

Opinnäytetyö (AMK)

Tietojenkäsittely

Sähköisen liiketoiminnan järjestelmät

2010

Topias Laaksonen ja Aleksis Mäkeläinen

# LASKUTUSJÄRJESTELMÄN TOTEUTUS MS ACCESS 2007:llä

Ykkösmiehet Oy



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

TURUN AMK:N OPINNÄYTETYÖ | Topias Laaksonen ja Aleksis Mäkeläinen

## Topias Laaksonen ja Aleksi Mäkeläinen

# LASKUTUSJÄRJESTELMÄN TOTEUTUS MS ACCESS 2007:LLÄ

Tämä opinnäytetyö tehtiin toimeksiantona Ykkösmiehet Oy:lle, jolla oli tarvetta uudelle, vaihtoehtoiselle laskutusjärjestelmälle.

Järjestelmän toteutus jakautui useaan eri vaiheeseen. Aluksi kartoitettiin asiakkaan tarpeet, vaatimukset, resurssit ja kriteerit. Tilannekartoitusta seurasi järjestelmällinen sovelluksen toteuttaminen niin, että jokainen haluttu kriteeri ja yksityiskohta onnistuttaisiin toteuttamaan annettujen vaatimusten perusteella. Työn edetessä sovelluksesta valmistui useita eri raakaversioita, joita tarkastelimme yhdessä yrityksen edustajan kanssa ja loimme kuvaa tulevasta sovelluksen viimeisestä versiosta.

Sovellus tehtiin MS-Access 2007 tietokantaohjelmistolla pienyrittäjän laskustarpeita ajatellen. Tässä projektissa hyödynsimme lisäksi useita eri pienohjelmia, kuten esimerkiksi ScreenHunter -kuvankaappausohjelmistoa. Laskutusjärjestelmälle asetettiin selkeät tavoitteet työn hyväksymiseksi. Sovelluksen toimivuuden kannalta oleelliset tiedot kerättiin puhelimitse ja kasvotusten toteutuneilla henkilöstöhaastatteluilla.

Projektissa ilmeni myös muutamia vastoinkäymisiä, jotka johtuivat lähinnä sovellukseen syötettyjen tietojen toimimattomuudesta ja tietyistä rakenteellisista ominaisuuksista tai niiden puutteista. Ongelmat ovat kuitenkin pääosin ratkaistu ja sovelluksen viimeinen versio valmistuu sovittujen aikataulujen mukaisesti joulukuussa 2010.

Tämän projektin tuloksena valmistui toimiva MS-Access 2007 pohjainen laskutusjärjestelmä. Uusi laskutusjärjestelmä mahdollistaa asiakas-, tilaus-, palvelu-, lasku-, ja henkilöstötietojen hallinnoinnin usein eri toiminnoin.

Yhteistyö Ykkösmiehet Oy:n kanssa on sujunut erinomaisesti ja tiedonkulku toiminut ongelmitta. Toimeksiannon järjestelmällisen toteutumisen ehtona on ollut se, että yhteydenpito on ollut aukotonta ja molempien osapuolten näkemyksistä on saatu täysin yhtenevät. Tämä on edellytyksenä sille, että asiakastyytyväisyys ohjelmistoa kohtaan saavutetaan viimeisen version julkaisuvaiheessa, jolloin sovellus olisi tarkoitus ottaa virallisesti toimeksiantajan käyttöön.

### ASIASANAT:

MS-Access 2007, ohjelmistosuunnittelu, laskutusjärjestelmä

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Business Information Systems

January 2011 | 43 pages

Päivi Killström

Topias Laaksonen and Alekski Mäkeläinen

## Implementation of Invoicing System with MS Access 2007

This thesis was commissioned by Ykkösmiehet Ltd as they needed an updated Invoicing System.

The implementation of is divided into many phases. First we surveyed the needs of our commissioner and the possible resources and criteria for the upcoming program. After collecting all the required information for the upcoming program, we started doing it frequently so that every single criteria and detail would be according to requirements.

During the process, we managed to create many different versions which were analyzed in order to create the best and the most realistic final version of the upcoming program.

There were a few problems, mostly due to the difference between versions of the MS-Access 2003 and 2007. The problems were caused mainly because of lacking co-operation of information applied into program. Also some basic structures of the program brought challenges. However, problems have been mostly solved and the final version of the program will be published as planned.

The program was made by using MS-Access 2007 database tool. During the process we also used many different applets, for example ScreenHunter –screenshot program. The final version of our Invoicing System was given certain criteria for the approval by our commissioner. The information required was mainly gathered through phone conversations and face-to-face interviews.

Our co-operation with Ykkösmiehet Ltd was successful due to frequent communication and a proper co-operation to unite our views completely. During publishing stage the program will taken into use as planned at December 2010.

KEYWORDS:

MS-Access 2007, software design, Invoicing System

# SISÄLTÖ

Tiivistelmä

Abstract

<b>1 JOHDANTO</b>	<b>5</b>
<b>2 TAUSTATIEDOT</b>	<b>7</b>
2.1 Toimeksianto	7
2.2 Asiakasyritys	8
<b>3 OHJELMISTOTUOTANTO JA -SUUNNITTELU</b>	<b>9</b>
3.1 Sovelluksen vaihejako- ja elinkaarimalli	9
3.2 Ohjelmistosuunnittelu	11
3.3 Sovelluksen määrittely ja vaatimuskartoitus	12
3.4 Tietojen etsiminen ja järjestäminen	16
3.5 Tietojen jakaminen taulukoihin	17
3.6 Tieto-osien muuntaminen sarakkeiksi	18
3.7 Pääavainten määrittäminen	20
3.8 Taulukoiden välisten yhteyksien luonti	20
3.9 Käyttöliittymäsuunnittelu	21
<b>4 TESTAUS</b>	<b>25</b>
4.1 Testaussuunnitelma	25
4.2 Integraatiotestaus	27
4.3 Systeemitestaus	34
<b>5 KÄYTTÖÖNOTTO</b>	<b>40</b>
<b>6 OMAN TYÖN ARVIOINTI</b>	<b>41</b>
<b>LÄHTEET</b>	<b>43</b>

## KUVAT

Kuva 1. Ykkösmiehet Oy –logo	8
Kuva 2. Ohjelmiston vaihejakona sovellettu vesiputousmalli	10
Kuva 3. Sovellettu nopean kehityksen malli (RAD)	11
Kuva 4. Käyttötapaus tilauksen lisäämisestä asiakkaalle	13
Kuva 5. Käyttötapaus asiakastietojen päivityksestä	14
Kuva 6. Käyttötapaus tilauksen laskuttamisesta	15
Kuva 7. Tietojen etsiminen ja järjestäminen	17
Kuva 8. Asiakastiedot-taulu	18
Kuva 9. Asiakastaulun tieto-osat sarakkeina	19

Kuva 10. Tilaukset-aulun tietotyypivalinnat	19
Kuva 11. Taulukoiden väliset yhteydet relaatiotietokannassa	21
Kuva 12. Käyttöliittymäsuunnittelu käyttäjätoiminnan pohjalta	22
Kuva 13. Asiakastiedot-lomakkeen graafinen ulkoasu	23
Kuva 14. Tilaukset-lomakkeen graafinen käyttöliittymä	24
Kuva 15. Ohjelmistotestauksen suunnitteluvaiheet	26
Kuva 16. Navigointipainikkeiden toimivuus	27
Kuva 17. Uuden tilauksen luominen	29
Kuva 18. Virheilmoitus väärästä tietotyypistä	29
Kuva 19. <i>Palvelut</i> -taulun tietojen päivittyminen tietokantaan	30
Kuva 20. Asiakastiedot -lomakkeen ja asiakashistoria -raportin reaaliaikainen päivittyminen	31
Kuva 21. Lisätyn asiakastiedon päivittyminen tilaukset -lomakkeelle	33
Kuva 22. Lasku-lomakkeen tulostustila	36
Kuva 23. Kirjautumisyritys väärällä salasanalla	38
Kuva 24. VBA-ohjelmistolauseiden muokkausyritys	39

# 1 JOHDANTO

Tässä toimeksiannossa on tarkoitus luoda uusi ja täysin asiakkaan toivomusten mukainen reskontrajärjestelmä MS-Access 2007 -ohjelmistolla muuttofirma Ykkösmiehet Oy:lle. Kyseinen ohjelmisto valittiin asiakkaan toivomuksesta, sillä ohjelmisto on pienten ja keskisuurten yritysten keskuudessa tunnettu ja näin ollen yritykselle entuudestaan tuttu. Toimeksiannon toteutusta tuki myös se, että koulutusohjelmamme painoarvo sähköisiä liiketoimintajärjestelmiä kohtaan on ollut merkittävä, joten aiheina tietokantojen suunnittelu, toteutus ja testaus MS-Access 2007:lla oli tietotaitojamme ajatellen varsin luonnollinen valinta.

Tiedonhankinnassa valitsimme lähteiksi kirjallisuuden osalta esimerkiksi seuraavat kirjat: *Microsoft Access 2007 Step by Step*, jonka on kirjoittanut Steve Lambert, M. Dow Lambert III ja Joan Preppernau, sekä *Tietokantojen suunnittelu & indeksointi*, Ari Hovi, Jouni Huotari ja Tapio Lahdenmäki. Sähköisinä tiedonhankintamenetelminä hyödynsimme Wikipedian ja Google scholar:n ohella muun muassa maailman suurinta sosiaalista mediaa, Facebookia. Tietoja kerättiin myös sovituin asiakastapaamisin kontaktihenkilön kanssa, jotka toteutuivat seuraavin aikatauluin:

1. 3.3.2010: Toimeksianto, sovelluksen pääpiirteet
2. 18.3.2010: Toimeksiantosopimukset, kehityskeskustelua
3. 4.4.2010: Ensimmäisen version esittely, muutosehdotukset
4. 7.10.2010: Toisen version esittely, muutosehdotukset
5. 13.12.2010: Sovelluksen systeemitestausvaihe
6. 29.12.2010: Sovelluksen luovutus, käyttöönotto, keskustelua ylläpidosta

Ohjelmistomme tuotantoprosessin vaihejakomallina hyödynnämme sovellettua vesiputousmallia, sekä ohjelmistotuotannon puolelta nopean kehityksen mallia (RAD), kuten Steve McConnel kirjoittaa kirjassaan *Rapid Development: Taming Wild Software Schedules:* ” Kaikki ohjelmistosuunnittelumallit yhtenevät joiltain osin vesiputousmalliin, mutta useimmat suosituista uusista suunnittelumalleista pyrkivät tarjoamaan enemmän joustavuutta suunnitteluprosessiin, jolloin jokaista vaihetta ei tarvitsisi pystyä toteuttamaan täydellisesti ennen seuraavaan siirtymistä ja joissa muutoksen johonkin aiempaan vaiheeseen eivät aiheuttaisi yhtä merkittäviä ongelmia kuin vesiputousmallissa.” Yllä mainitussa vaihejakomallissa suunnittelu- ja toteutusprosessit etenevät vaihe vaiheelta alaspäin, kuten vesiputouksessa, mutta nopean kehityksen menetelmälle ominaisella aikataulusidonnaisella ja tuotantopainotteisella tavalla.

Alapuolelle on esitelty ohjelmistomme vaihejako niin, että prosessi on loogisesti toteutettu ja vaihejako todenmukaisesti etenevä:

- Esitutkimus
- Määrittely
- Suunnittelu
- Toteutus
- Testaus
- Asennus
- Ylläpito

(Oulun seudun ammattiopisto, 2010)

## 2 TAUSTATIEDOT

### 2.1 Toimeksianto

Opinnäytetyömme teimme toimeksiantona muuttofirma Ykkösmiehet Oy:lle johtuen Aleksi Mäkeläisen aiemmasta IT-alan työhistoriasta kyseisessä yrityksessä. Keskusteltaessa yrityksen edustajan kanssa kävi ilmi, että tarve uudelle laskutusjärjestelmälle olisi suuri. Kattavan pohjakoulutuksemme sekä henkilökohtaisen mielenkiintomme perusteella katsoimme toimeksiannon olevan tarpeeksi mielenkiintoinen, haasteellinen ja täysin sopiva alamme opinnäytetyöaiheeksi. Näin ollen päätimme ryhtyä toteuttamaan uutta sovellusta MS-Access 2007 -ohjelmistolla.

Toimeksianto toteutetaan MS-Access 2007 -ohjelmistolla asiakkaan toiveiden mukaisesti. Asiakkaan tarpeita kartoittaessamme kävi ilmi, että paras ohjelma kyseisen sovelluksen toteutukselle olisi yllä mainittu ohjelma, koska asiakkaalla on jo voimassaolevat lisenssioikeudet kyseiseen ohjelmistoon. Valintaa tuki myös se, että ohjelmiston käyttökoulutusajan ollessa niukka, MS-Access 2007 -ohjelmisto tarjoaa sekä helppokäyttöisen että selkeän mahdollisuuden toteuttaa ammattimaista ja yhtenäistä laskutusjärjestelmää.

Toimeksiannon tiedonkeruumenetelmänä olemme pääsääntöisesti käyttäneet asiakastapaamisia. Kyseisellä menetelmällä olemme kartoittaneet muun muassa asiakkaan tarpeet, vaatimukset, aikataulut sekä muut halutut yksityiskohdat tulevaa toimeksiantoa ajatellen.

Toteutus on palkaton ja se toteutetaan osana Turun ammattikorkeakoulun tietojenkäsittelyn koulutusohjelmaa.

Projektin aloitusaikatauluksi sovittiin kevät 2010. Toteutus tulisi aikatauluttaa niin, että sovellus olisi testattu, asennettu ja käyttövalmis jouluna 2010.



## 2.2 Asiakasyritys

Asiakasyrityksenä toimeksiannossa toimii Ykkösmiehet Oy, joka perustettiin vuonna 2003. Yritys on työnkuvaltaan muuttofirma, jonka erikoisalana on piano- ja kassakaappikuljetukset. Toimipiste sijaitsee Liedossa, jossa työskentelee kolme vakituista toimihenkilöä sekä tarvittaessa enemmän tilausten määrästä riippuen. Aktiivialueena toimii koko Suomi, sekä muut maat sopimuksen mukaan. Yritysmuotona on osakeyhtiö.

Yrityksen asiakasrekisteri ja laskutusjärjestelmä on aikaisemmin hoidettu Arkhimedes -nimisellä sovelluksella. Sovellus vastasi yrityksen silloista liiketoimintaa hyvin ja mahdollisti asiakastietojen ja laskutusasioiden esimerkillisen hoitamisen. Liiketoiminnan kasvaessa ja helppokäyttöisyyttä etsittäessä heräsi ajatus yksinkertaisemmasta sekä paremmin heidän nykyisiä tarpeita palvelevasta sovelluksesta. Aikaisempi sovellus osoittautui yksinkertaisesti liian laajaksi ja monimutkaiseksi nykyisiä ja tulevia käyttötarpeita ajatellen. Lisäksi ajatusta sovellusvaihdosta vei eteenpäin Arkhimedes -sovelluksen maksullinen tuotetuki-, ylläpito- ja käyttöönottopalvelut.

Tietomäärä, joka yrityksessä liikkuu, voidaan luokitella pieneksi. Työntekijät, tilaukset, palvelut ja asiakasmäärät ovat täten helposti yhdistettävissä tulevan sovelluksen tietokantaan.



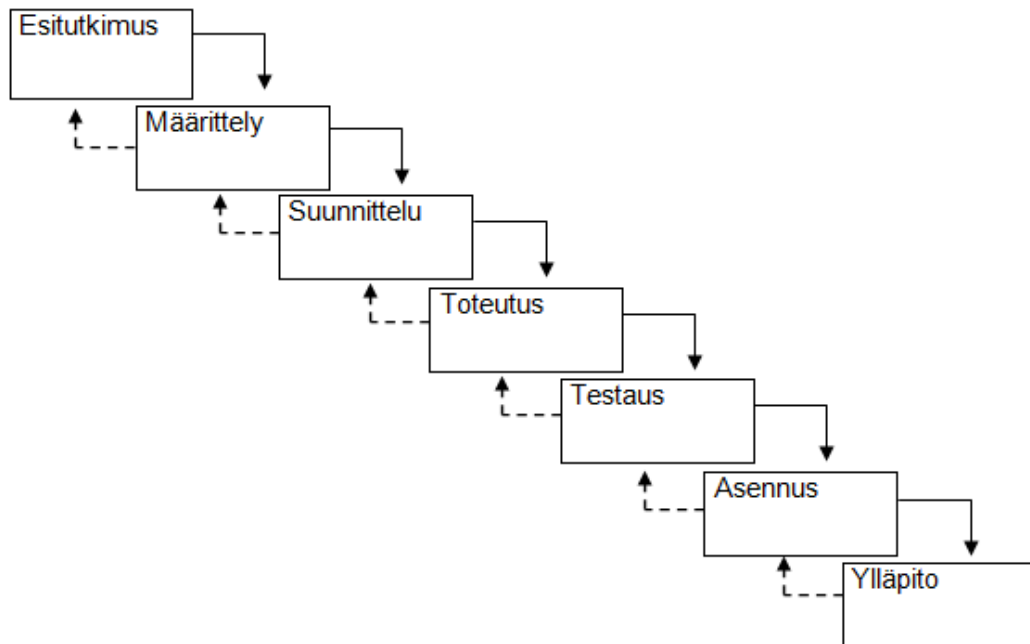
Kuva 1. Ykkösmiehet Oy –logo

### 3 OHJELMISTOTUOTANTO JA -SUUNNITTELU

#### 3.1 Sovelluksen vaihejako- ja elinkaarimalli

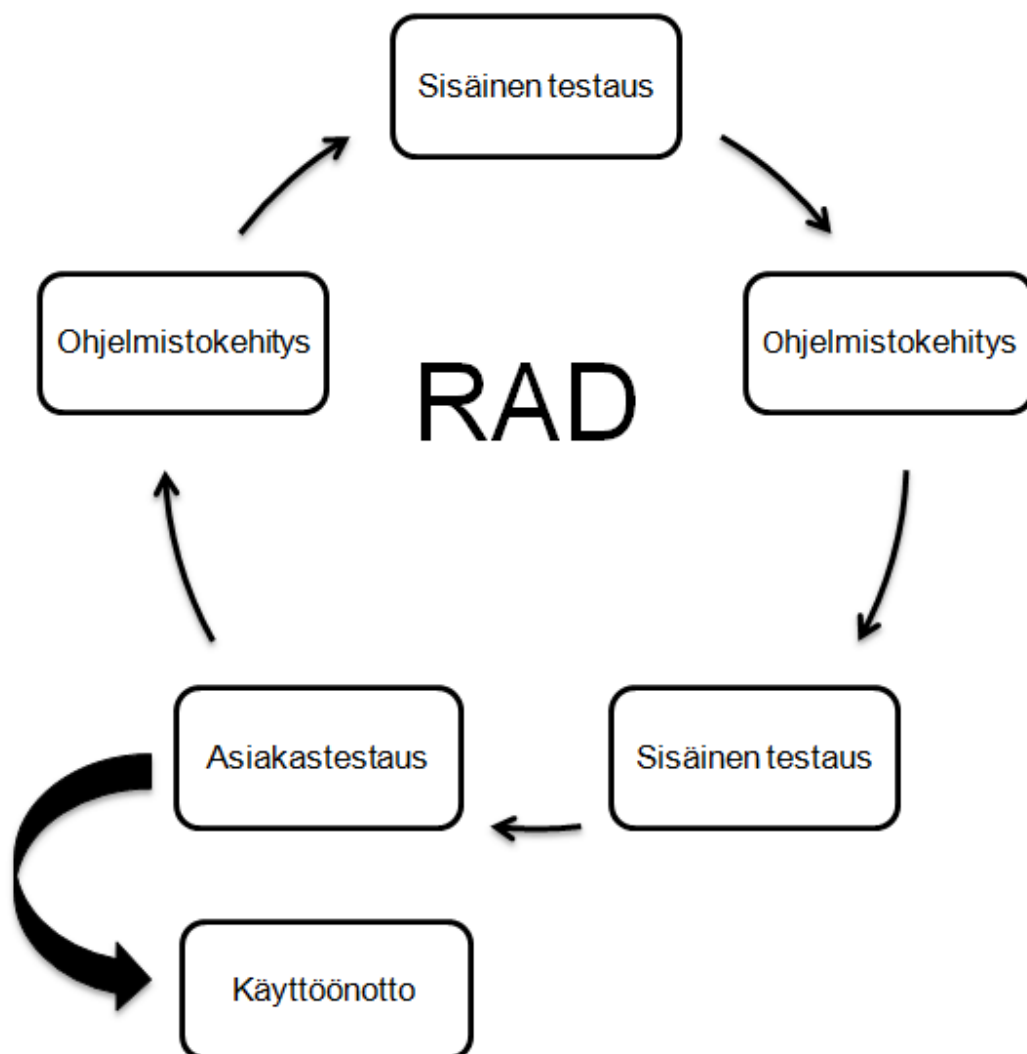
Ohjelmiston elinkaarimallina (*life cycle model*) käsitetään aika, joka kuluu sovelluksen kehittämistyön alkupisteestä sen poistamiseen järjestelmästä. Eri vaihejakomallit käsitetään tapana jakaa sovelluksen tuotanto useisiin eri vaiheisiin (Haikala & Märijärvi 2004, 36). Reskontrasovelluksen kehittämistyössä käytimme nopean kehityksen menetelmällä (*RAD*) sovellettua vesiputousmallia (*waterfall model*). Vesiputousmallia voidaan pitää yleisimpänä ohjelmistotuotannon toteutusmenetelmänä. Muita käytössä olevia ohjelmistotuotantomalleja ovat muun muassa spiraalimalli (*the spiral model*) ja protoilumalleista esimerkiksi RPDM (*rapid prototype design model*).

Kuvassa 2. on esitelty yksi versio vesiputousmallista (*waterfall model*). Vaihejako on alaspäin etenevä. Jokainen vaiheista hoidetaan ns. kerralla kuntoon niin, että vanhempiin vaiheisiin ei enää tarvitse palata. Usein iterointi, eli ajoittainen palaaminen aikaisempaan vaiheeseen on kuitenkin välttämätöntä. Ohjelmistotuotannon kehityskaari etenee loogisesti ylhäältä alas (Haikala & Märijärvi 2004, 36). Katkoviivoilla hahmotetaan nopean kehityksen menetelmälle (*RAD*) tyypillistä testaustoimintaa ja iterointia projektin eri vaiheissa, joka mahdollistaa vesiputousmallille vastakkaisen paluun toteutussuunnitelmassa taaksepäin ilman suuria haasteita.



Kuva 2. Ohjelmiston vaihejakona sovellettu vesiputousmalli

Kuvassa 3. on havainnollistettu sovellettua nopean kehityksen mallia (*RAD*), jota hyödynnämme toimeksiannossamme. Kyseinen sovellettu ohjelmiston kehitystapa korostaa nykyhetkeä niin, että kauaskantoisia suunnitelmia ei tehdä, vaan sovellus on jatkuvan muutoksen alaisena ja suunnittelutyö tapahtuu osittain projektin edetessä. Kehitystapa mahdollistaa sovelluksen muutokset kaikissa eri vaiheissa. Kehitystyö jatkuu kuvan mukaisella tavalla niin kauan, kunnes toimeksiantaja hyväksyy sovelluksen, jonka jälkeen sovellus on käyttöönottovalmis. Kokonaisuus etenee kuitenkin vesiputousmallille (*waterfall model*) ominaisella alaspäinetenevällä, porrastetulla toteutustavalla niin, että jokainen erillinen toteutusvaihe tai koko vaihejako voi toistaa itseään iteroiden kuvalla 3. ominaisella tavalla.



Kuva 3. Sovellettu nopean kehityksen malli (RAD)

### 3.2 Ohjelmistosuunnittelu

”Ohjelmistosuunnittelu koostuu kahdesta vaiheesta. Näistä ensimmäinen on toiminnallinen määrittely. Siinä kuvataan kaikki järjestelmän toteuttamat toiminnot ja liitännät järjestelmän ulkopuolelle. Vaiheen tuotoksena syntyvä *määrittelydokumentti* kuvaa siis mitä kaikkea järjestelmällä voi tehdä sekä miten *käyttäjät* voi ne tehdä. Järjestelmä ei kuvaa sitä, miten toiminnot tulee *toteuttaa*. Näin määrittelydokumentti kuvaa esimerkiksi tekstinkäsittelyohjelman eri näytöt, valikot ja niissä olevat asetukset ja kaikki käyttöliittymäkontrollit. Mikäli

järjestelmä tuottaa tulosteita, niin määrittelydokumentin tulisi sisältää sekä visuaaliset että riittävät tekstuaaliset kuvaukset kaikista järjestelmän eri näytöistä.

Ideaalinen määrittelydokumentti on niin kattava, että teknisessä suunnittelussa tai sitä seuraavissa vaiheissa ei ole enää missään vaiheessa epäselvää, että miten ohjelman tulee toimia kussakin tilanteessa, miten tulosteet ja syöte on muotoiltu jne. Vaikka tähän on käytännössä mahdotonta päästä, niin on kuitenkin havaittu, että siihen tulee pyrkiä.” (Wikipedia, 2010.)

Tietokannan suunnittelu toteutettiin seuraavin eri päävaihein:

1. Tietokannan tehtävien määrittäminen
2. Haluttujen tietojen kerääminen ja järjestäminen
3. Tietojen organisointi taulukoiksi
4. Kerättyjen tietojen muuntaminen sarakkeiksi
5. Pääavainten määrittäminen
6. Taulujen välisten yhteyksien luonti
7. Rakenteen analysointi ja testaus

### 3.3 Sovelluksen määrittely ja vaatimuskartoitus

Tämän näyttötyön tavoite on luoda reskontra-sovellus MS-Access 2007 – ohjelmistolla. Tulevan sovelluksen aluerajaus on toteutettu niin, että ohjelma keskittyy luomaan, muokkaamaan, tallentamaan, tulostamaan ja näyttämään halutut asiakas-, palvelu-, tilaus- ja laskutustiedot käyttöliittymässä, joka on luotu asiakasyrityksen määrittelemien vaatimusten mukaisesti.

Käyttötapausten ensisijaisena tarkoituksena voidaan pitää toimintaa kommunikointivälineenä kartoitettaessa asiakkaiden vaatimuksia ja verrattaessa niitä ohjelmistovaatimuksiin.

Käyttötapauksia voidaan havainnollistaa ainakin kahdella tavalla: Tekstimuotoisina tai use-case -kaavioilla (*use-case-diagram*). Lisäksi käyttötapauksilla pyritään yleisesti havainnollistamaan muun muassa tiettyä sovelluksen toimintaa tai sen osaa, luotua tapahtumaa tai vuorovaikutusta käyttäjän ja sovelluksen välillä. Käyttötapaukseksi voidaan määritellä laajempi kokonaisuus, tai jokin tietty haluttu yksityiskohta (Haikala & Märijärvi 2004, 158). Alla olevat käyttötapaukset on esitetty use-case -kaavioilla ja ne kuvaavat sovelluksen toimintoja sen eri käyttötapauksissa.

<b>Käyttötapaus 1: Tietojen hallinto: Tilauksen lisääminen asiakkaalle</b>	
<b>Käyttäjät:</b>	Työntekijät
<b>Yleisyys:</b>	Aina asiakkaan tehdessä uuden tilauksen
<b>Esiehdot:</b>	Asiakastiedot löytyvät tietokannasta
<b>Kuvaus:</b>	Suoritetaan tilauksen kirjaaminen tietokantaan valitulle asiakkaalle
<b>Lopputulos:</b>	Asiakkaan tilaus on kirjattu tietokantaan
<b>Poikkeukset:</b>	P1. Uusi tilaus on aina lisättävä tietokantaan uusi-toiminnon avulla Tilaukset-lomakkeella. Muuten järjestelmä päivittää vanhan tilauksen uuden tilauksen tiedoilla. Tällöin tilausten lukumäärä ei muutu.

Kuva 4. Käyttötapaus tilauksen lisäämisestä asiakkaalle

<b>Käyttötapaus 2: Tietojen hallinointi: Asiakastietojen päivitys</b>	
<b>Käyttäjät:</b>	Työntekijät
<b>Yleisyys:</b>	Jo olemassa olevien asiakastietojen muuttuessa
<b>Esiehdot:</b>	Asiakastiedot löytyvät tietokannasta
<b>Kuvaus:</b>	Kirjataan muuttuneet asiakastiedot tietokantaan
<b>Lopputulos:</b>	Valitun asiakkaan asiakastiedot ovat päivittyneet ajan tasalle
<b>Poikkeukset:</b>	P1. Päivitettävien tietojen oltava yhteneviä lomakkeen kenttien tietotyyppien kanssa

Kuva 5. Käyttötapaus asiakastietojen päivityksestä

<b>Käyttötapaus 3: Tietojen hallinointi: Tilauksen laskuttaminen</b>	
<b>Käyttäjät:</b>	Työntekijät
<b>Yleisyys:</b>	Aina toteutuneen tilauksen jälkeen
<b>Esiehdot:</b>	Asiakas-, palvelu-, ja tilaustietojen löydyttävä tietokannasta
<b>Kuvaus:</b>	Luodaan asiakkaan tilaukselle lasku laskuta-toimintoa käyttäen tilaukset -lomakkeelta.
<b>Lopputulos:</b>	Tilauksen lasku tallentuu tietokantaan ja lasku on toimitettavissa asiakkaalle
<b>Poikkeukset:</b>	P1. Tilaukset-lomakkeen pakolliset kentät oltava täydennettyinä oikeaoppisesti. Laskutus tapahtuu suoraan tilaukset-lomakkeen tietojen kautta.

Kuva 6. Käyttötapaus tilauksen laskuttamisesta

Asiakastapaamisten perusteella sovellukselta vaadittiin ainakin seuraavia toimintoja ja ominaisuuksia:

- Sovellus ei vaadi tietokoneelta suuria resursseja
- Internetyhteys ei ole välttämätön
- MS-Access 2007
- Ei ylläpito-, tuotetuki- tai käyttökustannuksia
- Selkeä, helppokäyttöinen ja yhtenäinen käyttöliittymä
- Reaaliaikainen asiakas-, laskutus-, ja palvelutietokanta
- Mahdollisuus yksittäisten tietojen hallinointiin
- Selkeä vuosikatsaus tilauksista



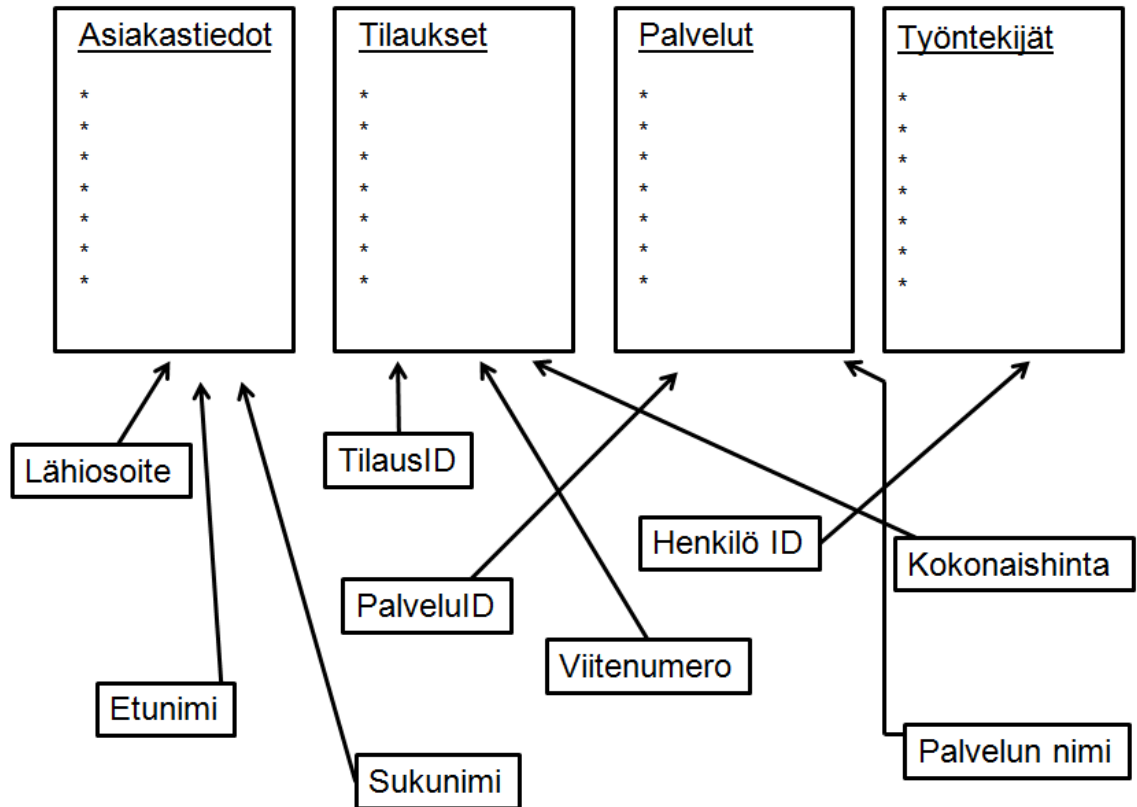
- Lista asiakkaista, laskuista palveluista ja tilauksista
- Sovelluksen turvallisuus (kirjautuminen salasanalla)
- Asiakas-, tilaus- ja laskutushistoria oltava aina saatavissa
- Asiakas- ja tilaushaku oltava olemassa
- Sovelluksessa pystyttävä päivittämään tietoja niin haluttaessa (asiakastiedot, laskutustiedot, palvelutiedot, tilaustiedot)
- Yhteystietojen hallinta
- Palveluiden esittely
- Sovellusta voitava kehittää
- Sovelluksen oltava vakaa ja tietoturvallinen

### 3.4 Tietojen etsiminen ja järjestäminen

Yleensä etsimis- ja järjestämisvaihe aloitetaan vanhojen tietojen kartoituksella, muun muassa vanhojen asiakastietojen säilyttämiseksi. Kerätyistä tiedoista saadaan tarvittavat materiaalit sovelluksen taulukoiden sarakkeiden nimeämiseen sekä esimerkiksi raporttien suunnitteluun. Järjestäminen on tärkeä vaihe, sillä looginen tietojen järjestäminen alussa helpottaa tietojen jäsentelyä, käytettävyyttä ja tulevia ohjelmistosuunnittelun vaiheita ([www.office.microsoft.com](http://www.office.microsoft.com)).

Kyseinen vaihe tässä näyttötyössä suoritettiin kuvitteellisten testitietojen pohjalta, koska yrityksen asiakastiedot ovat luottamuksellisia. Sovelluksen taulukoiden sarakkeet nimettiin asiakashaastatteluista kerätyillä tiedoilla. Haastattelutilanteessa tuotiin esille sovelluksen toimivuuden kannalta oleelliset vaatimukset taulukoiden sarakkeiden nimeämiseksi ja tuotettavien toimintojen toteuttamiseksi.

Kuvasta 7. käy ilmi looginen etsittyjen tietojen järjestäminen LexTop –reskontrasovelluksen suunnitteluvaiheessa.



Kuva 7. Tietojen etsiminen ja järjestäminen

### 3.5 Tietojen jakaminen taulukoihin

Kerätyt ja järjestetyt tiedot jaetaan seuraavaksi taulukoihin niin, että kukin taulu muodostaa selkeän pääkokonaisuuden. Aikaisemman tietojen järjestelyn tuloksena taulut suunniteltiin niin, että jokainen sarakkeen nimike esiintyy kaikissa tauluissa vain kerran tulevien päivitysten helpottamiseksi.

Kuvan 8. kerätyt tiedot on järjestelty sarakkeiksi ja ryhmitelty pääkokonaisuuden mukaan omaksi asiakastiedot-tauluksi.

<u>Asiakastiedot</u>
- #Asiakas ID
- Etunimi
- Sukunimi
- Lähiosoite
- Postinumero
- Kaupunki
- Maa
- Puhelinnumero
- Sähköposti

Kuva 8. Asiakastiedot-taulu

### 3.6 Tieto-osien muuntaminen sarakkeiksi

Tulevan tietokannan erilliset tieto-osat kirjataan sarakkeittain niin, että jokainen kirjattu tieto muodostaa taulussa oman kentän. Kuvassa 9. on esitetty reskontrasovelluksen asiakastiedot-tauluun sarakkeittain lisätyt tieto-osat, kuten esimerkiksi *asiakas ID*, *etunimi* ja *lähiosoite*.

Sarakkeiden kirjaamisvaiheessa määritetään tulevien taulujen sisältämät kentät. Tässä vaiheessa syötetään jokaisen tieto-osan tietotyyppi siten, että esimerkiksi tieto-osa ”*asiakas ID*” sisältää vain numeerista (*number*) tietoa ja ”*etunimi*” vain tekstuaalista (*text*) tietoa. Tietotyyppivalinnat on syytä tehdä harkiten, sillä se muun muassa helpottaa järjestelmän käytettävyyttä tulevaisuudessa siten, että esimerkiksi numeraalisen tiedon syöttäminen kenttään ”*etunimi*” ei ole mahdollista. Järjestelmä ilmoittaa lisäysvirheestä, jos tietotyyppi kentässä ei täsmää.

The screenshot shows the Microsoft Access interface with a table named 'Asiakastiedot'. The table has the following columns: Asiakas ID, Etunimi, Sukunimi, Lähiosoite, Postinumero, Kaupunki, Maa, Puhelinnumero, and Sähköposti. The data is as follows:

Asiakas ID	Etunimi	Sukunimi	Lähiosoite	Postinumero	Kaupunki	Maa	Puhelinnumero	Sähköposti
1	Aleksi	Mäkeläinen	Kuninkaantie	03600	Kuusamo	Venäjä	066545643	Allu@google.c
2	Topias	Laaksonen	Lordikatu 6	89423	Turku	Kazakistan	4352534	Topsu@tl.com
3	Arke	Pera	Kiikarikuja 7 n	50441	Kaarina	Hawaji	545451541	dsfsffs@kille.c
4	Marko	Juudaskuja	Katukuja 6	20449	Nyttisaari	Kaukasia	32592350	Arnold@donit
5	Kalle	Kunnari	Karikatu 6 h	83488	Zerokingel	Latvia	9867876	kunnari@base
6	Merja	Nahkakuja	Kirveskatu 89	53985	Limsa	Ruotsi	438529853	Laku@petteri.
7	Markku	Meikäläinen	jeiajie 89	99098	City	Skanssi	050897738	markkum@jipi
8	keke	randelin	ranekatu 8	92300	Kiina	Shanghai	0405554343	
9	Eino	Harju	Harjutie 666	93600	Kuusamo	Suomi	0700111111	Eikka@Kemi.fi
10	Kalle	Kekkele	Kunataateli 2	91820	Kehveli	Suomi	0405151515	
11	Hannu	Hanhi	Hanetie 98	12200	Veteli	Suomi	0501244594	
12	Nakke	Nakuttaja	Haapamäki 6 T	26110	Haaparanta	Suomi	-	nakken@disne
13	Mikko	Majava	Närhipolku 6	20399	Makedonia	Israel	+35489281924	Mikko@Makec

Kuva 9. Asiakastaulun tieto-osat sarakkeina

Kuvassa 10. on esitetty kirjattujen kenttien tilaukset-taulun tietotyyppivalinnat LexTop-reskontrasovelluksessa.

The screenshot shows the 'Tilaukset' table structure with the following fields and data types:

Field Name	Data Type
Tilaus ID	AutoNumber
Palvelu ID	Number
Kokonaishinta	Currency
Lisätiedot	Text
Asiakas ID	Number
Päivämäärä	Date/Time
Vastaanottaja	Text
Maksettu	Text
Laskun päivämäärä	Date/Time
Viitenumero	Number
Henkilö ID	Text

Kuva 10. Tilaukset-taulun tietotyyppivalinnat

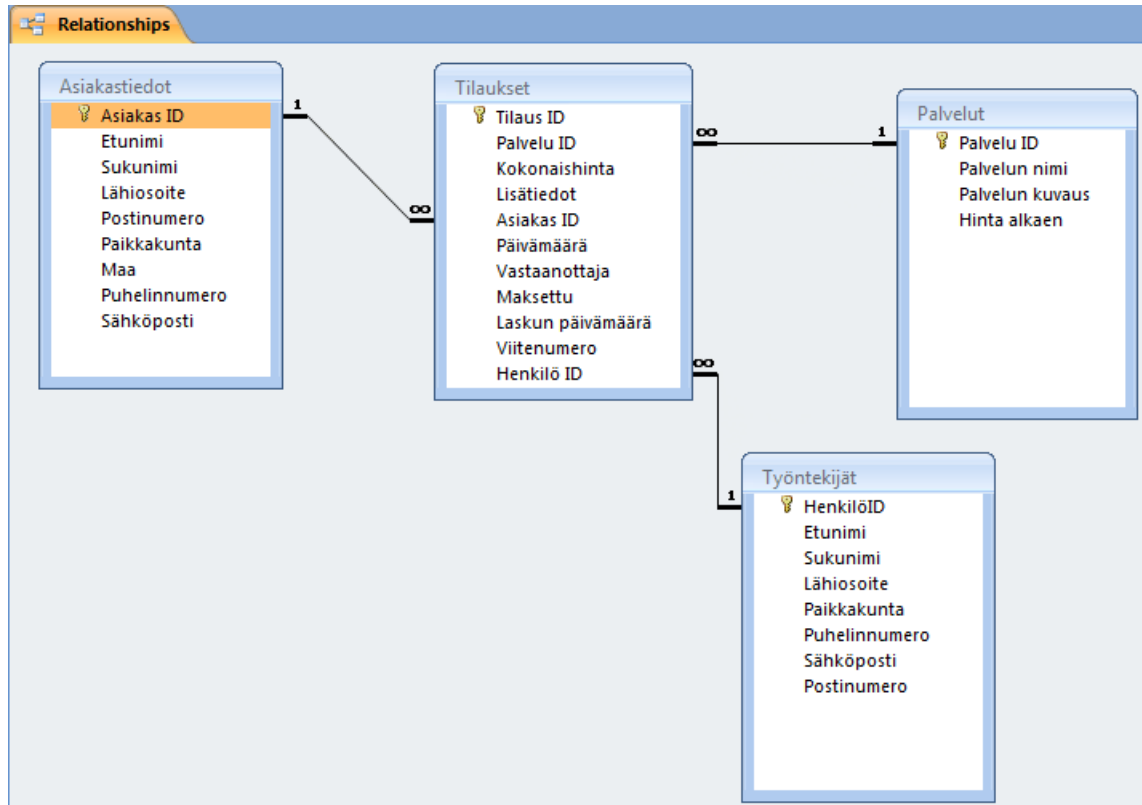
### 3.7 Pääavainten määrittäminen

Pääavainten (*primary key*) määrittäminen toteutetaan sarakkeiden tietojen kirjaamisen jälkeen. Jokaisen taulun sarakkeisto sisältää vähintään yhden tunnistetekijän, eli pääavaimen, joka yksilöi taulun rivit. Yleensä tunnistetietona, eli pääavaimena (*primary key*) käytetään automaattista, numeerista tietoa, kuten esimerkiksi ”*tilaus ID*” tai ”*asiakas ID*”. Autonumeerisen (*autonumber*) tiedon päällekkäisyysmahdollisuudet ovat käytännössä mahdottomat, sillä järjestelmä luo jokaiselle lisätylle tiedolle oman, yksilöllisen tunnistetiedon, jolloin kahdella tietueella ei voi olla samaa arvoa valitussa kentässä. Rivien yksilöinti on parasta toteuttaa autonumeerisella (*autonumber*) tietotyypillä, kuten kuvasta 10. voidaan todeta. Kuvassa ”*tilaus ID*” on pääavain ja tietotyypinä ”*autonumber*”.

### 3.8 Taulukoiden välisten yhteyksien luonti

Tässä vaiheessa tarkastellaan jokaista taulua yksilöllisesti niin, että pystytään hahmottamaan taulukoiden väliset yhteydet. Pyritään hahmottamaan kokonaisuutta siten, että ymmärretään miten yhden taulukon tiedot liittyvät muiden taulukoiden tietoihin. Uusien kenttien tai taulukoiden luominen on kyseisessä vaiheessa suositeltavaa, jos selkeitä yhteyksiä taulujen välille ei löydy.

Tässä reskontrasovelluksessa luotiin taulujen väliset yhteydet jokaisen taulun välille niin, että tiedonkulku taulujen välillä toimii halutulla tavalla. Esimerkiksi yhteydet *#asiakas\_ID* → *asiakas\_ID* ja *palvelu\_ID* → *#palvelu\_ID* toteutettiin, koska asiakkailta on tilauksia ja jokainen tilaus sisältää palvelun. Luotujen yhteyksien pohjalta rakennettiin sovellus, jossa tilaukset on sidottu asiakkaisiin ja palvelut tilauksiin. Kuvassa 11. on havainnollistettu LexTop-reskontrasovelluksen taulukoiden väliset yhteydet relaatiotietokannassa.

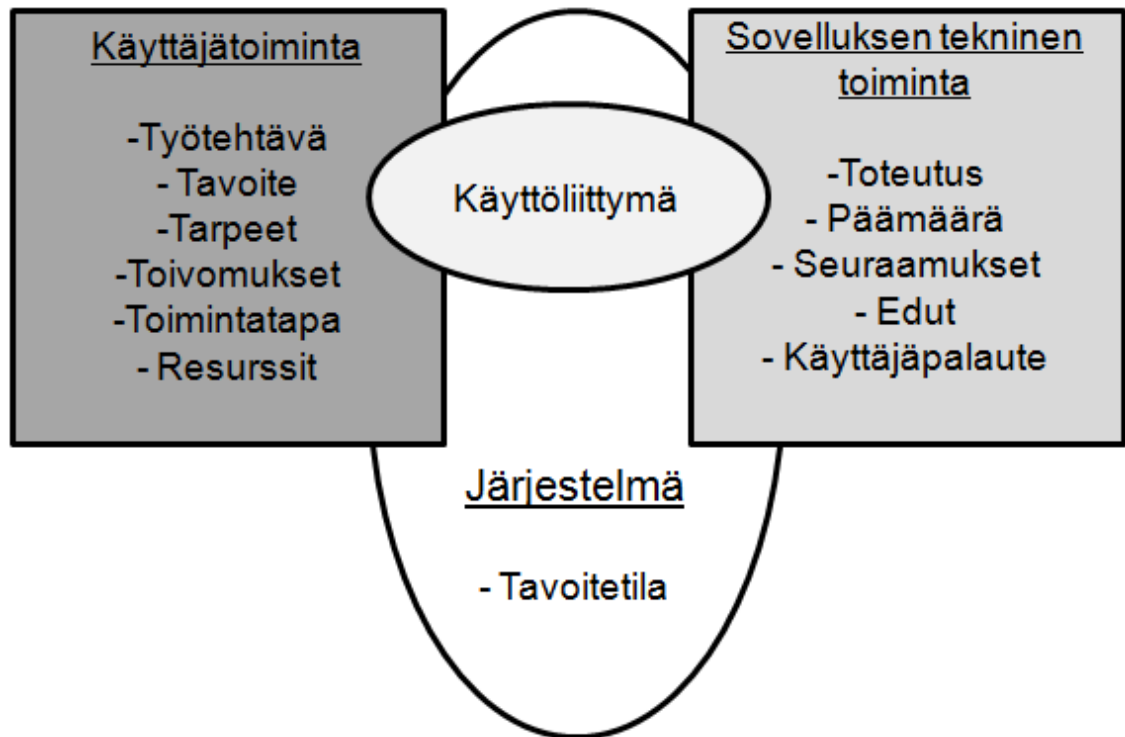


Kuva 11. Taulukoiden väliset yhteydet relaatiotietokannassa

### 3.9 Käyttöliittymäsuunnittelu

Käyttöliittymällä tarkoitetaan jonkin ohjelman visuaalista työkenttää. Työkenttä on nimensä mukaisesti ainoa käyttäjälle näkyvässä oleva kenttä ja siten onnistuneen käyttöliittymäratkaisun tulee olla helppokäyttöinen, selkeä, opastava ja ns. itsestään selvä - mahdollisuutta virhetilanteisiin ei pitäisi olla loogisen käyttöliittymäsuunnittelun tuloksena ([www.vtt.fi](http://www.vtt.fi)). Kuvassa 12. on havainnollistettu käyttöliittymäsuunnittelun peruserä ja lähtökohta, eli käyttäjäkokemus ja -toiminta. Alla olevaa käyttöliittymäsuunnittelun kaavio etenee seuraavassa järjestyksessä:

Käyttäjätoiminta → sovelluksen tekninen toiminta → käyttöliittymä



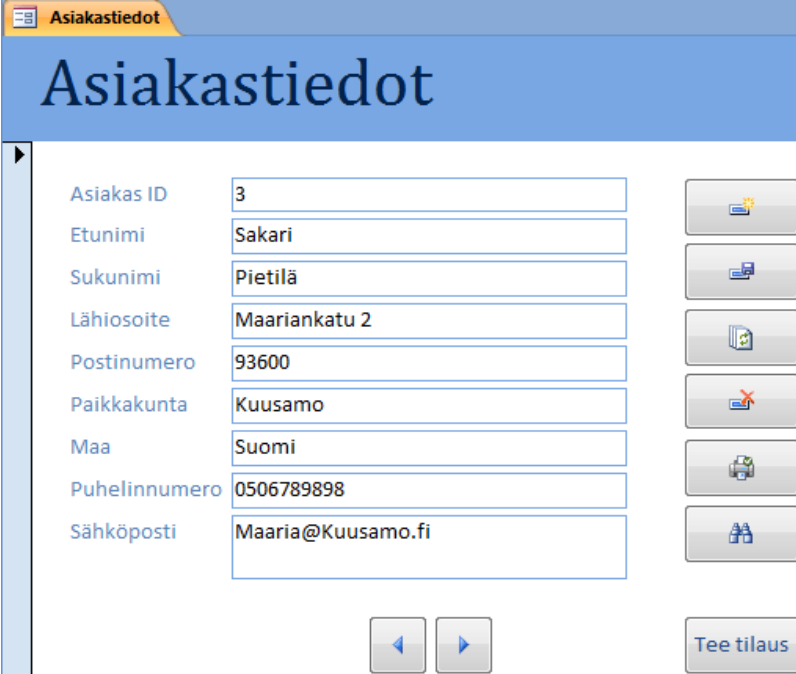
Kuva 12. Käyttöliittymäsuunnittelu käyttäjätöiminnan pohjalta

Sovelluksen käyttäjä toteuttaa luodussa käyttöliittymässä kaikkia sovelluksen tietohallinnan tehtäviä, joilla tarkoitetaan muun muassa tietojen poistoa, -päivitystä, -lisäämistä ja esimerkiksi tietojen eri hakutoimintoja.

Esimerkillisen käyttöliittymäsuunnittelun lähtökohtana voidaan pitää käyttäjän tarpeiden ja toiminnan tarkastelua. Tarkemmin sanottuna sovelluksen toiminta ja sen informaatiokäsittely ovat käyttöliittymäsuunnittelun keskeisimpiä kulmakiviä. Suunnitteluvaiheessa huomioidaan esimerkiksi käyttöliittymän toimintojen tulevat mahdolliset muokkaukset, tulevien toiminnallisten muutosten, sekä virheiden korjaamisen helpottamiseksi ja virheensietokyvyn (*fault tolerance*) maksimoimiseksi. Looginen ajattelu ja –käyttöliittymätoteutus ovat avainasemassa toimivaa käyttöliittymää suunniteltaessa. Viime kädessä kuitenkin käyttäjän käyttötehtävät ja –olosuhteet määrittelevät luodun käyttöliittymän laadun. Ohjelmistosuunnittelijan ajattelumalli eroaa usein poikkeuksetta käyttäjien ajattelumallista. Eroavaisuus korostaa

käyttäjätarpeiden kartoitusta entisestään ja muodostaa yhden suurimmista haasteista käyttöliittymäsuunnittelussa (www.vtt.fi).

Tässä työssä suunnitellun reskontrasovelluksen käyttöliittymä luotiin toimeksiantajan käyttötarpeiden pohjalta. Kuvasta 13 ja 14 käy ilmi käyttöliittymän loogisen toimintatavan lisäksi yritykselle ominainen graafinen ulkoasu. Huomioituja käyttötarpeita ja haluttuja toimintoja lueteltiin muun muassa kappaleessa 3.3, jossa käytiin läpi sovelluksen määrittelyä ja vaatimuskartoitusta.



Asiakastiedot	
Asiakas ID	3
Etunimi	Sakari
Sukunimi	Pietilä
Lähiosoite	Maariankatu 2
Postinumero	93600
Paikkakunta	Kuusamo
Maa	Suomi
Puhelinnumero	0506789898
Sähköposti	Maaria@Kuusamo.fi

Navigation buttons: < >

Action buttons: [Add], [Edit], [Delete], [Cancel], [Print], [Share]

Final button: Tee tilaus

Kuva 13. Asiakastiedot-lomakkeen graafinen ulkoasu



Tilaukset

## Tilaukset

Asiakas ID	Etunimi		Sukunimi	
3	Sakari		Pietilä	
Lähiosoite			Postinumero	
Maariankatu 2			93600	
Paikkakunta		Maa	Puhelinnumero	
Kuusamo		Suomi	0506789898	
Sähköposti				
Maaria@Kuusamo.fi				
Tilaus ID	Vastaanotettu	Vastaanottaja		
28		Kerkko	Kallio	
Palvelu ID	Palvelun nimi			
1	Kassakaappi			
Palvelun kuvaus				
Kassakaapin siirto				
Lisätiedot				
Laskun päivämäärä	Viitenumero	Kokonaishinta	Laskun tila	
		600,00 €	Maksettu	

Asiakas: Pietilä    Palvelu: Kassakaappi    Vastaanottaja: Kallio

←    →

Laskuta

Etusivu

Kuva 14. Tilaukset-lomakkeen graafinen käyttöliittymä

## 4 TESTAUS

### 4.1 Testaussuunnitelma

Ohjelmistotestaus käsitetään suunnitelmallisena virheiden etsintänä sovellusta tai sen eri osia ajettaessa läpi. Testaus on yksi tärkeimmistä ohjelmistosuunnittelun vaiheista, sillä suunnitelmallisen ja huolellisen testauksen tuotoksena saadaan huomattavasti parempia tuloksia, kuin umpimähkäisesti kokeilemalla. Kuvassa 15. on havainnollistettu tämän sovellustestauksen eri suunnitteluvaiheet.

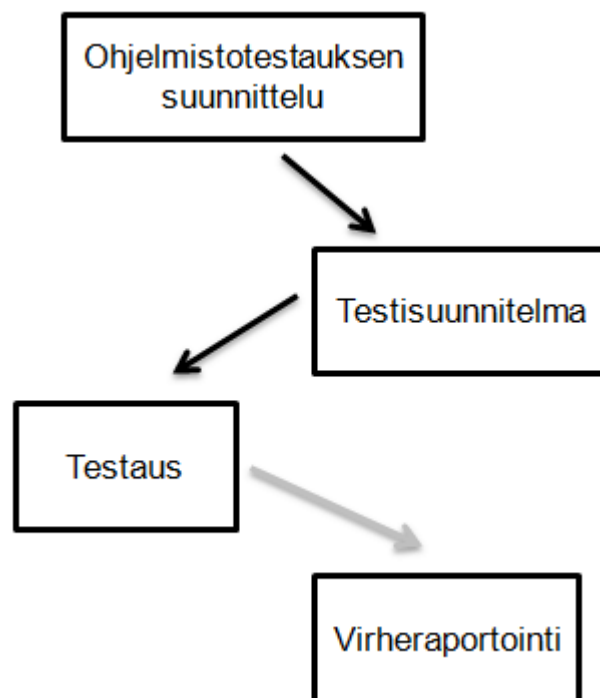
Tämän reskontrasovelluksen testausvaihe on jaettu kahteen kappaleeseen: Integraatiotestaus (*integration testing*) ja systeemitestaus (*system testing*). Integraatiotestauksella tarkoitetaan vaiheita, jossa luodun sovelluksen pienemmät osat yhdistetään suurempiin kokonaisuuksiin ja testataan osien yhteensopivuus ja -toimivuus. Tässä testausvaiheessa ei testata yksittäisten osien sisäistä toimivuutta ([www.mit.jyu.fi](http://www.mit.jyu.fi)). Ihannetilanteessa integraatiotestauksen tuloksena löydetään yhteensopimattomia ohjelmiston eri osia, eli virheitä (*errors, bugs*) joiden tuloksena syntyy ohjelmistotestauksen virheraportti.

Systeemitestauksella (*system testing*) tarkoitetaan valmiin sovelluksen läpikäyntiä niin, että selviää täyttääkö valmis sovellus testaukselle asetetut lopetus- ja määrittelykriteerit. Sovellus altistetaan kokonaan testattavaksi ennen sen asiakkaalle toimittamista. Systeemitestauksen päätavoite on, että järjestelmä vastaa täysin sille annettuja määritteitä. Tämä testausvaihe toteutetaan samankaltaisessa ympäristössä kuin missä sovellusta tullaan todellisuudessa käyttämään ([www.mit.jyu.fi](http://www.mit.jyu.fi)).

Testaukset toteutetaan taulujen; *Asiakastiedot, tilaukset, henkilöstö ja palvelut* välillä niin, että jokaisen taulun sarakkeet, lomakkeet, raportit, tietotyypit ja toiminnot pystytään luokittelemaan oikeaoppisiksi ja toimiviksi, asiakasvaatimusten mukaisesti. Varsinaiset testausvaiheet tehdään tässä

opinnäytetyössä ohjelmistosuunnittelun loppuvaiheessa niin, että kyseistä vaihetta toteutetaan järjestelmällisesti kaksi kokonaista työviikkoa, ajanjaksoilla 8.11.2010 – 12.11.2010 ja 06.12.2010 – 13.12.2010.

Sovellustestauksen lopettamiskriteerit ovat yhtenevät kappaleessa 3.3 mainittuihin sovelluksen määrittely ja vaatimuskartoituksiin. Testaus lopetetaan, kun toimeksiantajan asettamat ohjelmistovaatimukset on toteutettu niin, että sovellus toimii toimeksiantajan määrittelyjen mukaisesti.



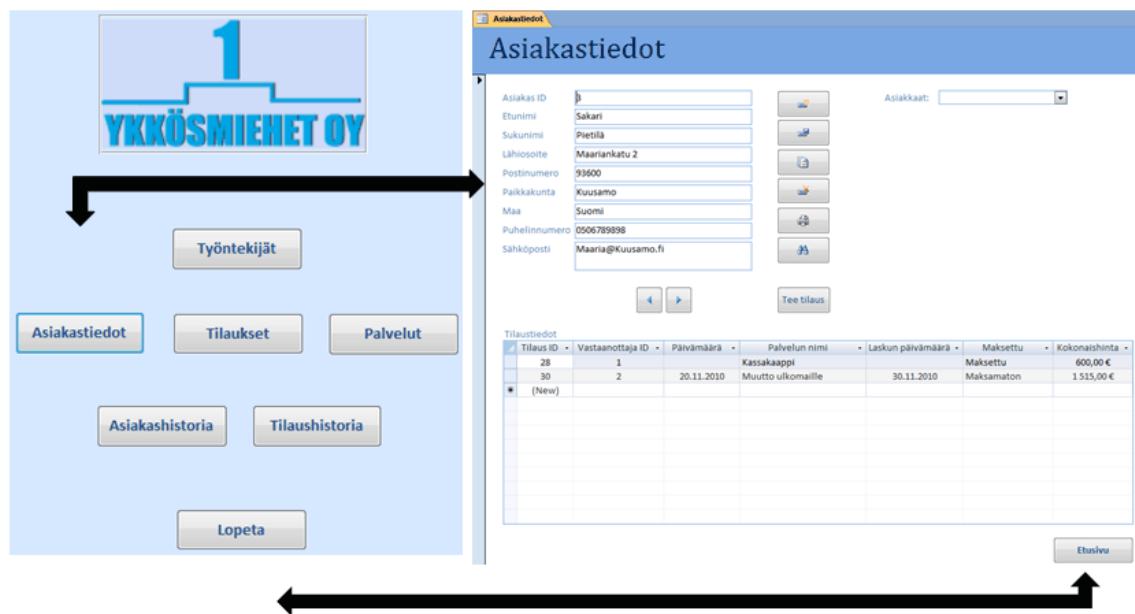
Kuva 15. Ohjelmistotestauksen suunnitteluvaiheet

## 4.2 Integraatiotestaus

Testitapaus 1: *Etusivulomake* ja navigoinnin toimivuus.

Testasimme reskontrasovelluksessa navigointia etusivulta lomakkeille; *Asiakastiedot*, *tilaukset* ja *palvelut*, sekä raporteihin *tilaushistoria* ja *asiakashistoria*. Jokaiselta lomakkeelta ja raportilta täytyi löytyä myös paluutoiminto takaisin *etusivu*-lomakkeelle.

Testituloksena todettiin kaikkien yllämainittujen navigointitoimintojen toimivuus. Navigointi lomakkeiden välillä toimii esimerkiksi lomakkeiden; *etusivu* → *asiakastiedot* → *etusivu* kesken kuvan 16. osoittamalla tavalla.



Kuva 16. Navigointipainikkeiden toimivuus

Testitapaus 2: *Asiakastiedot*-, *tilaukset*-, ja *palvelut*-lomakkeen tiedonhallintatoimintojen toimivuus

Tässä testitapauksessa testattiin eri lomakkeiden ja raporttien toimintojen toimivuutta. Testi suoritettiin kaikkien sovelluksen taulujen ja raporttien osalta niin, että jokaisesta lomakkeesta tai raportista umpimähkään valitut toiminnot täyttivät niille asetetut määrittelykriteerit.

Tuloksena havaittiin kaikkien lomakkeiden ja raporttien toimivuus oikeaoppiseksi. Luodut toiminnot, kuten esimerkiksi uuden tilauksen lisääminen tietokantaan, päivittävät tehdyt muutokset tietokantaan toimintopainikkeiden avulla. Vääräntyyppisen tiedon syöttäminen, kuten tekstuaalisen tiedon syöttö numeerisen tiedon kenttään, antaa oikeaoppisesti virheilmoituksen väärästä tietotyypistä kuvan 18. osoittamalla tavalla *asiakastiedot*-taulussa. Testitapauksessa syötimme numeeriseen *asiakas ID* -kenttään tekstuaalista tietoa.

Kuvassa 17. on esitetty *tilaukset*-lomakkeen ”*luo uusi tilaus*” -toiminnon tapahtuma.

Tilaukset

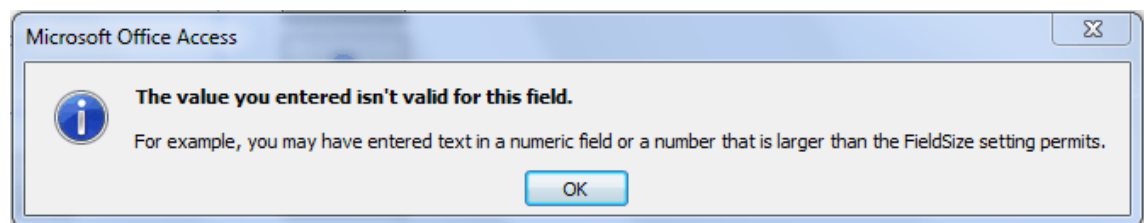
# Tilaukset

Asiakas ID	Etinimi	Sukunimi	
Lähiosoite		Postinumero	
Paikkakunta	Maa	Puhelinnumero	
Sähköposti			
Tilaus ID	Vastaanotettu	Vastaanottaja	
(New)			
Palvelu ID	Palvelun nimi		
Palvelun kuvaus			
Lisätiedot			
Laskun päivämäärä	Viitenumero	Kokonaishinta	Laskun tila

Asiakas:  Palvelu:  Vastaanottaja:

Etusivu

Kuva 17. Uuden tilauksen luominen



Kuva 18. Virheilmoitus väärästä tietotyypistä

Testitapaus 3: *Palvelut*-taulun tietojen päivittyminen *tilaustiedot*-alilomakkeelle ja tilaukset –lomakkeelle.

Palveluiden päivittäminen, poistaminen ja lisääminen tehtiin tässä sovelluksessa niin, että muutokset tehdään suoraan *palvelut* –tauluun käyttöliittymän kautta. Tehdyt muutokset päivittyvät automaattisesti *palvelut* –lomakkeen tietoja käyttäviin lomakkeisiin (*tilaustiedot*, *tilaukset*). *Palvelut* –taulun toimivuus on avainasemassa sovelluksen käyttöä ajatellen, sillä palveluiden määrä muuttuu jatkuvasti ja vanhojen palveluiden tietoja on pystyttävä päivittämään. Tässä testitapauksessa lisäsimme *palvelut* –tauluun kuvitteellisen palvelun ja testasimme syötettyjen tietojen päivittymistä muualle tietokantaan (*tilaustiedot*, *tilaukset*).

Kuvasta 19. käy ilmi *palvelut* –tauluun lisätyn tiedon päivittyminen *tilaustiedot* –alilomakkeelle ja *tilaukset* –lomakkeelle.

The image shows a software interface with two main panels. The left panel is titled 'Tilaukset' and contains a form with fields for customer information, address, and order details. The right panel is titled 'Palvelut' and contains a table with columns: 'Palvelu ID', 'Palvelun nimi', 'Palvelun kuvaus', and 'Hinta alkaen'. Row 7 in this table is highlighted with a red border. Below the 'Palvelut' table is a 'Tilaustiedot' table with columns: 'Tilaus ID', 'Vastaanottaja ID', 'Päivämäärä', 'Palvelun nimi', 'Laskun päivämäärä', 'Maksettu', and 'Kokonaishinta'. Row 1 in the 'Tilaustiedot' table is highlighted. A dropdown menu is open below the 'Palvelut' table, showing a list of services with their prices and IDs. Row 7 in this dropdown is also highlighted. Arrows indicate the flow of data from the 'Palvelut' table to the 'Tilaustiedot' table and to the dropdown menu.

Palvelu ID	Palvelun nimi	Palvelun kuvaus	Hinta alkaen
1	Kassakaappi	Kassakaapin siirto	300,00 €
2	Muutto kotimaassa	Muutto	600,00 €
3	Muuttolaatiko	Laatikoita kansalle	15,00 €
4	Pianot ja flyygelit	Suurien soitinten siirto	200,00 €
5	Muutto ulkomaille	Muutto Suomen rajojen ulkopuolelle	800,00 €
6	Trukkikuljetus	Ison laitteen siirto	1 000,00 €
7	Varastointi	Tavaroiden varastointi tietyille määräjalle	750,00 €

Tilaus ID	Vastaanottaja ID	Päivämäärä	Palvelun nimi	Laskun päivämäärä	Maksettu	Kokonaishinta
36	4	30.12.2010	Varastointi	7.12.2010	Perinnässä	1 200,00 €

Kuva 19. *Palvelut* –taulun tietojen päivittyminen tietokantaan

Testitapaus 4: Asiakashistoria-raportin näkymän oikeellisuus ja päivittyminen uuden asiakastiedon kirjaamisen jälkeen.

Testasimme *asiakashistoria* –raportin näkymän päivittymistä lisäämällä *asiakastiedot* –tauluun uuden asiakastiedon. *Asiakashistoria* –raportin tulee päivittyä reaaliaikaisesti *asiakastiedot* –lomakkeen kanssa. Kyseistä raporttia käytetään esimerkiksi vuosikatsauksessa kokonaisasiakasmäärien tarkasteluun. Kuvassa 20. näkyy *asiakastiedot* –lomakkeen ja *asiakashistoria* –raportin yhteneväisyys, kun lisätty tieto syötettiin *asiakastiedot* –lomakkeelle.

**Asiakastiedot**

Asiakas ID: 7  
 Etunimi: Markku  
 Sukunimi: Kekäläinen  
 Lähiosoite: Kuusitorppa 8  
 Postinumero: 93600  
 Paikkakunta: Kuusamo  
 Maa: Suomi  
 Puhelinnumero: 0405656989  
 Sähköposti: Markku@google.com

Tee tilaus

**Asiakaslista**

Asiakas ID	Etunimi	Sukunimi	Lähiosoite	Postinumero	Paikkakunta	Maa
3	Sakari	Pietilä	Maariankatu 2	93600	Kuusamo	Suomi
4	Kari	Kankaanpää	Muikokkatu 4	89444	Kernilä	Venäjä
5	Matti	Puuhamaa	Ruiskatu 2 D	30880	Kempele	Suomi
6	Lasse	Kulkonen	Maariankatu 6 A	30880	Hämeenlinna	Suomi
7	Markku	Kekäläinen	Kuusitorppa 8	93600	Kuusamo	Suomi

Kuva 20. Asiakastiedot –lomakkeen ja asiakashistoria –raportin reaaliaikainen päivittyminen




### Testitapaus 5: Asiakastietojen päivittyminen *tilaukset*-lomakkeelle

Tässä testitapauksessa lisättiin *asiakastiedot* –lomakkeelle uusi asiakastieto ja päivitettiin jo-olemassaolevien asiakkaiden tietoja. Tarkoituksena oli testata tietojen päivittyminen oikeaoppisesti asiakastietoja hyödyntävälle *tilaukset* –lomakkeelle. Tiedonkulun on oltava yhtenevä ja reaaliaikainen asiakkaiden ja tilausten välillä, sillä koko reskontrasovellus ja yrityksen tietohallinto perustuu asiakkaiden ja tilausten yhtenevään hallintaan. Sovellus on suunniteltu niin, että *tilaukset* –lomake hakee tiedot *asiakastiedot* –lomakkeelta ja tilaustiedot välittyvät *asiakastiedot* –lomakkeella olevaan *tilaushistoria* –alilomakkeeseen.

Kuvassa 21. on lisätty uusi asiakastieto *asiakastiedot* –lomakkeelle ja sen jälkeen luotu uusi tilaus uudelle asiakkaalle *tilaukset* –lomakkeella. Toimenpide ei onnistu ja lisätyt tiedot eivät päivyty lomakkeille, jos yhteydet (*relationships*) kyseisten taulujen välillä eivät ole oikeaoppisesti luotuja. Testauksen tuloksena voidaan kuitenkin todeta tietojen päivittymisen ja valittujen taulujen yhteyksien toimivan oikeaoppisesti.

## Asiakastiedot

Asiakas ID	<input type="text" value="3"/>	
Etunimi	<input type="text" value="Sakari"/>	
Sukunimi	<input type="text" value="Pietilä"/>	
Lähiosoite	<input type="text" value="Maariankatu 2"/>	
Postinumero	<input type="text" value="93600"/>	
Paikkakunta	<input type="text" value="Kuusamo"/>	
Maa	<input type="text" value="Suomi"/>	
Puhelinnumero	<input type="text" value="0506789898"/>	
Sähköposti	<input type="text" value="Maaria@Kuusamo.fi"/>	

## Tilaukset

Asiakas ID	Etunimi	Sukunimi	
3	Sakari	Pietilä	
Lähiosoite		Postinumero	
Maariankatu 2		93600	
Paikkakunta	Maa	Puhelinnumero	
Kuusamo	Suomi	0506789898	
Sähköposti			
Maaria@Kuusamo.fi			
Tilaus ID	Vastaanotettu	Vastaanottaja	
28		Kerkko	Kallio
Palvelu ID	Palvelun nimi		
1	Kassakaappi		
Palvelun kuvaus			
Kassakaapin siirto			
Lisätiedot			
Laskun päivämäärä	Viitenumero	Kokonaishinta	Laskun tila
		600,00 €	Maksettu

Asiakas: 
 Palvelu: 
 Vastaanottaja:

Kankaanpää	Kari	4
Kekäläinen	Markku	7
Kukkonen	Lasse	6
Pietilä	Sakari	3
Puuhamaa	Matti	5

Kuva 21. Lisätyn asiakastiedon päivittyminen tilaukset -lomakkeelle

### 4.3 Systeemitestaus

Systeemitestausvaihe (*system testing*) toteutettiin toimeksiantajan toimitiloja vastaavassa ympäristössä. Käyttöjärjestelmä, näyttö ja tärkeimmät keskusyksikön komponentit luotiin vastaamaan tulevaa ympäristöä, jossa sovellusta tullaan käyttämään. Testauksen vaiheet jaettiin kolmeen osaan: Kokoonpanotestaus (*configuration testing*), käytettävyydestaus (*usability testing*) ja tietoturvatestaus (*security testing*). Systeemitestausvaiheeseen valittuihin testausmuotoihin päädyttiin lähinnä sovelluksen pienimuotoisuuden vuoksi. Olisi aiheetonta testata suorituskykytestausta (*performance testing*) tai joustavuustestausta (*resilience testing*) näin pienellä, MS-Access 2007 pohjaisella laskutusjärjestelmällä, jota käytetään jatkuvasti vain yhdessä työpisteessä. Tätä testausvaihetta ei yleisesti suorita itse ohjelmistosuunnittelija, kutsuttiin toimeksiantajan edustaja suorittamaan systeemitestauksen 13.12.2010 seuraavalla laitteiston kokoonpanolla:

- Windows 7 professional edition
- Näyttö 19" (1280 x 1024)
- Prosessori AMD Athlon II X 2 245
- Keskusmuisti 4 Gt DDR3
- MS Office 2007
- Tulostin HP Deskjet 5655

#### Kokoonpanotestaus (*configuration testing*)

Kokoonpanotestaus on yksi osa systeemitestausta. Kyseisessä testausvaiheessa pyritään testaamaan luodun sovelluksen toimivuutta tietokoneen ulkoisten ja sisäisten laitteiden kanssa, esimerkiksi sovelluksen yhteentoimivuus määrätyn tulostimen kanssa. Tämän testausvaiheen oletusmäärittelynä pidetään tilaa, jossa sovellus kykenee toimimaan lähes missä tahansa laitteistokokoonpanossa, jos kokoonpano mahdollistaa MS-Access 2007:n asennuksen. Yleisesti tämä systeemitestauksen vaihe

suoritetaan kaikilla mahdollisilla kokoonpanoilla, joilla sovellusta ehkä tultaisiin käyttämään (Haikala & Märijärvi 2004, 290). Tässä tapauksessa testattavaksi riitti kuitenkin yksi yllämainittu testikokoonpano, sen vastatessa tulevaa käyttöympäristöä.

Yllä mainitulla kokoonpanolla suoritettu testausvaihe jakaantui ohjelmistoasennukseen, käyttöönottoon ja toimivuuteen, sekä tulostimen testaukseen. Testausvaiheet sujuivat oletusten mukaisesti ongelmitta, sillä valtaosa MS-Access 2007 toiminnan kannalta edellytetyistä kriteereistä oli jo projektin suunnitteluvaiheessa selvitetty. Sovelluksen toiminta testiympäristössä vastasi odotuksia eikä yhteentoimivuusongelmia ilmennyt.

Oheislaitteista testasimme HP deskjet 5655 –tulostimen yhteentoimivuuden luodun sovelluksen kanssa. Tulostimen asennus sujui ongelmitta MS Windows 7:n tunnistettua oheislaitteen automaattisesti, jolloin esimerkiksi ajurien erillistä asentamista ei tarvittu. Kuvasta 22. voidaan havaita asennetun sovelluksen lasku-lomakkeen tulostusnäkyvän oikeellisuus sekä toimivuus laskua tulostettaessa.

<b>Osoite: Kuormakatu 8, 20500 Lieto Puhelin: 040-7878336 Pankki: 1238989-556849 ALV rek.</b>		
<b>Saajan tilinumero:</b>	1238989-556849	<b>Palvelun nimi:</b> Kassakaappi
<b>Saaja:</b>	<b>YKKÖSMIEHET OY</b> <b>Kuormakatu 8 20500 Lieto</b>	
<b>Etunimi:</b>	Sakari	<b>Viitenumero:</b>
<b>Sukunimi:</b>	Pietilä	
<b>Lähiosoite:</b>	Maariankatu 2	
<b>Postinumero:</b>	93600	
<b>Paikkakunta:</b>	Kuusamo	
<b>Maa:</b>	Suomi	
<b>Eräpäivä:</b>		<b>Hinta:</b> 600,00 €

Print

Printer

Name: hp deskjet 5600 series (HPA) Properties

Status: Valmis

Type: hp deskjet 5600 series (HPA)

Where: USB001

Comment:  Print to File

Print Range

All

Pages From:  To:

Selected Record(s)

Copies

Number of Copies: 1

Collate

1 1 2 2 3 3

Setup... OK Cancel

Etusivu

Kuva 22. Lasku-lomakkeen tulostustila

### Käytettävyystestaus (*usability testing*)

Käytettävyystestauksella varmistetaan luodun sovelluksen helppokäyttöisyyttä, toimivuutta ja hyödyllisyyttä testiympäristössä sovelluksen tulevia käyttäjiä ajatellen. Tämä testausvaihe koostuu valtaosin käyttöliittymätestauksesta, jolloin tärkeimpänä testattavana toimintona voidaan pitää navigoinnin toimivuutta (Haikala & Märijärvi 2004, 291). Olennaiseksi testauskohteeksi

valittiin tiedonsyöttö sovellukseen, navigointi lomakkeiden välillä, sekä sovelluksen sulkemista. Tarkasteltavana oli myös edellä mainittujen toimintojen helppokäyttöisyys, sekä suoritustavan selkeys. Kyseisiä testausvaiheita sivuttiin myös integraatiotestauksen aikana.

Testausvaiheen tuloksena voidaan todeta yllä mainittujen toimintojen oikeellisuus. Navigointi lomakkeiden välillä, sovelluksen sulkeminen ja avaaminen, toimintojen selkeys ja helppokäyttöisyys toimivat kaikki määrittelyjen mukaisesti. Testiympäristö täytti sovelluksen toiminnalle olennaiset kriteerit ja vaatimukset.

#### Tietoturvatestaus (*security testing*)

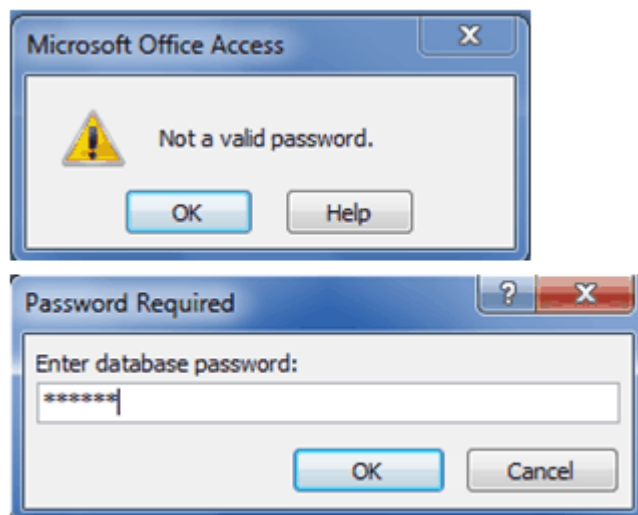
Tietoturvatestaus on välttämätön testivaihe kaikille sovelluksille, joissa voidaan arvioida liikkuvan luottamuksellista tietoa. Käytännössä testausvaiheella pyritään kartoittamaan sovelluksen tietosuojaa sisäisiä tai ulkoisia, ei-toivottuja tunkeutumisyriksiä kohtaan. Testivaiheen aikana sovellus kohdistetaan tunkeutumisyriksen alaiseksi niin, että sovellukseen pyritään tunkeutumaan sisään ilman salasanaa tai ohjelmistolauseita yritetään muokata tai kopioida. Järjestetyt tunkeutumisyriukset tapahtuivat samalta työpisteeltä ilman erillisiä tunkeutumishjelmistoja.

Testattavan sovelluksen ollessa rakenteeltaan pieni ja mahdollisten tietoturvaohjelmien vähäisiä, loimme sovellukseen kirjautumisen yhteyteen salasanan, sekä salasimme ohjelmiston VBA-ohjelmalauseet erillisellä salasanalla. Kuten Lambert, Lambert III ja Preppernau kirjassaan mainitsee: Luvaton pääsy MS-access 2007 pohjaiseen sovellukseen voidaan estää kahdella tavalla: suojaamalla VBA-ohjelmalauseet salasanalla, tai tallentamalla tietokannan Microsoft Database Executable (ACCDE) -tiedostona. ”Jos salana asetetaan ohjelmalauseisiin se säilyy saatavilla muokkausta varten kenelle tahansa, joka tietää salasanan. Jos tietokanta tallennetaan ACCDE-tiedostona, ihmiset jotka käyttävät tietokantasovellusta voivat suorittaa

ohjelmalauseet, mutta he eivät voi tutkia tai muokata sitä.” (Lambert, Lambert III & Preppernau 2008, 277).

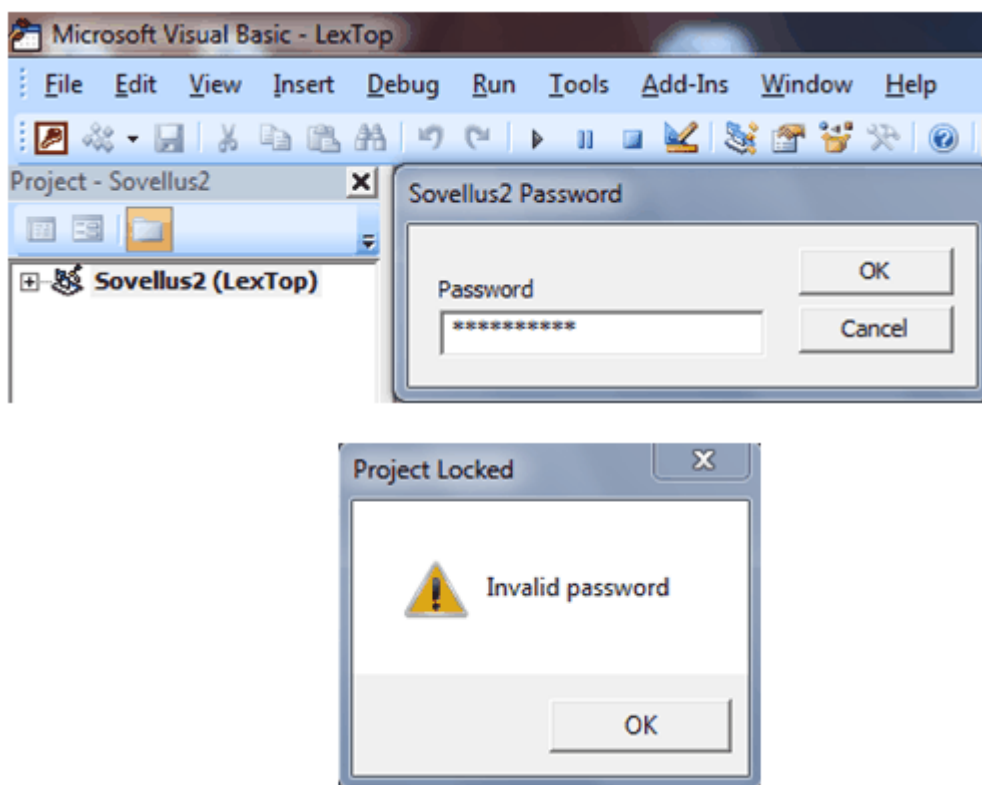
Katsoimme edellä mainitut salaustoiminnot riittäviksi tulevia yritystietoja ajatellen.

Kuvassa 23. on esitetty kuvakaappaus tilanteesta, jossa sovellukseen yritetään kirjautua sisään väärällä salasanalla. Kuten kuvasta käy ilmi, herjaa sovellus salasanan epäpätevyydestä eikä kirjautuminen näin ollen onnistu. Sovelluksen käyttö on rajattu vain asiakasyrityksemme työntekijöille, joille salana on jaetaan käyttöönoton yhteydessä.



Kuva 23. Kirjautumisyritys väärällä salasanalla

Kuvassa 24. on esitetty tilanne, jossa ohjelmistoon on päästy kirjautumaan onnistuneesti. Tunkeutumisyhteydessä ohjelmiston VBA-lausekkeita yritetään päästä muokkaamaan tai kopioimaan, mutta kuten kuvasta käy ilmi, ei toivottu toimenpide ole mahdollista ilman pyydettyä salasanaa. Ohjelmiston VBA-lauseet ovat nähtävissä ja muokattavissa vain, jos käyttäjä antaa oikean salasanan.



Kuva 24. VBA-ohjelmistolauseiden muokkausyritys



## 5 KÄYTTÖÖNOTTO

Sovelluksen käyttöönotto ja luovutus toimeksiantajalle tapahtuu 29.12.2010. Käyttöönottovaiheessa valmis sovellus asennetaan toimeksiantajan työasemalle ja varmistetaan sen toimivuus yhdessä toimeksiantajan edustajan kanssa uudessa työympäristössä. Käyttöönotto toteutetaan yllä mainittuna päivänä niin, että prosessi viedään alusta loppuun yhdessä päivässä.

Käyttöönoton yhteydessä tapahtuu myös sovelluksen salasanojen luovutus pääkäyttäjälle, jonka vastuulla on jakaa tietoja tarvittaessa eteenpäin. Tiedonsiirron vastuu vanhan ja uuden laskutusjärjestelmien välillä on toimeksiantajalla.

Tulemme sopimaan myös tietyistä ylläpitotoimenpiteistä sovellusta koskien, sillä kuten ohjelmistosuunnittelussa yleensä, kestää viikkoja ennen kuin uudet käyttäjät ovat sisäistäneet luodun sovelluksen ja havainneet siinä esiintyneet muutostarpeet. Ylläpidolliset toimenpiteet hoidetaan tilannekohtaisesti aina tarpeen niin vaatiessa.

## 6 OMAN TYÖN ARVIOINTI

Opinnäytetyömme tavoite oli suunnitella ja toteuttaa laskutusjärjestelmä toimeksiantona Ykkösmiehet Oy:lle. Työ toteutettiin MS-Access 2007 – tietokantaohjelmistolla. Ohjelmiston kehitystyön vaihejako- ja toteutusmalleina hyödynsimme sovelletun vesiputousmallin ja nopean kehityksen menetelmiä. Projekti alkoi tarvittavien tietojen kartoituksella, jota seurasi suunnittelu-, toteutus-, ja testausvaihe.

Vaihejakomallina vesiputousmalli oli selkeä valinta MS-access –pohjaiseen laskutusjärjestelmään. Eteneminen vaihe vaiheelta suunnittelusta testaukseen on ollut järjestelmällistä ja selkeää loogisen vaihejaon tuloksena. Lisäksi nopean kehityksen menetelmä on tuonut oman mielekkyytensä ohjelmistokehityksen eri vaiheisiin, sillä keskittyminen nykyhetkeen ja sovelluksen osittainen suunnittelu sitä tehtäessä on osoittautunut meille sopivaksi työskentelytavaksi.

Luotu laskutusjärjestelmä vastaa sille alussa määriteltyjä vaatimuksia. Laskutusjärjestelmän toiminnot toteutettiin helppokäyttöisiksi ja selkeiksi, sekä koko sovellus toimeksiantajalle ominaista ulkoasua käyttäen. Suunnitteluvaiheessa tavoitteemme oli luoda sovelluksesta toimeksiantajan näköinen, helppokäyttöinen laskutusjärjestelmä, jonka toiminnot kykenevät vastaamaan lukuisiin laskutusjärjestelmille ominaisiin haasteisiin. Sovelluksen toimivuus ja helppokäyttöisyys testattiin yhdessä toimeksiantajan edustajan kanssa.

Palaute laskutusjärjestelmästä on ollut positiivista. Erityisesti sovelluksen yhtenevä ulkoasu, selkeä käyttöliittymä, sekä monipuoliset toiminnot ovat saaneet erityiskiitosta toimeksiantajan suunnalta. Onnistunut ja kiiteltä yhteistyö meidän ja toimeksiantajan välillä kruunasi projektin ja teki siitä ylipäänsä mahdollisen toteuttaa. Olemme myös itse äärimmäisen tyytyväisiä luotuun sovellukseen ja yhteistyöhön meidän ja Ykkösmiehet Oy:n välillä.

Opinnäytetyönä laskutusjärjestelmän toteutus toimeksiantona oikealle yritykselle on ollut haastava ja kasvattava kokemus. Toimivan yrityksen tarpeiden kartoittaminen ja niiden pohjalta täysin uuden laskutusjärjestelmän rakentaminen toi eteen lukuisia mielenkiintoisia haasteita. Koulutusohjelmamme tarjoamat työkalut antoivat hyvät lähtökohdat tietokantasovelluksen toteutukseen, mutta kuten yleensä, työ on paras opettaja.

Opinnäytetyö parityönä on sujunut erinomaisesti. Työn haasteellisuus tietysti aina lisääntyy henkilömäärän kasvaessa, sillä esimerkiksi aikataulut, elämäntilanteet ja henkilökemiat tuovat mukanaan omat haasteensa. Kokonaisuutta arvioiden, voimme kuitenkin todeta, että yllä mainitut haasteet ollaan voitettu ja toimeksianto toteutettu säännöllisin aikatauluin ja kannustavin ilmapiirein sovitussa määräajassa.

## LÄHTEET

Aalto-yliopisto 1998. WEB-palveluiden käytettävyys ja tuotanto. Viitattu 24.04.2010  
<http://www.uiah.fi/mediastudio/survey4/index.html>.

Haikala, Ilkka. ; Märijärvi, Jukka. 2004. Ohjelmistotuotanto.

Hovi, Ari. ; Huotari, Jouni. ; Lahdenmäki, Tapio. 2005. Tietokantojen suunnittelu & indeksointi.

Jyväskylän yliopisto 2010. Ohjelmistotestaus ja siinä käytettävät työkalut. Viitattu 27.11.2010  
<http://www.mit.jyu.fi/opiskelu/seminaarit/ohjelmistotekniikka/testaus/#RTFToC7>.

Lambert, Steve. ; Lambert, M. Dow III. ; Preppernau, Joan. 2008. Microsoft Access 2007 step by step.

McConnel, Steve. 1996. Rapid Development: Taming Wild Software Schedules.

Microsoft 2010. Tietokannan suunnittelun perusteet. Viitattu 05.11.2010  
<http://office.microsoft.com/fi-fi/access-help/tietokannan-suunnittelun-perusteet-HA001224247.aspx#BMfindandorganize>.

Oulun seudun ammattiopisto 2010. Johdatus tietojärjestelmiin. Viitattu 18.10.2010  
[http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien\\_kaytto\\_ja\\_kehittaminen/johdatus\\_tietojarjestelmiin/kehittamistyon\\_vaiheet\\_ja\\_elikaarimallit/kehittamistyon\\_vaiheet\\_ja\\_elinkaarimallit\\_asia.htm](http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elinkaarimallit_asia.htm).

Valtion teknillinen tutkimuskeskus 2010. Hyvä käyttöliittymäsuunnittelu lähtee käytön tarpeista. Viitattu 15.11.2010  
<http://www.vtt.fi/inf/julkaisut/muut/1998/av7-98.pdf>.

Wikipedia 2010. Ohjelmistotuotanto. Viitattu 14.10.2010  
<http://fi.wikipedia.org/wiki/Ohjelmistotuotanto#Ohjelmistosuunnittelu>.