



Toimistopalvelualan yrityksen tietojärjestelmä

Lauri Pellikka

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2014



Tekijä(t) Lauri Pellikka	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Toimistopalvelualan yrityksen tietojärjestelmä	Sivu- ja liitesivumäärä 18 + 6
Opinnäytetyön otsikko englanniksi Information management system for an office service company	
<p>Opinnäytetyön päämäärä on luoda sovellus toimistopalvelualan yrityksen käyttöön. Sovelluksella on tarkoitus hallinnoida yrityksen tilauksia ja asiakastietoja. Sovellus perustuu erään suomalaisen yrityksen tarpeisiin. Yrityksen nykytilanteessa tilauksia hallinnoidaan paperilomakkeilla. Projektin tarkoitus on luoda vaihtoehtoinen järjestelmä, jossa tilauslomakkeet ovat alusta alkaen sähköisessä muodossa. Tämä voi teoriassa nopeuttaa tilausten käsittelyä.</p> <p>Sovelluksen tarkoitus on saada luettua, lisättyä, muokattua ja poistettua yrityksen asiakasorganisaatioita, organisaatioiden henkilöstöä, yrityksen myymiä tarvikkeita ja palveluita, sekä vastaavasti hallinnoida näistä tehtäviä tilauksia. Projekti toteutetaan siten, että sovellus toimii internet-sovelluksena Debian-palvelimella. Toiminnallisuus ohjelmoidaan PHP-kielillä. Tietojen tallentamiseen käytetään MySQL-tietokantaa.</p> <p>Projekti alkoi maaliskuussa 2014 ja päättyi syyskuussa 2014.</p> <p>Valmistuneella sovelluksella on mahdollista tietysin edellytyksin ylläpitää yrityksen asiakasrekisteriä, palveluja sekä näistä tehtäviä tilauksia. Sovelluksen käyttöönotto yrityksessä ei kuulu opinnäytetyön laajuuteen.</p>	
Asiasanat PHP, SQL, asiakashallinta	

Author Lauri Pellikka	
Degree Programme Business Information Technology	
The title of thesis Information management system for an office service company	Number of pages and appendices 19 + 6
<p>This project thesis aimed at creating an application for the needs of an office service company. The application was supposed to manage the company's customer orders and customer records. The application was based on the needs of a Finnish company. In its current state, the company uses paper forms to manage customer orders. The purpose of the thesis project was to create an alternative system in which the order forms are in electronic form from the beginning. In theory, this could speed up the order processing.</p> <p>The meaning of the application was to read, add, modify and delete the company's customer organizations, the personnel of the organizations, the products and services which the company sells, and manage the orders made by the customers. The project application runs as an Internet application using a Debian server. The functionality is programmed with PHP scripting language. Application data will be stored in a MySQL relational database.</p> <p>The thesis work started in March 2014 and ended in September 2014. The finished application can, with certain prerequisites, manage the company's customer register, items and services, and orders made for these products.</p> <p>Implementation of the application in the company was out of project scope.</p>	
Key words PHP, SQL, customer management	

Sisällys

1	Johdanto	1
1.1	Yritystausta	1
1.2	Opinnäytetyön tarkoitus	1
1.3	Vaatimusmäärittely	2
1.4	Opinnäytetyön rajausta	2
1.5	Käytetyt tekniikat.....	2
2	Teoriataustaa	3
2.1	HTML.....	3
2.2	PHP	3
2.3	CSV	4
2.4	Relaatiotietokanta	4
2.4.1	SQL ja MySQL	5
2.4.2	Load data infile.....	5
2.5	Debian	6
3	Sovellustoteutus.....	8
3.1	Tietokannan suunnittelu ja toteutus.....	8
3.2	Kehykset ja sovellus	9
3.2.1	Sopimusasiakkaat.....	9
3.2.2	Henkilöstö	9
3.2.3	Palvelut.....	10
3.2.4	Tuotteet.....	10
3.2.5	Työtilaukset.....	12
4	Pohdinta.....	14
4.1	Tulokset	14
4.2	Kehitysehdotukset ja eettiset kysymykset	14
4.2.1	Tarviketilaukset ja työtilaukset.....	14
4.2.2	Työtilaukset, pienet asiat.....	15
4.2.3	Hinnoittelu	15
4.2.4	Tarvikkeet ja palvelut	15
4.2.5	Pois rajatut aiheet	16
4.2.6	Lomakkeet ja relaatiösäännöt.....	16
4.2.7	Graafinen ilme.....	16
4.2.8	Tietoturvallisuus	17
4.3	Opinnäytetyöprosessin ja oman oppimisen arviointi.....	17
5	Lähteet.....	18
6	Liitteet	19
6.1	Liite 1. Tietokannan taulujen luontilauseet.....	19

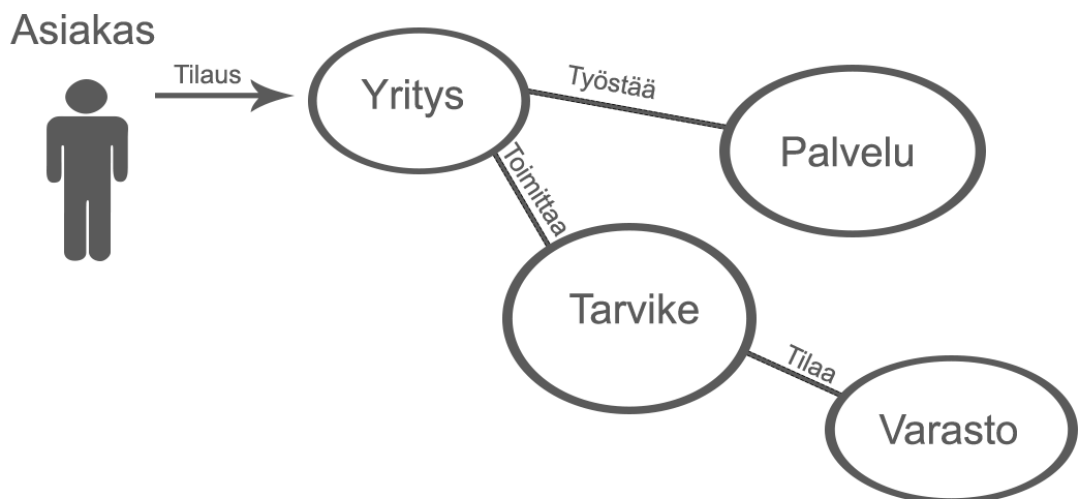
6.2	Asiakasorganisaatioiden tulostaminen listana	22
6.3	Asiakashenkilön tietojen muokkaus lomakkeella, ensimmäinen sivu.....	23
6.4	Asiakashenkilön tietojen muokkaus lomakkeella, toinen sivu	24

1 Johdanto

Opinnäytetyön tarkoituksena on luoda tietohallintajärjestelmä toimistopalvelualan yrityksen tarpeisiin. Sovelluksen tarkoitus on hallinnoida yrityksen asiakasrekisteriä, tuote- ja palvelurekisteriä, sekä luoda ja muokata asiakkaiden tilauksia.

1.1 Yritystausta

Opinnäytetyö perustuu suomalaisen toimistopalvelualan yrityksen tarpeisiin. Yritys myy toimistopalveluja, esimerkiksi kopiointipalveluja, julistetulostusta, toimistotyötä ja postituk-
sia. Lisäksi yritys myy toimistotarvikkeita kattavasta käsivarastosta, sekä tuhansia muita nimekkeitä 1-3 arkipäivän toimitusajalla. Yrityksen asiakasrekisteri koostuu sopimusasiak-
kaista eli asiakasorganisaatioista, joilla on laskutussopimus yrityksen kanssa.



Kuvio 1. Yrityksen nykytilanne.

Yrityksen nykytilanteessa ruohonjuuritason työtehtäviä hallinnoidaan paperilla. Työntekijä täyttää jokaisesta tehdystä tilauksesta paperisen tilauslomakkeen johon merkitään esimerkiksi asiakkaan nimi, yritys tai yhdistys jota hän edustaa, sekä tilauksen tiedot: tuotekoodi, nimeke, määrä ja hinta jokaista tilattua tarviketta tai palvelua kohden. Näitä tietoja käytetään myöhemmin laskutuksen perusteena. Laskutukseen käytetään erillistä toimin-
nanohjausjärjestelmää.

1.2 Opinnäytetyön tarkoitus

Opinnäytetyön tarkoitus on esittää yksi tapa korvata se työosuus, joka nykyään yrityksellä hallinnoidaan pääosin paperilla. Paperilomakkeen täyttämisen sijaan työntekijä täyttää sovelluksen lomakkeen Internetissä. Lomakkeelle ei tarvitsisi kirjoittaa tuotteen tuotekoodin lisäksi tuotteen nimeä tai hintaa, sillä sovellus löytäisi nämä tiedot itsestään. Tuote-

koodi ja määrä olisivat siis riittäviä tietoja. Tämän kaltainen sähköinen lomake voisi säästää yrityksen aikaa ja vaivaa.

1.3 Vaatimusmäärittely

Valmiin sovelluksen on tarkoitus mahdollistaa yrityksen asiakasorganisaatioiden lisäämisen, muokkaamisen ja poistamisen. Tämä sisältää laskutus-, katu- ja toimitusosoitetiedot sekä muut yhteystiedot. Jokaiselle organisaation alaisuuteen on tarkoitus lisätä henkilöasiakkaita, jotka tekevät työtilauksia. Henkilötietoja on voitava muokata, ja henkilöt on voitava poistaa tarvittaessa.

Sovelluksella tulee voida ylläpitää listaa yrityksen tarjoamista palveluista, sekä erillistä listaa yrityksen myymistä tuotteista. Tuotteita on useita tuhansia. Lisäksi sovelluksen tulee voida lisätä, ylläpitää ja muokata asiakkaiden tekemiä tilauksia.

1.4 Opinnäytetyön rajaus

Opinnäytetyön ei ole tarkoitus korvata yrityksen nykyisin käytössä olevia menetelmiä, eikä sen ole tarkoitus vastata toiminnallisuudeltaan kaikkia niitä menetelmiä, jotka yrityksellä nykytilanteessa on käytössään. Työn tarkoitus on arvioida vaihtoehtoisen toimintatavan mahdollisesti antamia hyötyjä.

1.5 Käytetyt tekniikat

Sovellus toteutetaan Debian-palvelimella siten, että toiminnallisuus ohjelmoidaan PHP:llä ja HTML:llä, ja tietokantana toimii MySQL.

2 Teoriataustaa

Sovellus ohjelmoidaan PHP-kielen versiolla 5.3.3. PHP valittiin ohjelmointikieleksi sen oletetun yksinkertaisuuden ansiosta. Allekirjoittanut ei ennen projektia tuntenut PHP:tä. PHP:llä on kuitenkin helposti opittavan kielen maine ja etenkin yksinkertaisiin WWW-sovelluksiin PHP:tä pidetään varmana valintana. Tämän takia PHP oli luontevin valinta.

Tietokanta toteutetaan MySQL-tietokannan versiolla 5.1.49. MySQL valikoitui projektin tietokannaksi sen ilmaisuuden sekä ennen kaikkea laajan Internet-tuen ansiosta. Vaikka kaikilla muilla suurilla tietokantajärjestelmillä on mittavat viralliset dokumentaatiot, käytännön ongelmatilanteessa Internet on osoittanut tarjoavansa eniten hakutuloksia MySQL:lle.

Toteutuksen käyttöjärjestelmänä toimii Debianin versio 6.0.6. Debian oli helppo valinta, sillä se tarjottiin projekti käyttöön ilmaiseksi ja esiasennettuna.

2.1 HTML

HTML (HyperText Markup Language) on Internet-sivujen ohjelmointiin yleisesti käytetty merkintäkieli. HTML:n tyypillinen käyttötarkoitus on dokumentin rakenteen kuvaaminen. HTML:n kehityksen aloitti Tim Berners-Lee vuonna 1989. Internet, vaikkakin nykyistä suppeampi, oli tuolloin jo olemassa. Berners-Leen ajatuksena oli dokumenttien yhdistäminen keskenään. Sen sijaan että dokumentti olisi vain ladattavissa palvelimelta, se voisi sisältää linkin toiseen dokumenttiin, sekä toisen dokumentin osa olisi nähtävissä jo ladatussa dokumentissa. Hypertext-nimitys juontaa juurensa tekstin linkkikäyttöön.

Seuraavien vuosien aikana Berners-Lee keskusteli HTML:n ideasta muiden kehittäjien kanssa. 1993 Useampi taho alkoi kehittää omaa selaintaan Berners-Leen ideaan perustuen. Tästä kehittyi World Wide Web (WWW), jonka ansiosta Internet kasvoi nopeasti alusta alkaen ja monet suuryritykset huomasivat Internetin tärkeyden. Suurimmat kehitystyön tekijät, esimerkiksi IBM, Microsoft, Netscape ja World Wide Web Consortium, yhdistivät voimansa helmikuussa 1996 luodakseen HTML:lle standardin, joka valmistui tammikuussa 1997 HTML:n version 3.2 myötä. (Raggett, D., Lam, J., Alexander, I., Kmiec, M. 1998.)

2.2 PHP

PHP (Hypertext Preprocessor, alun perin Personal Home Page) on ohjelmointikieli, joka sai alkunsa Rasmus Lerdorfin vuonna 1994 Perl- ja C-ohjelmointikielillä kirjoittamana ohjelmakoodina. Lerdorf kirjoitti koodia alun perin kotisivuilleen, mutta koodin kasvattaessa suosiotaan ja monipuolistuessa muutkin alkoivat työstää sitä. PHP kehittyi vähitellen omaksi ohjelmointikielekseen, ja sen kehitystiimi kasvoi. (The PHP Group 2014.)

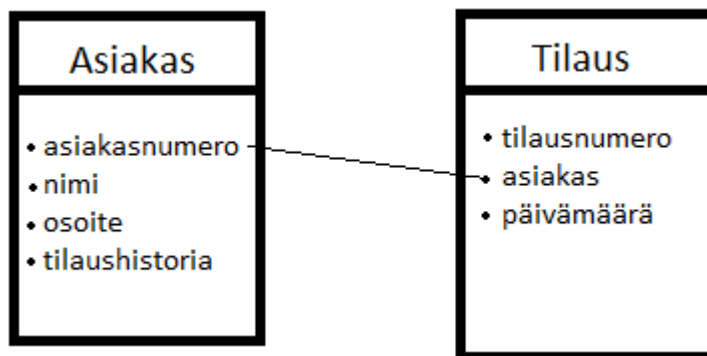
PHP eroaa esimerkiksi JavaScriptistä sillä, että ohjelman koodi upotetaan ajettavalle WWW-sivulle ja palvelin ajaa koodin sen sijaan, että käyttäjän pääte ajaisi sen. Käyttäjä saa pelkän lopputuloksen ajetusta koodista. (The PHP Group 2014.)

2.3 CSV

CSV (Comma-Separated Values) on tiedostomuoto, jolla voidaan tallentaa tekstiä taulukon muodossa. Tyypillisesti taulukkolaskentaohjelmassa taulukon kentät jaetaan riveihin ja sarakkeisiin, kun taas CSV-tiedosto erottaa kentät toisistaan määritellyillä merkeillä. Esimerkiksi pilkku tekstisyötteiden välissä voi määritellä sarakkeen vaihtumisen, ja puolipiste rivin vaihtumisen. (Internet Society 2005.)

2.4 Relaatietietokanta

Relaatiomallinen tietokanta on yleinen menetelmä tallentaa tietoa sähköiseen muotoon. Tieto tallennetaan tauluihin. Rivien arvoille voidaan määrätä loogisia yhteyksiä eli suhteita muihin tauluihin. Näin yhdestä taulusta voidaan viitata toiseen tauluun. (Connolly & Begg 2010, 91.)



Kuvio 2. Esimerkki relaatiomallista.

Kuvio 2 havainnollistaa relaatiomallia. Yksi taulu käsittää listan asiakkaista, jotka yksilöidään asiakasnumerolla. Toinen taulu käsittää listan asiakkaiden tekemistä tilauksista, jotka yksilöidään tilausnumeroilla. Asiakas määritetään tilauksen tietoihin samalla numerolla jolla tämä on yksilöity Asiakas-taulussa. Taulujen asiakasnumerollisten sarakkeiden välille on määritelty yhteys, jolloin nämä muodostavat suhteen toisiinsa. (Connolly & Begg 2010, 94-95.)

Relaatietietokannan ideaa määritteli julkisesti ensimmäisenä F.E. Codd 1970 työskennellessään IBM:llä. Coddin mukaan tiedon varastoinnilla oli kolme päätavoitetta:

- Pitää tieto itsenäisenä, ettei se olisi riippuvainen käytettävästä sovelluksesta.

- Helpottaa tiedon rakennetta, yhtenäisyyttä ja poistaa turhaa tietoa.
- Tehdä mahdolliseksi sellaiset kyselykielet, jotka keskittyvät tiedon muokkaamiseen samanaikaisesti useissa kentissä.

IBM kehitti 1970-luvulla ensimmäiset relaatiotietokannat. Näiden käyttökelpoisuus johti SQL-kyselykielen kehitykseen, sekä siihen että muut kehittäjät IBM:n lisäksi alkoivat kehittää omaa relaatiotietokantaansa. (Connolly & Begg 2010, 92-93.)

2.4.1 SQL ja MySQL

SQL (Structured Query Language) on standardisoitu ja laajasti käytetty kieli jolla tehdään kyselyjä relaatiotietokantaan. SQL on yleisesti jaettu kahteen osaan: DDL:ään (Data Definition Language) joka hallinnoi tietokannan rakennetta ja tiedonsaantia, ja DML:ään (Data Manipulation Language) jota käytetään tiedon hakemiseen ja päivittämiseen. SQL standardisoitiin aluksi 1987, mutta asiantuntijat pitivät kyseistä määritelmää alusta alkaen epäkelpona. Standardia on sittemmin päivitetty, ja nykyään SQL:ää itseään pidetään epävirallisesti tietokantahallintajärjestelmien standardikielenä. Suurin osa maailman merkittävistä tietokannoista on SQL-pohjaisia. (Connolly & Begg 2010, 134-137.)

MySQL on alun perin ruotsalaisen MySQL AB:n kehittämä SQL-pohjainen tietokannan hallintajärjestelmä. Sen ensimmäinen versio ilmestyi 1995. MySQL luotiin alun perin toisen tiedonhallintajärjestelmän lukemiseen ja muokkaukseen, mutta nykyään MySQL tulkitsee useita SQL-järjestelmiä. Tunnetuimmat näistä ovat InnoDB ja MyISAM.

MySQL on vapaan lähdekoodin ohjelmisto, eli se on ladattavissa ja asennettavissa ilmaiseksi. Nykyään MySQL:n omistaa Oracle Corporation. MySQL on yksi asennetuimmista ja käytetyimmistä tietokantahallintajärjestelmistä tällä hetkellä. (MySQL 2014.)

2.4.2 Load data infile

Load data infile on MySQL:n sisäänrakennettu toiminto, jolla voidaan siirtää suuria tekstimääriä esimerkiksi CSV-tiedostosta relaatiomuotoon. CSV on yleisesti käytetty tiedostomuoto tapa joka auttaa MySQL-toimintoa erottamaan rivit ja sarakkeet toisistaan.

Komento alkaa määrittelemällä, mistä tiedostosta käsiteltävä sisältö otetaan. Tiedoston sijainnin mukaan lähteeksi ilmoitetaan joko palvelimen isäntäkone tai sijainti ulkoisessa lähteessä. Seuraavaksi määritetään mihin tauluun tiedot luetaan, kuinka tietueet ja rivit erotetaan toisistaan ja kuinka monta riviä tiedoston alusta mahdollisesti jätetään lukematta. Lisäksi komennossa on lueteltava relaatiotaulun kohdesarakkeet siinä järjestyksessä kun ne tiedostossa on erotettu. (MySQL 2014.)

```

load data local infile 'tuotelista.csv'
into table item
fields terminated by ',';
lines terminated by '\n';
ignore 1 lines
(i_code, i_name, i_description, i_unit, @price, i_product_category, i_commission_account)
set i_price = replace (@price, ',', '.');
;

```

Kuva 1. Esimerkki load data infile –komennon käytöstä.

Kuva 1 havainnollistaa komennon käyttöä. Ensimmäinen riviin mukaan tiedosto tuotelista.csv ladataan paikalliselta koneelta, oletushakemistosta. Toinen rivi määrittää kohteeksi taulun item. Kolmas rivi määrittää puolipisteen merkiksi, jolla kentät erotetaan toisistaan. Neljäs rivi määrittää suurempi kuin –merkin erottavan rivit toisistaan. Viides rivi määrittää, että tiedoston ensimmäinen eroteltu, yleensä otsikkotietoja sisältävä rivi jätetään lukematta ja lisäämättä tietokantaan. Seuraava rivi luettelee item-tilin rivit, joihin csv-tiedoston rivien sarakkeet lisätään. Luetellun järjestyksen tulee vastata csv-tiedoston järjestystä.

Seitsemäs rivi määrittää, että item-tilin kenttään i_price luetaan muuten csv-tiedoston asianmukaisen järjestyksen vastaavaa tietueetta vastaavat tiedot, mutta kyseisen tietueen jokainen pilkku korvataan pisteellä. Tähän on syynä MySQL:n tapa käyttää desimaalipistettä. (MySQL 2014.)

2.5 Debian

Debian on vapaata lähdekoodia hyödyntävä tietokoneille suunnattu Linux-käyttöjärjestelmän jakeluversio. Debianin kehityksen aloitti Ian Murdock 1993, ja järjestelmän ensimmäinen versio ilmestyi 1996.

Jakelu-termi viittaa siihen, että Debian muodostuu Linux-ydinkäyttöjärjestelmästä ja Debianin kehittäjien valikoimasta ohjelmistokokonaisuudesta ja ylläpitojärjestelmästä. Debiania voi käyttää graafisen käyttöliittymän kanssa, mutta myös pelkkänä komentoliittymänä. Debianiin sisältyy pakettinhallintajärjestelmä, jolla käyttäjä voi valita sadoista tuhansista ilmaisista ohjelmistoista eli paketeista haluamansa. Käyttäjä voi muokata Debian-jakelua mielensä mukaan ja jakaa sitä eteenpäin.

Debianin uudet versiot ilmestyvät sykleissä. Uuden version ensimmäinen julkaisu on epävakaa. Debianin kehitystiimi ja vapaaehtoiset käyttäjät testaavat julkaisua ja korjaavat sen vikoja. Kun testausta on tapahtunut tarpeeksi, julkaisun tila vaihtuu virallisesti testausvaiheeseen. Tämä tarkoittaa sitä, että vaikka testausta on tapahtunut, julkaisuversio ei ole vielä valmis vakaaksi julkaisuksi. Vakaa julkaisuversio on Debianin kehitystiimin ja vapaaehtoisten käyttäjien perusteellisesti testaama. Vakaa julkaisu ei tyypillisesti kaadu, ja sopii

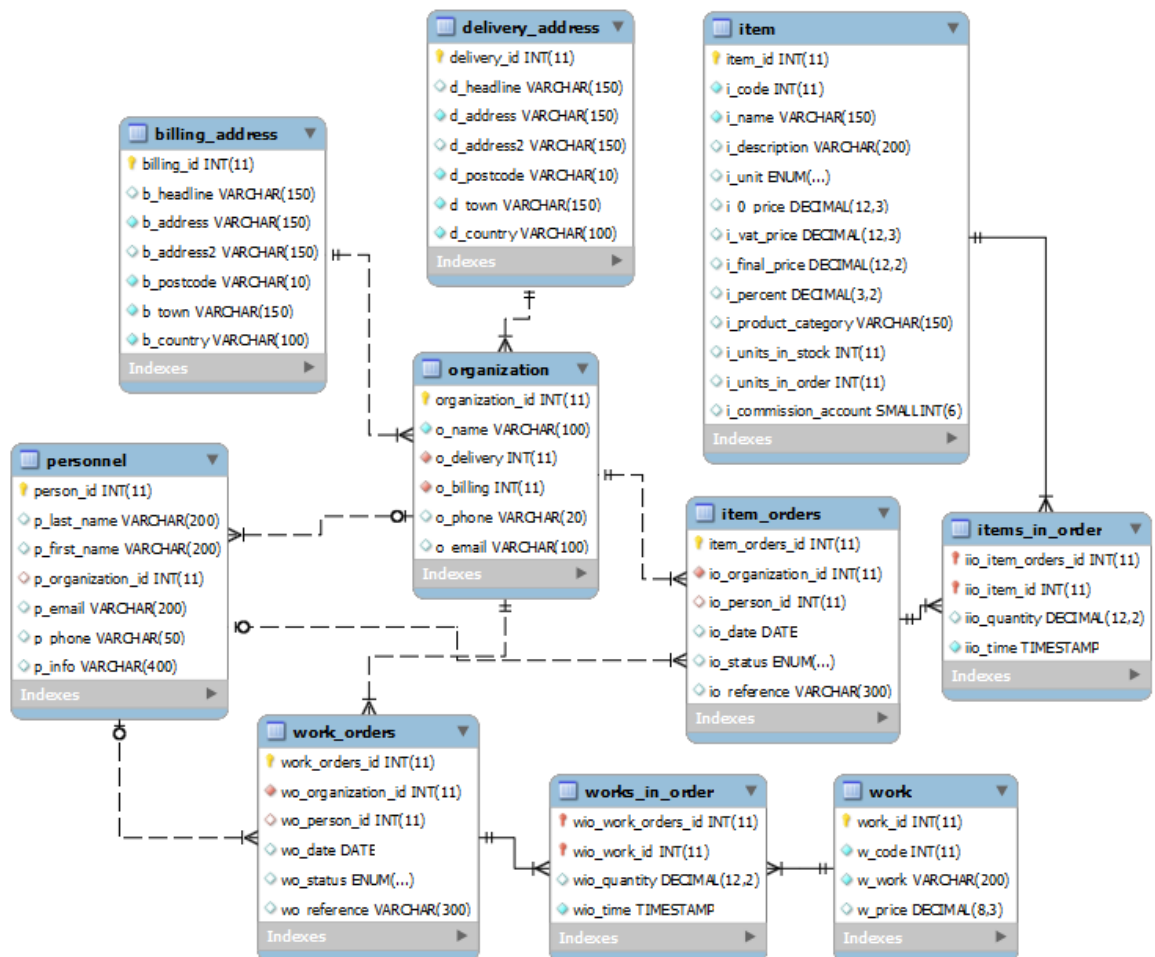
siten pysyvään palvelinkäyttöön. Vakaan julkaisun huonona puolena pidetään sen paketteja, jotka ovat tyypillisesti vanhentuneita siinä vaiheessa kun julkaisuversion voi mieltää vakaaksi. (Debian Project 2013.)

3 Sovellustoteutus

Sovelluksen tarkoitus on toimia asiakas- ja tilaustietojen sekä tuote- ja palveluvalikoiman hallintaohjelmana. Tämä osio kertoo sovelluksen kehityksestä ja toiminnallisuudesta.

3.1 Tietokannan suunnittelu ja toteutus

Sovelluksen työstäminen alkoi tietokannan suunnittelusta. Sovellus tulisi tarvitsemaan taulut yrityksen palveluille ja tuotteille, asiakkaille ja asiakasorganisaatioille, työtilauksille sekä tarviketilauksille. Suunnitelma normalisoitiin atomisuuden periaatteita noudattaen, jolloin asiakkaiden osoitetiedot saivat omat taulunsa, sekä tilausten ja tuotteiden välille suunniteltiin välitaulut. (Connolly & Begg 2010, 378-380.)



Kuvio 3. Tietokannan taulut kuvattuna MySQL Workbench –ohjelmalla.

Käytännön syyt johtivat siihen, että osoitteistotaulut toteutettiin erikseen laskutus- ja toimintusosoitteille. Lopullinen kanta koostuu kymmenestä taulusta.

3.2 Kehykset ja sovellus

Sovellusta varten suunniteltiin ja HTML-tekniikalla toteutettiin yksinkertainen Internet-sivusto, sovelluskehukset. Kehys koostuu index-sivun lisäksi kuudesta pääsivusta: sopimusasiakkaat, henkilöstö, palvelut, tuotteet, tarviketilaukset ja työtilaukset. Jokaisen sivun yläalaidassa on valikko, josta pääsee jokaiselle pääsivulle tai pääsivun mahdollisille alisivuille.

3.2.1 Sopimusasiakkaat

Sopimusasiakkaat-sivun pääsivulla listataan yrityksen asiakasorganisaatiot nimen mukaan. Organisaation nimen klikkaaminen avaa uuden sivun, joka näyttää asiakkaan tarkempia tietoja: mahdollinen puhelinnumero (vaihe), mahdollinen yleinen sähköpostiosoite, sekä toimitus- ja laskutusosoitteet. Toimitusosoite tarkoittaa oletusosoitetta, johon yritys toimittaa tehdyt tilaukset. Molemmat osoitetiedot koostuvat otsakkeesta, kahdesta osoiterivistä, postinumerosta ja toimipaikasta sekä maasta.

Pääsivulla asiakkaan nimen vieressä voi myös avata muokkaa-sivun, josta yllämainittuja tietoja voi muokata. Lomake tallentaa käyttäjän syöttämät tiedot niiden oikeudellisuudesta riippumatta. Tyhjäksi jätetty rivi korvaa aiemmin tallennetun tiedon tietokannassa tyhjällä.

Pääsivun asiakkaan nimen vierestä voi myös poistaa valitun asiakasorganisaation. Organisaatiota ei voi poistaa jos tämän alaisuuteen on määritelty henkilöstöä. Tämä johtuu tietokannan suhteista: organisaatiota ei voi poistaa, jos kannassa on yksikin henkilö jonka tiedot viittaavat organisaatioon.

Käyttäjä voi lisätä uuden organisaation valikosta. Avautuva lomake kysyy samoja tietoja kuin organisaation muokkauslomake. Pakollisia tietoja, joita ilman asiakkuutta ei voi lisätä, ovat nimeke sekä toimitus- ja laskutusosoitteen ensimmäinen rivi postinumerot ja toimipaikat. Käyttäjän on siis lisättävä kyseiset tiedot aluksi, sillä niiden oletetaan olevan yritykselle tarpeen, mutta tilanteen mukaan tietoja voidaan poistaa.

3.2.2 Henkilöstö

Henkilöstö-pääsivu on lista kaikista järjestelmään määritellyistä henkilöistä, asiakasorganisaatioiden työntekijöistä. Lista kertoo näiden yksilöllisen tunnusteen, etunimen, sukunimen, minkä organisaation alaisuuteen nämä kuuluvat, mahdollisen sähköpostin ja mahdollisen puhelinnumeron. Lisäksi yksi sarake on jätetty mahdollista lisäinformaatiota varten.

Listasta pääsee muokkaamaan henkilöiden tietoja tunnisteet pois lukien. Tunniste määräytyy automaattisesti henkilön luonnin yhteydessä.

Lopuksi henkilö on mahdollista poistaa listalta ja järjestelmästä kokonaan. Tämä edellyttää, ettei henkilöä ole merkattu yhdenkään tilauksen tekijäksi, sillä tietokannan tilaustaulujen tilaaja-merkintä viittaa määrättyyn henkilöön.

Valikosta voi valita henkilön hakutoiminnon. Henkilöjä voidaan hakea nimen tai asiakasorganisaation perusteella. Nimen perusteella haettaessa sovellus näyttää yhteensopivat osumat kaikkien järjestelmän henkilöiden etu- tai sukunimen perusteella. Hakutoiminto etsii kaikkialta nimitiedoista, eikä se edellytä isoja alkukirjaimia. Esimerkiksi haku "ilka" tuottaa tuloksina sekä Ilkan että Hilkkamajan. Organisaatiohaku toimii vastaavasti etsien kaikkialta organisaation nimestä.

Valikosta voi myös valita henkilön luontitoiminnon. Lomakkeeseen syötetään pakollisina tietoina henkilön etu- ja sukunimi sekä organisaatio. Sähköposti, puhelinnumero sekä lisäinformaatio ovat vapaaehtoisia tietoja.

3.2.3 Palvelut

Palvelut-pääsivulla on luettelo yrityksen tarjoamista palveluista. Jokaisella palvelulla on yksilöllinen tunniste sekä erillinen tuotekoodi. Tuotekoodi on oletuksellisesti yksilöllinen, mutta tilanteesta riippuen sovellukseen voidaan lisätä sama tuote useasti esimerkiksi eri hinnalla tai tuotekuvauksella. Jokaisella palvelulla on nimeke ja myyntihinta. Tuotekoodia, nimekettä ja hintaa voidaan muokata pääsivulta, sekä palvelu voidaan poistaa tilanteessa jossa se ei sisälly tehtyihin tilauksiin.

Pääsivulta voidaan lisätä uusi palvelu. Tuotekoodi, kuvaus ja hinta ovat kaikki pakollisia tietoja, kun taas tunniste määräytyy automaattisesti. Hinta voidaan määrittää kahden desimaalin tarkkuudella, ja ilmoittaa käyttäen pilkkua desimaalierottajana. Sovellus kääntää pilkun pisteeksi kun palvelua kirjataan tietokantaan.

Sovelluskehityksen päättyessä pääsivun palvelulista koostui pääasiassa tiedoista, jotka on lisätty tietokantaan `load data infile` -komennolla.

3.2.4 Tuotteet

Tuotteet-pääsivulla on lista yrityksen myytävistä tarvikkeista. Lista koostuu noin 5200 nimekkeestä, jotka kuuluvat yrityksen tavarantoimittajan valikoimiin.

Tuotteet						
Lukumäärä: 5212						
Tuotekoodi	Nimeke	Yksikkö	Myyntihinta	Hyllyssä	Muokkaa	Poista
146103	Verkkokaapeli CAT5 UTP RJ45	kpl	12.29	0	Muokkaa	Poista
146136	Verkkokaapeli CAT5 UTP RJ45	kpl	2.81	0	Muokkaa	Poista
146135	Verkkokaapeli CAT5 UTP RJ45	kpl	3.67	0	Muokkaa	Poista
146178	Virtakaapeli, maadoitettu, 3 m	kpl	7.53	0	Muokkaa	Poista
146187	Logitech Z130 kaiutin 2.0	kpl	38.07	0	Muokkaa	Poista
204704	Canon EOS 550D EF-S 18-55IS	kpl	1026.16	0	Muokkaa	Poista
204705	Canon IXUS 240 HS	kpl	364.93	0	Muokkaa	Poista
204706	Canon PowerShot A3500 IS	kpl	187.49	0	Muokkaa	Poista
146185	Kuulokkeet Logitech PC 960	kpl	41.00	0	Muokkaa	Poista
207067	Lehtiötaulu 650 x 920 mm	kpl	190.84	0	Muokkaa	Poista
207042	Lehtiötaulu 700 x 1000 mm	kpl	202.55	0	Muokkaa	Poista
207069	Lehtiötaulu 750 x 1075 mm	kpl	236.03	0	Muokkaa	Poista
207043	Lehtiötaulu 750 x 1150 mm	kpl	249.43	0	Muokkaa	Poista
207068	Lehtiötaulu 760 x 1100 mm	kpl	602.64	0	Muokkaa	Poista
207066	Lehtiötaulu 760 x 1100 mm	kpl	582.55	0	Muokkaa	Poista
207044	Lehtiötaulu 850 x 1200 mm	kpl	639.47	0	Muokkaa	Poista
207022	Lehtiötaulu Dahle 670 x 900 mm	kpl	225.99	0	Muokkaa	Poista
204600	LED Näyttö BENQ	kpl	286.25	0	Muokkaa	Poista
204603	LED Näyttö LG	kpl	264.49	0	Muokkaa	Poista

Kuva 2. Tuoteluettelo, osa.

Kuva 2 esittää pääsivun tuotenäkymää. Tuotelista on sivutettu siten, että sata tuotetta näytetään kerralla. Tuotteesta kerrotaan ensisijaisesti tuotekoodi, nimeke, yksikkö, myyntihinta ja kuinka paljon yrityksellä on kyseistä tuotetta varastossa.

Tuotekoodia klikkaamalla käyttäjä pääsee näkemään tuotteen kaikki tiedot: edellä mainittujen lisäksi, tarkemman kuvauksen, tuotekategorian, kuinka paljon yrityksellä on tuotetta tilauksessa, hintatietoja sekä kirjanpidollisen myyntitilin. Päänäkymästä pääsee myös muokkaamaan valitun tuotteen tietoja, sekä poistamaan tuotteen. Poisto ei ole mahdollista, jos järjestelmään on syötetty tilaus johon kyseinen tuote sisältyy.

Tuotteet-sivuilla on lisäksi uuden tuotteen manuaalinen lisäystoiminto, sekä hakutoiminto.

[Kaikki tuotteet](#) [Lisää tuote](#)

Hakutulokset

4 osumaa

Tuotekoodi	Nimeke	Yksikkö	Hinta
618008	Pahvilaatikko A2 kpl		3.13
618007	Pahvilaatikko A3 kpl		1.72
618006	Pahvilaatikko A4 kpl		0.60
618005	Pahvilaatikko A5 kpl		0.84

[Takaisin](#)

Kuva 3. Tuotehaku.

Kuva 3 esittää hakutuloksia. Haku kohdistuu kaikkien kannan tarvikkeiden tuotekoodiin, nimekkeeseen ja kuvaukseen. Etsittävää merkkijonoyhdistelmää etsitään kaikkialta kyseisistä kentistä, jolloin esimerkiksi hakusana näyttö listaa sekä näytöt että näyttökaapelit. Tuotekoodin linkki näyttää tuotteen kaikki tiedot.

3.2.5 Työtilaukset

Työtilausten pääsivulla on lista järjestelmän syötetyistä tilauksista. Tilauksista näytetään näiden tunniste, tilaajan nimi ja organisaatio, tilauspäivämäärä ja tilanne (aloittamatta, kesken, toimitettu, laskutettu). Tilauksia näytetään pääikkunassa sata kerrallaan.

Tunnistetta klikkaamalla käyttäjä näkee tilauksen tarkemmat tiedot.

Tilaus 3

Asiakas: Firma Oy

Tilaaaja: Mikko Miettinen

Päivämäärä: 2014-09-06

Status: kesken

Lisätiedot: Toimitus viim. ke klo 12

Sisältö

Koodi	Nimeke	Hinta	Kpl	Yht.	Poista
102	Kopiot A4 1-20	0.118	160.00	18.88	Poista
103	Kopiot A4 21-100	0.106	240.00	25.44	Poista
106	Niitti	0.031	50.00	1.55	Poista
107	Lajittelu	0.022	200.00	4.4	Poista
117	Hyvitys 2-puol. kopiointi	-0.009	200.00	-1.8	Poista

Yhteensä: 48.47 euroa (sis. alv)

Lisää rivi: Koodi Määrä

Kuva 4. Työtilauksen tiedot.

Kuva 4 havainnollistaa tilauksen tarkkoja tietoja. Tilaukset otsikoidaan tilausnumeron mukaan. Asiakkaasta kerrotaan organisaatio ja tilaajan nimi. Tilauspäivämääräksi määräytyy se päivä, jolloin tilaus syötetään järjestelmään.

Tilauksen sisältö syötetään rivi kerrallaan (tuotekoodi ja määrä) tilauksen sivun kautta. Sovellus kertoo tuotteen hinnan määrällä, ja laskee syötetyt rivit yhteen tilauksen loppusummaksi. Tuotteet on hinnoiteltu yrityksen tarpeiden mukaan eurosentin kymmenesosan tarkkuudella, mutta yhteen kerrottaessa tulo pyöristetään sentin tarkkuuteen.

4 Pohdinta

Opinnäytetyön ideana oli suunnitella ja luoda tiedonhallintajärjestelmä toimistopalvelualan yrityksen tarpeisiin. Tavoitteena oli sovellus, jolla voi hallinnoida yrityksen asiakasrekisteriä, tuotevalikoimaa ja osittain tilauksia. Projekti alkoi virallisesti aloituskokouksen yhteydessä 25.3.2014. Opinnäytetyön arvioitu arviointipäivämäärä oli 4.9.2014, mutta työ arviointiin arviointikokouksessa 2.10.2014. Aikataulu myöhästyi siis noin kuukaudella.

Työ onnistui pääosin, kuten luvussa 4.1 on eritelty. Tämän kaltainen sovelluskehitys jättää kuitenkin paitsi oikeasti aiheellisia puutteita, myös jatkokehitettävää niin paljon kuin mielikuvitus antaa periksi.

4.1 Tulokset

Sovellus sisältää pääosin kaiken toiminnallisuuden, jonka sen piti sisältää. Asiakasorganisaatioita voi lisätä, katsoa ja niiden tietoja muokata. Organisaatioita voi poistaa tietyin edellytyksin. Organisaatioiden alaisuuteen voi lisätä henkilöstöä, näiden tietoja voi selata ja muokata, ja nämä voi poistaa tietyin edellytyksin. Tuotteita, niin tarvikkeita kuin palveluja, voi lisätä kerrallaan tai suuren määrän kerralla. Työtilauksia sisältöineen voi lisätä ja lukea. Sovellus toimii julkisessa Internetissä olemattomalla viiveellä. Tietokanta toimii nopeasti.

4.2 Kehitysehdotukset ja eettiset kysymykset

Vaikka projekti on pääosin valmis, jäi siihen vielä kehitettävää. Ajanpuute muodostui ongelmaksi, ja projekti jäi osittain keskeneräiseksi.

4.2.1 Tarviketilaukset ja työtilaukset

Suurin puute on tarviketilauksen puuttuminen kokonaan. Kyseinen sovelluksen haara toimii hyvin samankaltaisesti kuin työtilaukset, joten puutoksen lisäys olisi ollut enemmän ajalla hyvin mahdollinen. Tosin aikarajan salliessa olisin kehittänyt ensin työtilaukset loppuun asti. Toiseksi suurin puute projektissa on se, ettei työtilauksista voi poistaa tuotteita, eikä tilaustietoja voi muuttaa eikä tilauksen tuotteita poistaa.

Tarkoituksena oli myös, että tilauspäivämäärän voi määrittää itse haluamakseen. Ihanteellisessa sovelluksessa tilauksen voisi merkitä erillisellä aikamääreellä loppuun asti toimitukseksi, sekä vastaavasti laskutusajankohdan voisi merkitä halutulle ajankohdalle. Valmistuneessa versiossa tilaukseen voi lisätä tuotteita milloin vain, ja näiden tilaustuotteiden

yksittäiset rivien syöttöhetki merkitään sentään sekunnin tarkkuudella kantaan. Tosin tätä ajankohtaa ei näy itse sovelluksessa.

Oli myös idea, että uuden tilauksen luonnin yhteydessä käyttäjä voi lisätä samalla sivulla haluamansa määrän tilausrivejä kulkematta erikseen sivulta toiselle. Haastetta ratkottiin yhdistämällä JavaScriptiä ja PHP:tä, mutta lopulta aika loppui kesken.

4.2.2 Työtilaukset, pienet asiat

Työtilauksissa on muita pieniä asioita, jotka voisi aikataulun salliessa korjata. Työtilausten pääsivu voisi näyttää kaikkien tilausten sijaan aloittamattomiksi tai keskeneräisiksi merkatut työt. Toimitetut ja laskutetut voitaisiin listata erikseen. Sivulla voisi etsintätoiminto organisaaation, tilaajan, päivämäärän ja tuotteiden mukaan. Tilauksen yhteishinta ei tallennu tietokantaan, vaan se näkyy pelkästään sovelluksessa PHP-laskutoimituksena.

4.2.3 Hinnoittelu

Suurempi uudistuskohde on tuotteiden hinnoittelu tapauskohtaisesti. Yrityksellä tulee toisinaan vastaan tilanteita, joissa tuote myydään eri hinnalla kuin hinnastossa on määritelty. Tämä vaatisi tuotteen oletushinnan hakua tietokannasta tilauslomakkeelle muutettavaksi, ja lopullinen hinta tallentuisi tilauksen ja tuotteen välitaulun yhteyteen.

Tuotteiden ja palvelujen hinnat ja näiden tilausmäärät näytetään kaikkialla sovelluksessa kahden tai kolmen desimaalin tarkkuudella, ja aina pistettä desimaalierottimena käyttäen. Tähän on syynä tietojen tulostaminen sellaisenaan kun se tietokannasta tulee, ja MySQL:ssä numeraalisten tietojen desimaalierotin on piste. Tulostettavan datan pisteiden kääntäminen pilkkuiksi lienee toteutettavissa samankaltaisin menetelmin kuin kantaan tallennettavan numeraalisen datan pilkkujen kääntäminen pisteiksi.

4.2.4 Tarvikkeet ja palvelut

Tarvikepääsivun koko tarvikelistan sijaan oletussivu voisi listata ne tavarat, joita hyllyssä on, tavarat joita hyllyssä tulisi olla mutta ovat tilapäisesti loppu. Tarvikkeiden yhteydessä kerrottu myyntitili jäi projektissa alustavasti tarpeettomaksi, sillä sitä yrityksessä tarvitaan vain kirjanpidollisissa asioissa ja tilastoinnissa. Nämä yrityksen toiminnan piirteet jäivät kokonaan projektin ulkopuolelle. Myyntitili kuuluisi periaatteessa listata myös palveluille.

On kyseenalaista, onko hyvä ratkaisu pitää tarvikkeiden ja palveluiden tuotekoodit ja tunnisteet erillisinä tietoina. Sama tuotekoodi eri tunnisteilla mahdollistaa teoriassa erilaiset nimet, kuvaukset ja hinnoittelut, mutta nykyisen ohjelmakoodin se saa oireilemaan silloin

kun tietty tuotekoodi pitäisi lisätä tilaukseen, mutta kyseistä koodia vastaavia tuotteita on useampi. Ongelman voi yrittää ratkaista samojen tuotekoodien käytöllä, mutta itse en tätä kirjoittaessa tunne ratkaisua. Saman tuotteen syöttäminen eri tunnisteilla ja hinnalla tai kuvauksella olisi ainakin yksinkertaisempi ratkaisu. Käytettävyyden ja tehokkuuden kannalta voi silti olla tutkimisen arvoista, löytyykö saman tunnisteiden käytölle ratkaisu.

4.2.5 Pois rajatut aiheet

Projektin ulkopuolelle jäi kokonaan yrityksen kolmas päälaskutusaihe, postitukset. Postitustilausten sisällyttäminen sovellukseen olisi ollut mahdollista enemmän ajalla.

Sovellukseen voisi myös lisätä laskutusvaiheen. Tilauksista jotka ovat loppuun asti muutten käsitelty, voidaan muodostaa koontilasku. Lasku voi sisältää vaihtuvien tietojen lisäksi pohjan (esimerkiksi tilinumeron, maksuehdot) ja se voidaan päivätä halutulle päivälle. Laskujen tekeminen on osa yrityksen toimintaa, mutta se rajattiin kirjanpidollisten raporttien kanssa heti alussa projektin ulkopuolelle todennäköisyyden vaativuuden takia.

4.2.6 Lomakkeet ja relaatiösäännöt

Uuden tilauksen tai asiakkaan luonnin yhteydessä luotava kohde rinnastetaan johonkin tilaajaan tai organisaatioon. Projektin päättyessä tämä tapahtuu ilmoittamalla ko. tahon tunniste. Käyttökelpoisen sovelluksen olisi hyvä tarjota jonkinlaisia pudotusvalikkoja rinnastamista varten. Vaihtoehdot olisivat saatavilla tietokannasta.

Tietojen poistoa sovelluksesta haittaavat tietokannan relaatiösäännöt. Mitään ei voida poistaa, jos siihen viitataan toisaalla. Tuotetta ei voida poistaa jos sitä on joskus tilattu, asiakasta ei voida poistaa jos tämä on tehnyt tilauksen, organisaatiota ei voida poistaa jos sen alaisuudessa on henkilöstöä. Yksi ratkaisu tähän voisi tilausten osalta olla tuotteen tietojen kopiointi erilliseen tauluun tilauksen luonnin yhteydessä, ja tilauksen sisällöstä kyseiseen tauluun viittaaminen. Henkilöiden ja organisaatioiden poisto voidaan kiertää lisäämällä näiden tietoihin jonkinlainen status, joka määrittää onko henkilö sillä hetkellä organisaationsa palveluksessa, tai esimerkiksi lopettanut. Organisaatiot voidaan vastaavasti merkitä esimerkiksi luottokelpoisuuden menettäneiksi tai toimintansa lopettaneiksi.

4.2.7 Graafinen ilme

Sovelluksen visuaalisuus ei ollut missään vaiheessa osa projektin tavoitteita. Melkein jokainen tekstisyöttökenttä on pituudeltaan HTML:n oletusmallia. Lomakkeita ei ole tasattu siistiksi, eikä sovelluksessa ole muita värejä käytössä kuin HTML:n oletusvärit, pääasias-

sa musta ja valkoinen. Sovelluksen tekniikoihin ei kuulu ollenkaan CSS:aa tai tyylimäärittelyä. Itse asiassa allekirjoittanut teki visuaalisempia sivuja jo 1998 ala-asteella. Ulkoasuun olisi voinut kiinnittää huomiota ylimääräisellä ajalla.

4.2.8 Tietoturvaluus

Sovellus toimi Internetissä julkisella sivulla, mutta sen käyttö ei ollut juurikaan suojattu. Sivut joista ajettiin `load data infile` –komennot on suojattu salasanoin PHP-tekniikalla, mutta muuten kuka tahansa pystyy periaatteessa tekemään sivulla mitä tahansa. Sivustolle oli kuitenkin hankala löytää. Google ei sitä löytänyt, eikä kovin moni henkilö tiennyt sivuston olemassaolosta, joten jäi nähtäväksi miten kukaan olisi sen vahingossa löytää. Sovelluksen olisi voinut myös suojata Apache-palvelimen `.htaccess`-määrittelyllä.

4.3 Opinnäytetyöprosessin ja oman oppimisen arviointi

Opinnäytetyön tekeminen tuntui kokemuksena hyödylliseltä. Noin neljänsadan tunnin projektityö oli sopivan kokoinen valmistumista edeltävä haaste. Vaikka ammattikorkeakoulun pitäisi olla nimenomaan käytännöllinen opinahjo, en ollut tehnyt mitään tämän tapaista käymilläni kursseilla. Täysi vastuu kaikesta, laaja-alainen aihe jota ei ehtinyt saattaa läheläkään sellaista tilaa jonka voisi itse mieltää valmiiksi (vaikka tämän kaltaiselle sovellukselle voikin aina keksiä lisää toteutettavia ominaisuuksia) ja itsenäinen työskentely tuntuivat ”oikean elämän” koululta. Pidän nykymuotoista opinnäytetyötä vaihtoehtoiseen (projekti tai tutkimus) hyvänä käytäntönä.

Keksin aiheen opinnäytetyölleni noin vuoden ennen projektin alkua. Työnteko haittasi projektin etenemistä, mutta muutaman venähtäneen illan ansiosta sain lopulta projektin siihen pisteeseen, että se voitaisiin arvioida. Kuvaavaa lienee, että projektisuunnitelman kirjoittamiseen meni useampi kuukausi.

Kokemukseni PHP:stä oli ennen projektin alkua melko olematon, ja tämän yhteensovittaminen MySQL:n kanssa täysin tuntematonta. Asioita oppiessani tajusin, että kyseessä on ainakin perusasioissa melko yksinkertainen ohjelmointikieli. MySQL:ää tunsin jonkin verran ennestään, mutta sen käytöstä sain toki oppia uusia asioita. Etenkin `load data infile` –komento osoittautui erittäin käteväksi kun sen oppi.

Mielenkiintoisin vaihe projektissa oli tietokannan suunnittelu ja toteutus. Vaikka lopputulos oli yksinkertainen, se vastasi sovelluksen tarpeita. Haastavinta oli itse toiminnallisuuden kehitys eli niin sanottu PHP-osuus. Melkein jokainen yksittäinen sivu ja sivun toiminnallisuus oli oma haasteensa. Oppiminen oli hidasta mutta tehokasta.

5 Lähteet

Connolly, T. & Begg, C. 2010. Database Systems: A Practical Approach to Design, Implementation and Management. Fifth Edition. Addison-Wesley. Boston.

Debian Project 2013. The Debian GNU/Linux FAQ: Chapter 1 – Definitions and overview. Luettavissa: https://www.debian.org/doc/manuals/debian-faq/ch-basic_defs.en.html. Luettu: 2.9.2014.

The Internet Society 2005. Common Format and MIME Type for Comma-Separated Values (CSV) Files. Luettavissa: <http://www.ietf.org/rfc/rfc4180.txt>. Luettu: 6.9.2014.

MySQL 2014 a. About MySQL. Luettavissa: <http://www.mysql.com/about/>. Luettu: 2.9.2014.

MySQL 2014 b. LOAD DATA INFILE Syntax. Luettavissa: <http://dev.mysql.com/doc/refman/5.1/en/load-data.html>. Luettu: 4.9.2014.

The PHP Group 2014 a. History of PHP: PHP Tools, FI, Construction Kid, and PHP/FI. Luettavissa: <http://php.net/manual/en/history.php.php>. Luettu: 1.9.2014.

The PHP Group 2014 b. What is PHP? Luettavissa: <http://fi1.php.net/manual/en/intro-what-is.php>. Luettu: 6.9.2014.

Raggett, D., Lam,, J., Alexander, I., Kmiec, M. 1998. A History of HTML. Luettavissa: <http://www.w3.org/People/Raggett/book4/ch02.html>. Luettu: 3.9.2014.

6 Liitteet

6.1 Liite 1. Tietokannan taulujen luontilauseet

```
create table delivery_address (  
delivery_id int auto_increment not null primary key,  
d_headline varchar(150),  
d_address varchar(150) not null,  
d_address2 varchar(150),  
d_postcode varchar(10) not null,  
d_town varchar(150) not null,  
d_country varchar(100) not null  
) engine=InnoDB;
```

```
create table billing_address (  
billing_id int auto_increment not null primary key,  
b_headline varchar(150),  
b_address varchar(150) not null,  
b_address2 varchar(150),  
b_postcode varchar(10) not null,  
b_town varchar(150) not null,  
b_country varchar(100) not null  
) engine=InnoDB;
```

```
create table organization (  
organization_id int not null auto_increment primary key,  
o_name varchar(100) not null,  
o_delivery int not null,  
o_billing int not null,  
o_phone varchar(20),  
o_email varchar(100),  
foreign key (o_delivery) references delivery_address(delivery_id),  
foreign key (o_billing) references billing_address(billing_id)  
) engine=InnoDB;
```

```
create table personnel (  
person_id int not null auto_increment primary key,  
p_last_name varchar(200),  
p_first_name varchar(200),
```



```

p_organization_id int,
p_email varchar(200),
p_phone varchar(50),
p_info varchar(400),
foreign key (p_organization_id) references organization(organization_id)
) engine=InnoDB;

```

```

create table item (
item_id integer not null auto_increment primary key,
i_code integer not null,
i_name varchar(150) not null,
i_description varchar(200),
i_unit enum('kpl', 'pkt', 'srj', 'pak', 'ltk', 'ark', 'rs', 'rll', 'pss',
'ras', 'pari', 'kut', 'nid', 'nip', 'pll', 'ptk', 'rsi', 'sk', 'tus'),
i_0_price decimal(12,3),
i_vat_price decimal(12,3),
i_final_price decimal(12,2),
i_percent decimal(3,2),
i_product_category varchar(150),
i_units_in_stock integer default '0',
i_units_in_order integer default '0',
i_commission_account smallint
) engine=InnoDB;

```

```

create table work (
work_id integer not null auto_increment primary key,
w_code integer not null,
w_work varchar(200) not null,
w_price decimal(8,3)
) engine=InnoDB;

```

```

create table work_orders (
work_orders_id integer auto_increment not null primary key,
wo_organization_id int not null,
wo_person_id int,
wo_date date,
wo_status enum('aloittamatta', 'kesken', 'toimitettu', 'laskutettu'),
wo_reference varchar(300),

```

```
foreign key (wo_organization_id) references organization(organization_id),
foreign key (wo_person_id) references personnel(person_id)
) engine=InnoDB;
```

```
create table item_orders (
item_orders_id integer auto_increment not null primary key,
io_organization_id int not null,
io_person_id int,
io_date date,
io_status enum('aloittamatta', 'kesken', 'toimitettu', 'laskutettu'),
io_reference varchar(300),
foreign key (io_organization_id) references organization(organization_id),
foreign key (io_person_id) references personnel(person_id)
) engine=InnoDB;
```

```
create table works_in_order (
wio_work_orders_id integer not null,
wio_work_id integer not null,
wio_quantity decimal(12,2),
wio_time timestamp not null,
foreign key (wio_work_id) references work(work_id),
foreign key (wio_work_orders_id) references work_orders(work_orders_id),
primary key (wio_work_orders_id, wio_work_id)
) engine=InnoDB;
```

```
create table items_in_order (
iio_item_orders_id integer not null,
iio_item_id integer not null,
iio_quantity decimal(12,2),
iio_time timestamp not null,
foreign key (iio_item_id) references item(item_id),
foreign key (iio_item_orders_id) references item_orders(item_orders_id),
primary key (iio_item_orders_id, iio_item_id)
) engine=InnoDB;
```

6.2 Asiakasorganisaatioiden tulostaminen listana

```
<?php

$connect=mysqli_connect("palvelin", "käyttäjätunnus", "salasana", "tietokanta");
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error(); }

$result = mysqli_query($connect, "select organization_id, o_name from organization");

echo "<table border='0'>
<tr>
</tr>";

while($row = mysqli_fetch_array($result)) {
echo "<tr>";

echo "<td><a href=organization_info.php?id="
.$row['organization_id']. ">".$row['o_name']. "</a> ";

echo "<td><a href=organization_mod.php?id="
.$row['organization_id']. ">(muokkaa)</a> ";

echo "<td><a href=organization_del.php?id="
.$row['organization_id']. ">(poista)</a><br>";

echo "</tr>";
}

mysqli_close($connect);
?>
```

6.3 Asiakashenkilön tietojen muokkaus lomakkeella, ensimmäinen sivu

```
<?php
$conn = new mysqli('palvelin', 'käyttäjä', 'salasana', 'tietokanta');
if (mysqli_connect_errno()) { exit('Connect failed: '. mysqli_connect_error()); }

$p_id = $_GET['id'];
echo "<h3>Muokkaa: tunniste nro " . $p_id. "</h3>";

$sql = "select * from personnel
where person_id = '$p_id'";
$result = mysqli_query($conn, $sql);
while ($row = mysqli_fetch_array ($result)){

?>

<form method="post" action="personnel_mod2.php">
<input type="hidden" name="id" value="<?php echo $row['person_id']; ?>">
Etunimi: <input type="text" name="firstname" value="<?php echo $row['p_first_name'];
?>"><br>
Sukunimi: <input type="text" name="lastname" value="<?php echo $row['p_last_name'];
?>"><br>
Asiakkuus: <input type="text" name="org" value="<?php echo $row['p_organization_id'];
?>"><br>
Puhelinnumero: <input type="text" name="phone" value="<?php echo $row['p_phone'];
?>"><br>
Sähköposti: <input type="text" name="email" value="<?php echo $row['p_email'];
?>"><br><br>
Muu info: <input type="text" name="info" value="<?php echo $row['p_info']; ?>"><br><br>

<input type="submit" name="update" id="update" value="Muokkaa" />
</form>

<?php

}

?>
```

6.4 Asiakashenkilön tietojen muokkaus lomakkeella, toinen sivu

```
<?php

$errors = array();
$regexp_mail = '/^([a-zA-Z0-9]+[a-zA-Z0-9._%~]*@([a-zA-Z0-9]+\.)+[a-zA-Z]{2,4})$/';

if (isset($_POST['update']) ) {
    $_POST = array_map("strip_tags", $_POST);
    $_POST = array_map("trim", $_POST);

    if(count($errors)<1) {
        $conn = new mysqli('palvelin', 'käyttäjä', 'salasana', 'tietokanta');
        if (mysqli_connect_errno()) { exit('Connect failed: '. mysqli_connect_error()); }

        $p_id = $conn->real_escape_string($_POST['id']);
        $p_first_name = $conn->real_escape_string($_POST['firstname']);
        $p_last_name = $conn->real_escape_string($_POST['lastname']);
        $p_org = $conn->real_escape_string($_POST['org']);
        $p_phone = $conn->real_escape_string($_POST['phone']);
        $p_email = $conn->real_escape_string($_POST['email']);
        $p_info = $conn->real_escape_string($_POST['info']);

        $query = "update personnel set p_first_name='$p_first_name',
        p_last_name='$p_last_name', p_organization_id='$p_org', p_email='$p_email',
        p_phone='$p_phone', p_info='$p_info'
        where person_id = '$p_id'";

        if ($conn->query($query) === TRUE) { echo '<h3>Henkilötiedot päivitetty</h3>'; }
        else { echo 'Virhe: '. $conn->error; }
        $conn->close(); }

    else { echo implode('<br />', $errors); } }

?>
```