ylläpidon käyttöliittymä

Liite 3: Myyjän käyttöliittymä

Hallinnan käyttöliittymä



Ylläpidon käyttöliittymä



Myyjän käyttöliittymä

