

Laura Friberg

PIENYRITYKSEN PROJEKTIN
KUSTANNUSSEURANTAOHJELMA

Tietotekniikan koulutusohjelma

2009



Satakunnan ammattikorkeakoulu

PIENYRITYKSEN PROJEKTtien KUSTANNUSSEURANTAOHJELMA

Friberg, Laura
Satakunnan ammattikorkeakoulu
Tekniikka ja merenkulku Rauma
Tietotekniikan koulutusohjelma
Toukokuu 2009
Yritys: Viishanke Oy
Ohjaaja: yliopettaja, FT Yrjö Auramo
UDK: 004.65
Sivumäärä: 61

Asiasanat: Access, Visual Basic, Visual Studio, ohjelmointi

Tämän opinnäytetyön tavoitteena oli suunnitella ja luoda pienyrityksen projektien kustannusseurantaohjelma. Ohjelman avulla työn tilaaja, Viishanke Oy, voi helposti ja käyttäjätystävällisesti raportoida ja tulostaa projektiansa tiedot.

Ohjelma koostuu tietokannasta ja käyttöliittymästä. Työ aloitettiin ohjelman osioiden perusteellisella suunnittelulla ja tiedon keruulla. Ohjelman tietokanta tehtiin Microsoft Office Access 2007 -ohjelmalla ja käyttöliittymä Visual Studio 2008 -ohjelmalla Visual Basic 2008 -ohjelmointikielellä. Tietokanta luotiin ensin, jonka jälkeen luotiin käyttöliittymä.

Kustannusseurantaohjelma luotiin Viishanke Oy:n jokapäiväisen työn apuvälineeksi. Ohjelman avulla projektien eli toimeksiantojen tilaajien sekä tavarantoimittajien tiedot, kuten yritys- ja osoitetiedot, syötetään tietokantaan. Myös tuote- ja projektitiedot syötetään ohjelman avulla tietokantaan. Tietokantaan syötettyjä tietoja voidaan myöhemmin tarkastella ja muokata. Projektin lopulliset kustannukset tulostetaan näytölle selkeänä kustannusraporttina, jonka voi myös tulostaa paperiversioksi. Jo tehtyihin projekteihin ja niiden kustannuksiin voidaan helposti palata ohjelman avulla.

Kaikki työn rajauksessa määritellyt toiminnot toteutettiin ohjelmassa. Kustannusseurantaohjelman molemmat osiot, suunnittelu ja toteutus, onnistuivat. Ohjelman päätarkoitus, kustannusraportti, voidaan luoda helposti ja nopeasti. Kustannusraportista selviävät kaikki halutut tiedot yksinkertaisesti ja selkeästi.

PROJECT COST MONITORING PROGRAM OF SMALL BUSINESS

Friberg, Laura
Satakunta University of Applied Sciences
Technology and Maritime Management Rauma
Degree Programme in Information Technology
May 2009
Commissioned by Viishanke Oy
Tutor: Yrjö Auramo, PhD, Principal Lecturer
UDC: 004.65
Number of Pages: 61

Keywords: Access, Visual Basic, Visual Studio, programming

The purpose of this Bachelor's thesis was to design and create a cost monitoring program of projects of a small business. The commissioner of the program, Viishanke Oy, can easily report and print out information about projects with the program.

The program consists of a database and a user interface. The work was started with complete designing of the parts of the program and data collecting. The database of the program was made with Microsoft Office Access 2007 software and the user interface with Visual Studio 2008 software and Visual Basic programming language.

The project cost monitoring program was created for Viishanke Oy to make everyday work easier. All the information about the clients of projects, product suppliers, projects and products can easily be fed in to the database with the program. The information fed in to the database can be examined and edited later. The final costs of the project are printed out on the monitor in an uncluttered cost report, which can also be printed out on the paper.

All the functions assigned in the beginning meeting were executed in the program. Both parts of the cost monitoring program, designing and creating, were successful. The main purpose of the program, the cost report, can be created easily and fast.

ALKUSANAT

Haluan kiittää Viishanke Oy:n toimitusjohtajaa Osmo Haaslahtea tilaisuudesta toteuttaa tämä opinnäytetyö. Lisäksi esitän kiitokseni opinnäytetyön ohjaavalle opettajalleni Yrjö Auramolle yksityiskohtaisesta opastuksesta ja kullannarvoisista vinkeistä.

Erityiskiitokset kaikille työssä mukana olleille ja siihen vaikuttaneille.

Raumalla 20. huhtikuuta 2009

Laura Friberg

SISÄLLYS

ALKUSANAT

SISÄLLYS

SYMBOLI- JA TERMILUETTELO

1 YRITYSESITTELY	9
2 JOHDANTO	10
2.1 Työn rajaus	10
2.2 Ohjelman käyttötarkoitus	11
2.3 Työssä käytetyt käsitteet	11
2.3.1 Pienyritys	12
2.3.2 Projekti.....	12
2.3.3 Kustannus.....	12
2.3.4 Kustannusseuranta	12
2.3.5 Ohjelman suunnitleminen	13
2.3.6 Ohjelman luominen.....	13
2.3.7 Tuote	13
3 TYÖMENETELMÄT	14
3.1 Microsoft Office Access 2007	14
3.1.1 Relaatiotietokannat	14
3.2 Visual Studio 2008	15
3.2.1 Microsoft .NET Framework	15
3.2.2 Visual Basic 2008	16
3.2.3 ADO .NET	17
4 SUUNNITTELU	18
4.1 Tietokannan suunnitleminen	19
4.2 Käyttöliittymän suunnitleminen	20
5 TOTEUTUS	23
5.1 Tietokannan toteuttaminen	23
5.1.1 Yhteydet.....	25
5.1.2 Kyselyt	25
5.2 Yhteyden luominen tietokannan ja käyttöliittymän välille	27
5.3 Käyttöliittymän toteuttaminen.....	28

5.3.1 Virheentarkistus	28
5.3.2 Vetovalikkojen ominaisuudet	29
5.3.3 Tilamuuttuja-toiminto	31
5.3.4 Tietojen tallentaminen tietokantaan.....	33
5.3.5 Kustannusraportti	34
5.3.6 Tietojen poistaminen.....	35
5.3.7 Projektinnumero.....	37
5.3.8 Projektin aloitus- ja lopetuspäivämäärä-kentät.....	39
6 TULOKSET	40
6.1 Ohjelman yleiset ominaisuudet	40
6.2 Aloitus-/etusivu	41
6.3 Tietojen lisääminen	41
6.3.1 Tilaajien ja toimittajien lisääminen.....	42
6.3.2 Yhteyshenkilöiden lisääminen.....	43
6.3.3 Tuotteiden lisääminen.....	44
6.3.4 Projektien lisääminen.....	45
6.3.5 Yksiköiden lisääminen.....	46
6.3.6 Tuotteiden lisääminen projektiin	46
6.4 Tietojen tarkasteleminen ja muokkaaminen.....	48
6.4.1 Tilaajien ja toimittajien tarkasteleminen ja muokkaaminen	49
6.4.2 Yhteyshenkilöiden tarkasteleminen ja muokkaaminen	50
6.4.3 Tuotteiden tarkasteleminen ja muokkaaminen	51
6.4.4 Projektien tarkasteleminen ja muokkaaminen	52
6.4.5 Yksiköiden tarkasteleminen ja muokkaaminen	54
6.5 Tietojen poistaminen	54
6.6 Kustannusraportti	56
7 YHTEENVETO	59
7.1 Yleistä.....	59
7.2 Kehitysehdotukset	60
LÄHTEET	61
LIITE 1	

SYMBOLI- JA TERMILUETTELO

.NET Framework	Microsoftin kehittämä ohjelmistokomponenttikirjasto, jota Microsoftin Visual Studio .NET -ympäristössä kehitetyt ohjelmistot käyttävät.
ADO .NET	ActiveX Data Objects; kokoelma ohjelmistokomponentteja, joita käytetään tiedon hallintaan.
AutoCompleteMode	Vetovalikon ominaisuus, jonka avulla määritellään miten vetovalikko täydennetään.
AutoCompleteSource	Vetovalikon ominaisuus, jonka avulla määritellään lähde, jonka mukaan täydennys tehdään.
BASIC	Beginner's All purpose Symbolic Instruction Code; ohjelmointikieliperhe, johon kuuluu muun muassa Visual Basic.
BindingSource	Komponentti, joka hoitaa kommunikoinnin sovelluksen komponenttien ja DataSet-komponenttien välillä.
C#	Microsoftin .NET-konseptia varten kehittämä ohjelmointikieli.
C++	Yksi tärkeimmistä kaupallisessa ohjelmistokehityksessä käytettävistä ohjelmointikielistä.
COM	Component Object Model; Microsoftin vuonna 1993 kehittämä liittymärajapinta ohjelmistokomponenteille.
DataSet	Komponentti, johon tallennetaan tietokannan tietoja, jotta niitä voidaan käyttää koodissa.
DropDownList	DropDownStyle-ominaisuudeksi voidaan valita DropDownList; vetovalikkoon ei syötetä kirjaimia, vaan valinta suoritetaan valmiista vaihtoehdoista.
DropDownStyle	Vetovalikon ominaisuus, jolla määritellään miten vaihtoehdot näytetään.
IDE	Integrated development environment; samaan pakettiin integroitu ohjelmointiympäristö; joukko ohjelmia, jolla ohjelmoija toteuttaa ohjelmistoa.

J#	Visual Studio .NET:n tukema Microsoftin tekemä J++:aan pohjautuva versio Javasta.
Java	Sun Microsystemsin kehittämä laaja teknologiaperhe ja ohjelmistoalusta.
Käyttöliittymä	Sovelluksen osa, jonka avulla sovellus ja käyttäjä ovat vuorovaikutuksessa keskenään.
ListItems	AutoCompleteSource-ominaisuudeksi voidaan valita ListItems; vetovalikon vaihtoehtojen täydennys tehdään listalla olevien kohtien mukaan.
Microsoft	Yhdysvaltalainen maailman suurin ohjelmistoalan yritys; useimmissa tietokoneissa käytettävän Windows-käyttöjärjestelmän valmistaja ja myyjä.
Microsoft Access	Microsoft Office -ohjelmistopakettiin kuuluva tietokantojen hallintaohjelma.
Microsoft Office 2007	Microsoftin valmistama toimisto-ohjelmistopaketti Windows- ja Mac OS X -käyttöjärjestelmille.
Relaatiotietokanta	Tietokantamalli; kokoelma tietoja, joilla on yhteys toisiinsa.
SuggestAppend	AutoCompleteMode-ominaisuudeksi voidaan valita SuggestAppend; vetovalikossa esitetään syötetyillä kirjaimilla alkavat vaihtoehdot ja valinta täydennetään ensimmäisellä vaihtoehdolla.
TableAdapter	Välittäjäkomponentti, joka siirtää tietoa DataSetin ja tietokannan välillä.
Visual Basic	Microsoftin kehittämä BASIC-sukuinen kieli; nykyään Microsoftin .NET-perheeseen kuuluva ohjelmointikieli nimellä Visual Basic .NET.
Visual Studio	Microsoftin ohjelmankehitysympäristö, joka tukee useita ohjelmointikieliä.

1 YRITYSESITTELY

Opinnäytetyö tehtiin Viishanke Oy:lle. Viishanke Oy on vuonna 1978 perustettu raumalainen metallin ja suunnittelun alihankintayhtiö, jonka asiakaskuntaan kuuluu yhtiöitä aina perusteellisuudesta lääke- ja elektroniikkateollisuuteen. Viishanke Oy:n tuotantoa muokataan asiakkaiden tarpeiden mukaan, sillä se valmistaa lähinnä räätälöityjä toimeksiantoja. Tuotanto käsittää tuotteita maatalouskoneista ja teollisuuden alihankinnasta aina sairaalalaboratoriolaitteisiin. Erikoisosaamisena mainittakoon räätälöityjen säteilysuojien valmistus.

Viishanke Oy:ssä työskentelee tällä hetkellä 3 henkilöä. Yrityksen toimipiste sijaitsee Raumalla.



Kuva 1. Viishanke Oy:n logo

2 JOHDANTO

Tämän opinnäytetyön aihe on pienyrityksen projektien kustannus seurantaohjelman suunnitteleminen ja luominen. Ohjelma tehtiin Viishanke Oy:lle täysin sen tarpeita vastaavaksi. Viishanke Oy:llä on ollut käytössään samankaltainen ohjelma, mutta se ei vastannut toivotulla tavalla ominaisuuksiltaan yrityksen tarpeita.

Opinnäytetyön tavoitteena oli suunnitella ja luoda kustannus seurantaohjelma, jonka avulla Viishanke Oy voisi helposti ja käyttäjäystävällisesti raportoida projektiansa tiedot. Ohjelma koostuu käyttöliittymästä ja tietokannasta. Käyttöliittymä tehtiin Visual Basic -ohjelmointikielellä Visual Studio 2008 -ohjelmalla. Käyttöliittymään liitettiin Access-tietokanta. Työ aloitettiin ohjelman osioiden perusteellisella suunnittelulla ja tiedon keruulla. Tietokanta luotiin ensin ja kun se oli valmis, luotiin käyttöliittymä.

2.1 Työn rajaus

Työn aloitusta edelsi aloituspalaveri työn tilaajan Viishanke Oy:n toimitusjohtaja Osmo Haaslahden kanssa. Palaverissa rajattiin työn laajuus eli sovittiin ohjelmaan tulevista ominaisuuksista ja toiminnoista. Ohjelmaan tulevia ominaisuuksia ei voitu kuitenkaan rajata liian tiukasti, koska tiedettiin, että kaikkia ominaisuuksia ei voida heti työn alkumetreillä huomata tarpeellisiksi.

Aloituspalaverissa sovittiin, että ohjelmaan tulevat toiminnot, joilla tilaajan, toimittajan, tuotteen ja projektin tiedot syötetään tietokantaan. Tilaajan sekä toimittajan yrityksen nimi, osoite, puhelinnumero, faksi, sähköpostiosoite ja yhteyshenkilö ovat oleellisia tietoja. Tuotteen tietoihin syötetään tuotteen nimi, toimittaja, tuotekoodi ja hinta. Projektin nimi, tilaaja, aloitus- sekä lopetuspäivämäärä ja yhteyshenkilö ovat projektiin liitettävä tietoja.

Ohjelmaan haluttiin lisäksi toiminnot, joilla voidaan tarkastella ja muokata tietokantaan syötettyjä tietoja. Tärkein sovittu toiminto ja koko ohjelman päätarkoitus on se, että ohjelman avulla projektitiedot voidaan tallentaa niin, että kaikki projektiin tarvittut tuotteet hintoineen sekä projektin kokonaiskustannus voidaan tulostaa raporttina. Viishanke Oy:n taholta määriteltiin ohjelmaan halutut edellä mainitut toiminnot, mutta toimintojen toteutustapaa ei rajattu.

Syksyn 2009 aikana pidettiin jatkopalavereja, joissa käytiin läpi ohjelman sen hetkinen versio työn tilaajan kanssa. Ohjelmassa huomattiin ohjelmointivaiheessa muutamia puutteita, kuten yksilöllisen projektinumeron tarve ja tietojen poistamisen mahdollistavan toiminnon puute, joten välttämättömiä ominaisuuksia lisättiin jo melkein valmiiseen ohjelmaan myös ohjelmoinnin loppuvaiheessa.

2.2 Ohjelman käyttötarkoitus

Projektien kustannusseurantaohjelma luotiin Viishanke Oy:n jokapäiväisen työn apuvälineeksi. Ohjelman avulla projektien eli toimeksiantojen tilaajien sekä tavarantoimittajien tiedot, kuten yritys- ja osoitetiedot, syötetään tietokantaan. Myös tuote- ja projektitiedot syötetään ohjelman avulla tietokantaan. Projektin lopulliset kustannukset tulostetaan näytölle selkeänä kustannusraporttina, jonka voi myös tulostaa paperiversioksi. Ohjelma nopeuttaa ja selkeyttää Viishanke Oy:n työskentelytapoja, sillä projektien tietojen kirjaaminen paperille on monimutkaista ja hidasta. Vanhoihin projekteihin ja niiden kustannuksiin voidaan helposti palata ohjelman avulla.

2.3 Työssä käytetyt käsitteet

Tässä osiossa käydään läpi opinnäytetyön tärkeimmät käsitteet. Työn nimi on Pienyrityksen projektien kustannusseurantaohjelma. Työ jakaantui kahteen osioon, ohjelman suunnitteluun ja luomiseen. Näin ollen käsitteiden pienyritys, projekti, kustannus, kustannusseuranta, ohjelman suunnitteleminen ja ohjelman luominen merkitys tässä työssä esitellään yksityiskohtaisesti. Myös tuote-käsite esitellään.

2.3.1 Pienyritys

Kustannusseurantaohjelma tehtiin pienyrityksen tarpeita vastaavaksi. Viishanke Oy on pienyritys, sillä sen henkilökuntaan kuuluu kolme henkilöä. Ohjelma on suunniteltu siten, että sillä on yksi käyttäjä. Jos ohjelma halutaan tulevaisuudessa suuremman yrityksen käyttöön, on siihen tehtävä kattavia muutoksia.

2.3.2 Projekti

Tässä työssä projektilla tarkoitetaan yritykselle tulevaa tilausta eli toimeksiantoa. Projektilla viitataan myös projektin valmistamiseen tarvittaviin tuotteisiin ja niiden kustannuksiin. Viishanke Oy on esimerkiksi valmistanut tilaajalle kengänpohjan pesulaitteen, jolloin kengänpohjan pesulaite on projekti ja sen valmistamiseen tarvittavat tuotteet ja niiden kustannukset ovat projektin osia.

2.3.3 Kustannus

Käsite kustannus viittaa tässä työssä sekä yksittäisen tuotteen hintaan että kaikkien projektiin liitettyjen tuotteiden kokonaishintaan. Esimerkiksi levy voi olla yksi projektin tuotteista ja sillä on jokin hinta eli kustannus. Jos projektiin tarvitaan viisi levyä, on niiden yhteenlaskettu hinta projektin kokonaiskustannus.

2.3.4 Kustannusseuranta

Tässä työssä kustannusseurannalla tarkoitetaan kustannuksien tarkastelua ja dokumentointia. Projektin tuotteiden kustannukset listataan ja esitetään sekä erikseen että yhdessä kustannusraportissa. Raportista on helppo nähdä kulloisenkin ajankohdan hinnat niin yksittäisten tuotteiden kuin koko projektinkin osalta.

2.3.5 Ohjelman suunnitteleminen

Ohjelman suunnitteleminen toteutetaan ennen varsinaista ohjelman luontia. Suunnittelu on ohjelman perusta, joten siihen käytetään paljon aikaa ja vaivaa. Tietokannan suunnitteluvaiheessa mietittiin muun muassa mitä taulukoita ja kenttiä tietokantaan tarvitaan. Käyttöliittymän suunnitteluvaiheessa mietittiin esimerkiksi mitä lomakkeita tarvitaan ja mitä ne tulevat sisältämään. Yksityiskohtaisen suunnittelun avulla vältetään monet ongelmat itse ohjelmointivaiheessa ja säästetään aikaa. Ohjelman suunnitteleminen käydään läpi yksityiskohtaisemmin luvussa 4 ”Suunnittelu”.

2.3.6 Ohjelman luominen

Ohjelman luomisella tarkoitetaan ohjelman ohjelmointia eli ohjelman toteutusta. Tässä työssä ohjelman toteutus koostuu kahdesta osasta: tietokannan toteutus ja käyttöliittymän toteutus. Tietokanta luotiin Office Access 2007 -ohjelmalla ja sen jälkeen käyttöliittymä Visual Studio 2008 -ohjelmalla. Ohjelman luominen käydään tarkemmin läpi luvussa 5 ”Toteutus”.

2.3.7 Tuote

Käsite ”tuote” viittaa projektin toteutukseen tarvittavaan komponenttiin. Jos projektina on esimerkiksi pöytä, ja sen tekoon tarvitaan levy ja neljä jalkaa, ovat nämä komponentit projektin tuotteita. Itse pöytä ei ole tuote vaan projekti.

3 TYÖMENETELMÄT

Ohjelman tekoon käytettiin kahta ohjelmaa. Microsoft Office Access 2007 -ohjelmalla tehtiin ohjelman tietokanta ja Visual Studio 2008 -ohjelmalla Visual Basic 2008 -ohjelmointikielellä ohjelman käyttöliittymä. Ohjelman tietokanta on relaatiotietokanta.

3.1 Microsoft Office Access 2007

Microsoft Office Access 2007 on relaatiotietokantasovellus, joka kuuluu Microsoft Office 2007 -ohjelmistopakettiin. Se on tietokantojen hallintaohjelma, jonka lukuisten työkalujen avulla tietoja voidaan muun muassa jäljittää, jakaa ja raportoida. Access sisältää laajan kirjaston, joka mahdollistaa monet ohjatut toiminnot, kuten kyselyjen ja raporttien luomisen. Ohjattujen toimintojen ja mallipohjien ansiosta ohjelma on helppokäyttöinen ja näin ollen suosittu. /1, s. xi, 2/

Koska Access kuuluu Microsoft Office 2007 -pakettiin, samat komennot ja pikakuvakkeet pätevät kaikissa sen ohjelmissa. Tämän integroinnin ansiosta Accessin tietoja voidaan helposti jakaa esimerkiksi Excelin tai Wordin välillä. /1, s. 10/

3.1.1 Relatiotietokannat

Yksinkertaiset tietokantaohjelmat tallentavat tiedot vain yhteen taulukkoon. Access on monimutkaisempi relaatiotietokantaohjelma. /1, s. 9/ Relatiotietokanta sisältää useita taulukoita, joissa tiedot esitetään riveillä ja sarakkeissa. Tietokannan tiedot tallennetaan taulukoihin siten, että yksi tieto tallennetaan vain yhteen taulukkoon. Relatiotietokantaan tallennetaan myös tieto siitä, miten eri taulukot kytkeytyvät toisiinsa. /3/

Relaatiotietokannoissa on sääntöjä, joiden avulla tietokannasta tulee eheä ja looginen. Jokin sarake voidaan määritellä perusavaimeksi, jonka avulla tietue yksilöidään. /4, s. 2-3/ Relatiotietokannan tiedot voidaan ottaa käyttöön heti tietojen tallentamisen jälkeen. Eri taulukkojen välille luodaan yhteyksiä eli relaatioita. Näin tietojen muutos on tehtävä vain yhteen paikkaan kerrallaan, jolloin tietojen päivitys nopeutuu ja tarkentuu. /3/

3.2 Visual Studio 2008

Visual Studio on graafinen sovelluskehitin Windows-pohjaisiin ympäristöihin. /5, s. 2/ Siinä yhdistyvät kääntäjä, koodieditori, visuaaliset suunnitteluvälineet ja virheenjäljitin. /5, s. 2-3/ Se tukee useita ohjelmointikieliä, kuten Visual Basic, C++, C# ja J#. Visual Studio on osa Microsoft .NET -ympäristöä. /6, 7/ Visual Studion ohjelmien kirjoittamiseen käytettävää kehitysympäristöä kutsutaan nimellä IDE eli Integrated Development Environment. /5, s. 17/

Vuonna 2008 julkistettu Visual Studio 2008 eli Visual Studio 9.0 on uusin versio Visual Studio -sovelluskehittäjästä. /5, s. 15/ Visual Studiosta on julkaistu kuusi versiota, joista ensimmäinen, Visual Basic 97, julkaistiin vuonna 1997. Se oli ensimmäinen Microsoftin julkaisema usean eri ohjelmointikielen yhteinen ohjelmointityökalu. /7/

3.2.1 Microsoft .NET Framework

Helmikuussa 2002 julkaistiin Visual Studio 2002. Tällöin julkaistiin myös COM-ympäristöstä kehitetty Microsoft .NET -ohjelmointiympäristö. Vanhassa ja mutkikkaassa COM-ympäristössä käytettiin perinteisiä Win32-sovelluksia, kun taas Visual Studio 2002 alkaen käytetään vain .NET-ympäristössä toimivia sovelluksia. /5, s. 2 ja 24/

Microsoft .NET Framework eli .NET Framework tai lyhyesti .NET on virtuaalikoneeseen perustuva ajoympäristö sekä laaja luokkakirjasto. /5, s. 23/ .NET:ssä ja Java-ympäristössä on paljon samoja ominaisuuksia, sillä molemmat perustuvat virtuaalikoneen käyttöön. Muita ympäristöjen yhteneväisyyksiä ovat esimerkiksi oliopohjaisuus,

laaja luokkakirjasto sekä automaattinen muistinhallinta. /5, s. 24/ Toisin kuin Java, .NET tukee useita ohjelmointikieliä. /5, s. 2/

.NET yhteensopiville ohjelmointikielille on määritelty tyyppijärjestelmä, jonka avulla kaikki ohjelmointikielet voivat vaihtaa tietoja keskenään ja muun muassa periyttää uusia luokkia toisilla kielillä kirjoitetuista luokista. .NET-sovellukset vaativat ajamiseensa .NET-ajoympäristön, jonka voi ladata ilmaiseksi Microsoftin web-sivuilta. /5, s. 24/ .NET-ympäristön laajaa luokkakirjastoa voidaan hyödyntää Visual Studiolla tehdyissä sovelluksissa. Esimerkiksi .NET-versiot 3.0 ja 3.5 ovat tuettuina Visual Studiossa. /5, s. 3/ Visual Studio 2008 on ensimmäinen tuoteversio, joka tukee yhtä aikaa useampaa kuin yhtä .NET-versiota. /5, s. 30/

3.2.2 Visual Basic 2008

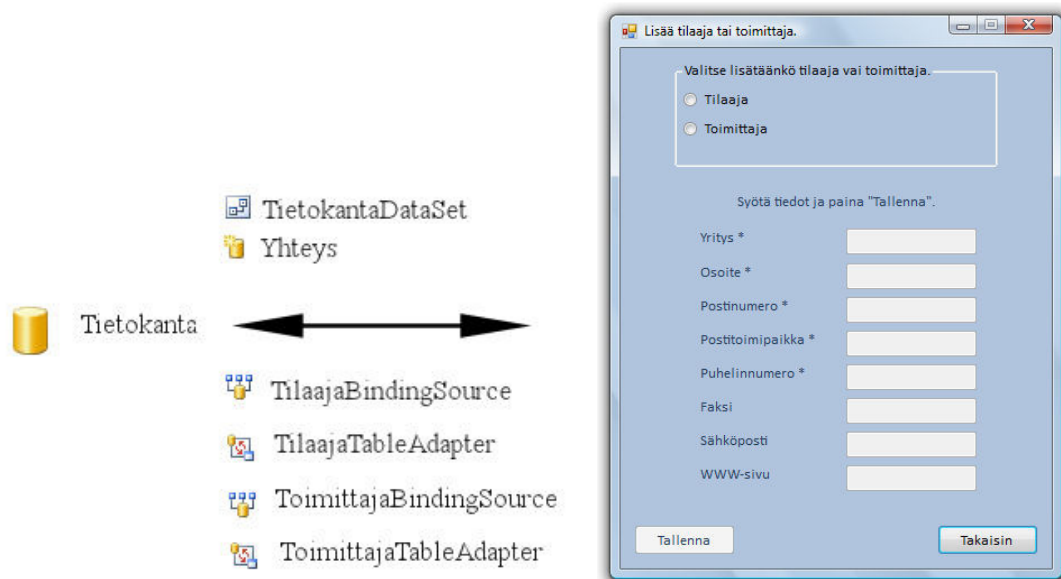
Visual Basic 2008 on Microsoft .NET -perheeseen kuuluva ohjelmointikieli. Se julkaistiin 19.11.2007 samaan aikaan Microsoft .NET Framework 3.5 -version kanssa. Visual Basic 2008:sta käytetään myös nimiä Visual Basic .NET 2008 ja VB.NET 2008 /6/ ja se on Visual Basic .NET -kielen versio 9.0. /5, s. 2/

Visual Basic on BASIC-sukuinen kieli. Tämä näkyy kielen lauseopissa, sillä se periytyy suoraan Microsoftin aiemmista tuotteista. Komennot erotetaan tarvittaessa kaksoispisteillä. Lohkot määritellään sijoittamalla lohkon alkuun ja loppuun varatut sanat. Visual Basicilla on mahdollisuus tehdä tyyppitöntä ohjelmointia eli muuttujia voidaan määritellä ilman tietotyypin määrittelyä /8, s. 33/. Visual Basic -ohjelmaa on mahdollista tulkata virheiden paikannusta varten. Ohjelman ajo voidaan keskeyttää halutulla hetkellä ja keskeytyksen aikana muuttujien arvoja voi tarkastella ja muuttaa. /6/

3.2.3 ADO .NET

ADO .NET on tietojenkäsittelytekniikka Visual Basic 2008:ssa. Se esiteltiin ensimmäisen kerran Microsoft Visual Studio .NET 2002:ssa ja se on tietokantaohjelmoinnin standardimalli myös Visual Studio 2008:ssa. /9, s. 438/

ADO .NET on keskeinen osa .NET arkkitehtuuria. Sen palveluita käytetään tietokantakäsittelyssä tiedon välittämiseen ja käsittelyyn. ADO .NET:llä voi hakea tietoa relaatio-tietokantojen lisäksi muun muassa tiedostoista ja sähköpostijärjestelmistä /4, s.44/. DataSet tarjoaa monipuoliset mahdollisuudet tiedon paikalliseen käsittelyyn ilman tietokantayhteyttä. ADO .NET-palveluita voi hyödyntää sekä Windows- että web-käyttöliittymien toteuttamisessa. ADO .NET:ssä tietokannan tietojen noutamiseen ja muokkaamiseen käytetään eri objekteja. /9, s. 439/



Kuva 2. ADO .NET

Kuvassa 2 esitellään projektien kustannuseurantaohjelman ”Lisää tilaaja tai toimittaja” -lomakkeeseen liittyvät ADO .NETin käyttämät objektit. TietokantaDataSet on tietojoukko, joka luodaan edustamaan tietokannan haluttuja kenttiä. TilaaajaBindingSource on välittäjäkomponentti, joka toimii välittäjänä Tilaaaja-taulukon ja lomakkeella olevien sidottujen objektien välillä. TilaaajaTableAdapter on välittäjäkomponentti, joka siirtää tietoa TietokantaDataSetin ja tietokannan välillä. /9, s. 452/

4 SUUNNITTELU

Ohjelman suunnittelu aloitettiin syyskuussa 2009. Työn tilaaja oli pyörätellyt ideaa yrityksen tarpeita vastaavasta kustannusseurantaohjelmasta jo tovin, joten ideoita ja ajatuksia oli heti työn suunnittelun alkuvaiheessa paljon. Ohjelman ominaisuuksista ja sen käyttötarkoituksista neuvoteltiin työn tilaajan kanssa moneen otteeseen syksyn 2009 aikana. Työn tilaajalla oli selkeä kuva siitä, mitä ominaisuuksia ohjelmaan haluttiin ja ne kirjattiin. Halutut ominaisuudet on määritelty osiossa 2.1 ”Työn rajaus”. Työn toteutuksen suunnitteluun käytettiin paljon aikaa, sillä tietokannan ja ohjelman tarkka suunnittelu säästää aikaa ja työvaiheita itse työn toteutusvaiheessa.

Ohjelman suunnittelussa pyrittiin erityisesti ottamaan huomioon käyttäjän rooli suunnittelijana. Työn tilaaja eli ohjelman lopullinen käyttäjä vastaa ohjelmasta ja tietokantaan syöttämistään tiedoista, joten käyttäjällä on myös valta päättää ohjelmasta. Siksi käyttäjän on oltava myös yksi ohjelman suunnittelija. Ohjelman kehittämisessä tehdään jatkuvasti päätöksiä ja lopullinen päätösvalta on oltava käyttäjällä. /10, s. 8-9/

Työ aloitettiin tietokannan perusteellisella suunnittelulla. Kun tietokanta oli suunniteltu ja tehty, aloitettiin käyttöliittymän suunnitteluprosessi ja tekovaihe. Tietokannan rakenne säilyi alkuperäisessä muodossaan, eli sitä ei tarvinnut muuttaa työn missään vaiheessa. Tarkasta suunnittelusta huolimatta käyttöliittymään piti tehdä muutamia muutoksia, jotka huomattiin olennaisiksi vasta, kun käyttöliittymää oli jo alettu luoda. Tämä ei kuitenkaan ollut ongelma, sillä muutoksiin oli varauduttu jo etukäteen joustavalla aikataululla.

4.1 Tietokannan suunnitteleminen

Tietokannan suunnitteluun on syytä käyttää voimavaroja, sillä oikein suunniteltu tietokanta mahdollistaa päivitettyjen ja tarkkojen tietojen käyttöönoton. Tietokannan suunnittelu takaa sen, että tietokannasta tulee tarpeita vastaava ja mukautettava.

Tietokannan tiedot pyrittiin jakamaan taulukoihin siten, että toistuvia tietoja olisi mahdollisimman vähän. Tietokannan tulee antaa Accessille tiedot, joita tarvitaan, kun on tarpeen liittää yhteen eri taulukoita. Hyvä tietokanta tukee tietojen tarkkuutta sekä yhdenmukaisuutta ja myös varmistaa tiedot. Tiedot on myös voitava mukauttaa käsittely- ja raportointitarpeisiin. /11/

Jokaisen taulukon kohdalla oli päätettävä, mitkä tiedot ovat pakollisia tietoja. Oli mietittävä, mitkä tiedot varmasti tiedetään tietoja syöttäessä ja mitkä selviävät mahdollisesti vasta myöhemmin. Kaikilla yrityksillä ei esimerkiksi ole www-sivuja, joten sitä ei voitu asettaa pakolliseksi tiedoksi. Projektin lopetuspäivämäärä on tieto, joka selviää usein vasta projektin tietojen lisäyksen jälkeen, joten myöskään sitä ei ole pakko asettaa projektin tietoihin. Pakolliset ja ei-pakolliset tiedot eritellään luvussa 6 ”Toteutus” erikseen lomakkeiden esittelyjen yhteydessä.

Tietokannan suunnittelussa oli seitsemän eri vaihetta. Tietokannan suunnittelu aloitettiin tietokannan tehtävän määrittelyllä. Tämän työn tietokannan tehtävänä on tallettaa kaikki yrityksen projekteihin liittyvät tiedot. Seuraavaksi kerättiin tiedot, jotka tietokantaan halutaan tallettaa, kuten yritysten nimet, osoitteet ja niin edelleen. Tiedot jaettiin taulukoihin pääkokonaisuuksien mukaan. Tässä työssä taulukoita ovat muun muassa toimittaja- ja tuote-nimiset taulukot. Seuraavaksi tieto-osat muunnettiin sarakkeiksi eli päätettiin, mitkä tiedot tallennetaan kuhunkin taulukkoon. Toimittaja-taulukko sisältää esimerkiksi kentät Yritys ja Osoite. Kentät näytetään taulukossa sarakkeina. Seuraavassa vaiheessa jokaiselle taulukolle määriteltiin perusavain, joka yksilöi jokaisen taulukon rivin. Jokaisen taulukon perusavaimeksi valittiin ID-kenttä, jotta tietokanta olisi mahdollisimman yksinkertainen. Taulukoiden välisten yhteyksien määrittämisessä eri taulukoiden tietoja liitettiin toisiinsa. Lopuksi tutkittiin rakenteen mahdolliset virheet tarkistamalla, saadaanko taulukoista haluttuja tuloksia. /11/

Tietokantaa suunniteltaessa päädyttiin erilliseen yksikkö-tilaukseen. Yksiköt syötetään omaan taulukkoonsa ennen tuotteiden tietojen syöttöä, jolloin tuotteelle valitaan yksikkö valmiista vaihtoehdoista. Näin kaikkien tuotteiden yksiköt ovat samanlaisia kirjoitusasultaan. Myös toimittajien ja tilaajien yhteyshenkilöille päätettiin luoda omat taulukonsa. Tilaajan yhteyshenkilö saadaan näin helposti liitettyä tarvittaessa moneen projektiin. Toimittajan yhteyshenkilöitä ei sidota yksittäisiin tuotteisiin, mutta toimittajalla voi silti olla monta yhteyshenkilöä, joten erillinen toimittajayhteyshenkilö-tilaukko todettiin tietokantaa suunniteltaessa tarpeelliseksi.

Tietokannan kokoon vaikutti neljä suunnitteluvaiheen päätöstä. Tietokannan kattavuus riippui siitä, mitä asioita tietokantaan päätettiin ottaa mukaan. Laajuuteen vaikutti se, mitä tietoja kustakin asiasta otettiin mukaan. Tietokannan karkeisuudella tarkoitetaan sitä yksityiskohtaisuuden tasoa, jolla tiedot tallennetaan tietokantaan. Laajuuteen vaikuttaa myös tietojen säilytysaika. /12, s. 90-91/

4.2 Käyttöliittymän suunnitteleminen

Käyttöliittymän suunnittelussa oli huomioitava monia seikkoja visuaalisen suunnittelun lisäksi. Ensin mietittiin käyttöliittymän käyttötarkoitus, joka voi olla esimerkiksi opetusohjelma tai mainos. Tämän käyttöliittymän tarkoitus on raportin luominen. Työn ongelma ja ratkaisutavat määriteltiin eli pohdittiin, mikä ongelma käyttöliittymällä ratkaistaan ja miten. Ongelma rajattiin ja jaettiin pienempiin osakokonaisuuksiin. Seuraavaan vaiheeseen kuului käyttöliittymän kohderyhmän sekä sen taustan määrittely. Tämän käyttöliittymän kohderyhmään kuuluu vain yksi ihminen, joten käyttöliittymä vastaa täsmälleen hänen tarpeitaan. Käyttöliittymän suunnittelun vaiheisiin kuului myös kustannusten ja aikataulun määrittely. Kustannuksia ei tämän käyttöliittymän teosta syntynyt, ja aikataulu sovittiin työn tilaajan kanssa joustavaksi. Käyttöliittymän suunnittelun viimeinen vaihe oli parhaiten työhön soveltuvan laitteiston ja ohjelmiston määrittely. Pohdinnan jälkeen käyttöliittymä päätettiin toteuttaa Visual Studio 2008 -ohjelmalla ja Visual Basic 2008 -ohjelmointikielellä. /13/

Käyttöliittymän suunnittelussa pyrittiin selkeyteen ja käyttäjäystävällisyyteen. Tämä toteutui esimerkiksi yhdistämällä kaksi eri toimintoa samalle lomakkeelle. Tilaaja- ja

toimittajataulukoissa on samanlaiset ja -nimiset kentät, joten tilaajan ja toimittajan tiedot suunniteltiin syötettäväksi samalla lomakkeella. Lomakkeelle tultaessa valitaan ensin, syötetäänkö tilaajan vai toimittajan tiedot, mutta tietojen syöttöä varten luodut kentät säilyvät samoina. Samanlainen toteutus suunniteltiin ”Lisää yhteyshenkilö”, ”Tarkastele ja muokkaa tilaajien tai toimittajien tietoja” ja ”Tarkastele ja muokkaa yhteys henkilöiden tietoja” -lomakkeille. Tämän lisäksi tilaajan tai toimittajan sekä yhteys henkilöiden tietojen lisäykseen varten luoduille lomakkeille suunniteltiin tilamuuttuja-toiminto. Se ennalta ehkäisee käyttäjän mahdollisia virheitä tietojen tallennuksessa. Tilamuuttuja-toiminto esitellään osiossa 5.3.3 ”Tilamuuttuja-toiminto”.

Käyttöliittymä suunniteltiin siten, että ensin syötetään tilaajien, toimittajien, tuotteiden ja projektien tiedot tietokantaan, ja tämän jälkeen tuotteita lisätään projekteihin. Tuotteiden lisäys projektiin haluttiin säilyttää yksinkertaisena ja vaivattomana. Tämän vuoksi päätettiin, että ”Lisää tuotteita projektiin” -lomakkeelle päästään sekä etusivulta että suoraan ”Lisää projekti” -lomakkeelta. Toimintoa suunniteltaessa päädyttiin ratkaisuun, jossa tuotteet lisätään tuoteriviä klikkaamalla ja syöttämällä tuotteen lukumäärä. Lomake esitellään yksityiskohtaisesti osiossa 6.3.6 ”Tuotteiden lisääminen projektiin”.

Tietojen tarkastelua ja muokkausta varten tarvittavien lomakkeiden suunnittelussa haluttiin säilyttää toimintojen yhdenmukaisuus. Siksi päätettiin, että kaikkien tietojen tarkastelu ja muokkaus suoritetaan omilla lomakkeillaan DataGridView-objekteissa. Projektitietojen kohdalla tiedot on tallennettu moneen eri taulukkoon, jolloin tietojen muokkaus ei onnistu samoin, kuin muilla käyttöliittymän lomakkeilla. Projektin tietojen muokkaus suunniteltiin erilliselle lomakkeelle tekstikenttien avulla. Toiminto esitellään osiossa 6.4.4 ”Projektien tarkasteleminen ja muokkaaminen”.

Kustannusraporttia suunniteltaessa pohdittiin, miten siinä näkyvät tiedot on yksinkertaisinta ja loogisinta esittää. Kaikki halutut tiedot oli löydyttävä raportista, ja myös raportin vaihteleva pituus oli huomioitava. Raportti oli suunniteltava muotoon, jollaisena se on helppo tulostaa, ja siksi raportti esitetään heti lomakkeelle tultaessa tulostusasettelussa. Kustannusraportti esitellään osiossa 6.6 ”Kustannusraportti” ja liitteessä 1.

Käyttöliittymän visuaalinen suunnittelu koostui objektien sommittelusta ja asettelusta kuvaruudulla. Suunnitteluun oli käytettävä aikaa jo etukäteen, ja kohderyhmä oli tässä

vaiheessa hyvä olla selvillä. Tärkeitä aiheita visuaalisessa suunnittelussa olivat kokonaisuus ja luovuus sekä sisältö ja toimivuus. Käyttäjällä tulisi olla esillä kaikki vaihtoehdot, jotka hän voi kulloinkin valita. Näin käyttöliittymä on käyttäjäystävällinen eikä toimintoja tarvitse muistaa ulkoa. Tässä käyttöliittymässä päädyttiin aloitussivulla perinteisen valikon sijasta painonappeihin, jotka ovat aina näkyvillä ja aktiivisia. Jokainen toiminto on oltava myös helposti peruutettavissa. Käyttöliittymän käyttö opetellaan keilemällä ja tutkimalla, joten kaikki kriittiset toiminnot on voitava peruuttaa. Jos jokin toiminto on sellainen, ettei sitä voi peruuttaa, pitää asia tuoda selkeästi esiin. Käyttöliittymän suunnittelussa oli myös huomioitava laitteet. Toiminnot on voitava valita sekä hiiren että näppäimistön avulla, kuten tässä käyttöliittymässä. /13/

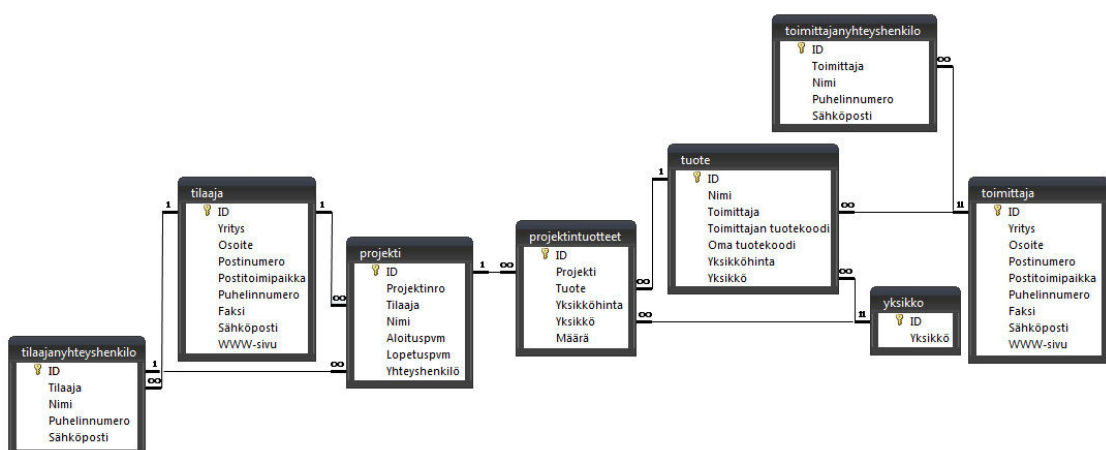
Pääosa ohjelman toiminnallisuudesta oli hyvä olla selvillä ennen käyttöliittymän yksityiskohtaista suunnittelua. Käyttöliittymä suunniteltiin teknisten ratkaisujen asettamilla ehdoilla, kun käyttöliittymän suunnittelu liitettiin osaksi toteutusvaihetta. Tässä vaiheessa ohjelmiston kokonaisuus oli jo hahmottunut. /14, s. 47/

5 TOTEUTUS

Ohjelman toteutus koostui kahdesta osasta: tietokannan toteutuksesta ja käyttöliittymän toteutuksesta. Ohjelman toteutusta edeltää ohjelman osien suunnittelu. Tietokanta toteutettiin Office Access 2007 -ohjelmalla ja sen jälkeen käyttöliittymä Visual Studio 2008 -ohjelmalla. Ohjelman toteutus oli aikaa vievin ja haasteellisin osuus tässä opinnäytetyössä. Ohjelma toteutettiin syksyn 2009 ja kevään 2009 aikana.

5.1 Tietokannan toteuttaminen

Tietokanta toteutettiin Office Access 2007 -ohjelmalla. Taulukot ja kentät luotiin, kuten oli suunniteltu. Tietokanta haluttiin säilyttää yksinkertaisena, mutta kuitenkin sellaisena, että kaikki tarvittavat tiedot on esitetty. Tietokannan toteutukseen kuului taulukoiden luonnin lisäksi myös kahden kyselyn luominen. Valmiiden taulukoiden välille luotiin tarvittavat yhteydet.



Kuva 3. Tietokannan taulukoiden väliset yhteydet

Tietokannassa on yhteensä 8 taulukkoa: tilaajanyhteyshenkilö-, tilaaja-, projekti-, projektintuotteet-, tuote-, toimittajanyhteyshenkilö-, yksikkö- ja toimittaja-taulukko. Kaikkien taulukoiden nimet on kirjoitettu pienellä alkukirjaimella ja kaikkien kenttien nimet isolla alkukirjaimella. Kaikissa taulukoissa yleisavaimena on ID-kenttä.

Tilaaja- ja toimittaja-taulukot sisältävät samanlaiset ja -nimiset kentät. Taulukoihin talletetaan tilaajan tai toimittajan yritys, osoite, postinumero, postitoimipaikka, puhelinnumero, faksi, sähköpostiosoite sekä WWW-sivu. Tilaaja-taulukolla on yhteys kahteen tauluun: sen ID-kenttä on linkitetty tilaajanyhteyshenkilö-taulukon Tilaaja-kenttään sekä projekti-taulukon Tilaaja-kenttään.

Tilaajanyhteyshenkilö- ja toimittajanyhteyshenkilö-taulukot ovat samanlaiset, ja niissä on viisi kenttää. ID-kentän lisäksi yhteyshenkilö-taulukoihin talletetaan yhteyshenkilön nimi, puhelinnumero ja sähköpostiosoite. Tilaaja-taulukon ID on linkitetty tilaajanyhteyshenkilö-taulukon Tilaaja-kenttään ja tilaajanyhteyshenkilö-taulukon ID-kenttä projekti-taulukon Yhteyshenkilö-kenttään.

Tuote-taulukko sisältää kaikki tiedot tuotteista. Taulukossa on ID-kentän lisäksi kuusi kenttää: Nimi, Toimittaja, Toimittajan tuotekoodi, Oma tuotekoodi, Yksikköhinta ja Yksikkö. ID-kenttä on linkitetty projektintuotteet-taulukon Tuote-kenttään, Toimittajakenttä toimittaja-taulukon ID-kenttään ja Yksikkö-kenttä yksikkö-taulukon ID-kenttään.

Yksikkö-taulukossa on ID-kentän lisäksi vain Yksikkö-kenttä. ID-kenttä on linkitetty sekä projektintuotteet-taulukon että tuote-taulukon Yksikkö-kenttiin.

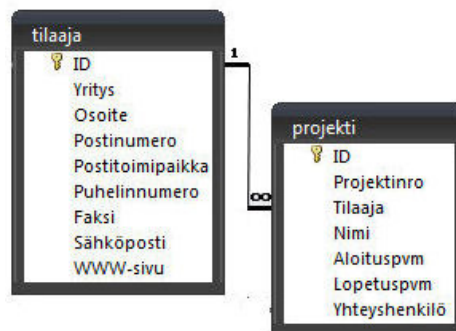
Projekti-taulukossa on ID-kentän lisäksi kuusi muuta kenttää: Projektinumero-, Tilaaja-, Nimi-, Aloituspvm-, Lopetuspvm- ja Yhteyshenkilö-kentät. Projektille määritellään siis projektinumero, projektin tilaajayritys, tilaajayrityksen yhteyshenkilö, projektin nimi sekä aloitus- ja lopetuspäivämäärät. Projekti-taulukon ID-kenttä on linkitetty projektintuotteet-taulukon Projekti-kenttään, Tilaaja-kenttä tilaaja-taulukon ID-kenttään ja Yhteyshenkilö-kenttä tilaajanyhteyshenkilö-taulukon ID-kenttään.

Projektintuotteet-taulukkoon talletetaan kuhunkin projektiin sisältyvät tuotteet. Taulukossa on kuusi kenttää: ID, Projekti, Tuote, Yksikköhinta, Yksikkö ja Määrä. Projekti-

kenttä on linkitetty projekti-taulukon ID-kenttään, Tuote-kenttä tuote-taulukon ID-kenttään ja Yksikkö-kenttä yksikkö-taulukon ID-kenttään.

5.1.1 Yhteydet

Yhteyksillä tarkoitetaan kahden eri taulukon tiettyjen kenttien välistä liitosta. Yhdessä taulukossa oleva tieto voidaan siis linkittää toisessa taulukossa olevaan tietoon. Yhteyksiä on kolmea lajia: yksi-moneen-yhteys, monta-moneen-yhteys tai yksi-yhteen-yhteys. Tässä tietokannassa on ainoastaan yksi-moneen-yhteyksiä, jollainen on esimerkiksi tilaaja-taulukon ID-kentän ja projekti-taulukon Tilaaja-kentän välillä. Yksi tilaaja voi siis tilata monta projektia, kun taas jokaisella projektilla voi olla kerrallaan vain yksi tilaaja. /1, s. 107/



Kuva 4. Yksi-moneen-yhteys

5.1.2 Kyselyt

Kysely on toiminto, jolla pystytään palauttamaan halutut tiedot halutuista taulukosta. Kyselyssä voidaan käyttää myös funktioita suorittamaan laskutoimituksia, kuten summa tai keskiarvo, taulukoiden tiedoista. Yleisimpiä kyselytyyppejä on kuusi: valinta-, kaksoisarvojen etsimis-, vastineettomien tietueiden etsimis-, parametri-, ristiintaulukointi- ja määrittäykysely. Tässä työssä tehtiin kaksi valintakyselyä. /1, s. 160-161/

Valintakysely on yleisin kyselytyyppi. Se noutaa tietoa yhdestä tai useammasta taulukosta ja myös näyttää tulokset taulukossa. Valintakyselyä voidaan käyttää tietueiden

ryhmittelyyn ja kokonaisarvojen laskemiseen. Access sisältää monia ohjattuja toimintoja, joiden avulla voidaan luoda kyselyjä, mutta kyselyjen luonti onnistuu myös käsin. /1, s. 160-161/



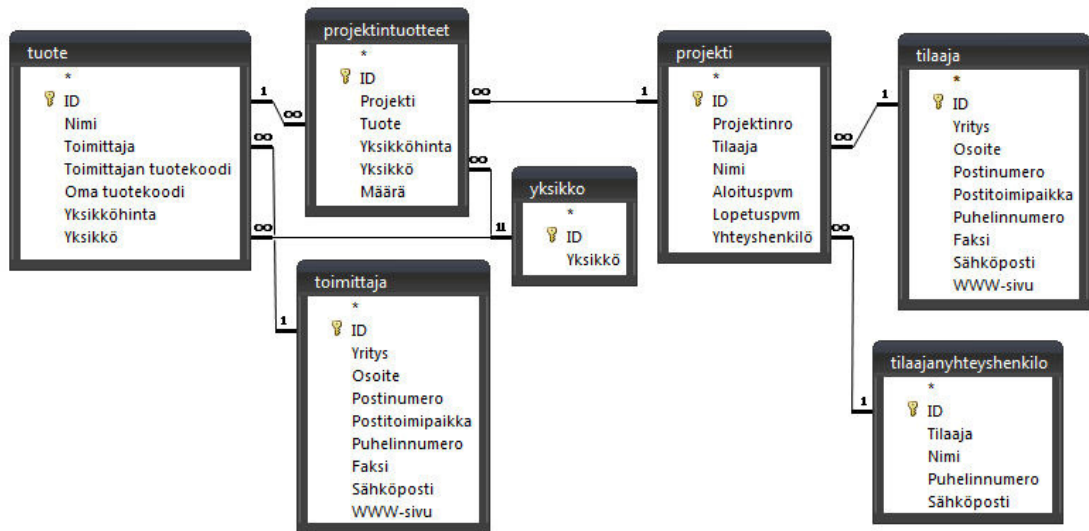
Kuva 5. Tuotekyselyssä käytetyt taulukot

Tuotekyselyä käytetään ohjelman ”Lisää tuotteita projektiin” -lomakkeessa. Kyselyn avulla saadaan esitettyä tuotteiden kaikki tiedot lomakkeen DataGridView-objektissa. Tuotekysely tehtiin, koska pelkän tuote-taulukon tiedot eivät olisi riittäneet kertomaan tuotteen kaikkia tietoja. Toimittaja-kenttään on talletettu toimittaja-taulun ID-kentän arvo, joka on siis luku, eikä näin ollen riitä kertomaan ohjelman käyttäjälle, kuka toimittaja on. Toimittajan nimi haetaan siis toimittaja-taulukon Yritys-kentästä. Oikea tietue löydetään ID-kentän avulla. Sama toiminto toistuu myös tuote-taulun Yksikkö-kentän kohdalla.

tuote.ID	Nimi	Toimittaja	Toimittajan tuotekoodi	Oma tuotekoodi	Yksikköhinta	Yritys	Yksikkö	yksikko.ID
40	Poljinämpäri 20L	26	364167		37,50 €	Bos Oy	€/kpl	15
42	Suodatinyksikkö PASU	24	PASU3		830,95 €	Etola Oy	€/kpl	15
43	Suodatinvahti	24	PASU3SUODATIN		333,75 €	Etola Oy	€/kpl	15
44	Nokkakärä 8,5kg	25	97157		141,00 €	Auramo Trukit	€/kpl	15

Kuva 6. Esimerkki tuotekyselyn tuloksista

Kuvassa 6 esitetään tuotekyselyn tuloksista neljä tietuetta. Kyselyn oleellisia tietoja ovat Nimi, Toimittajan tuotekoodi, Oma tuotekoodi, Yksikköhinta, Yritys ja Yksikkö.



Kuva 7. Kustannuskyselyssä käytetyt taulukot

Kustannuskyselyä käytetään kustannusraportin luomisessa. Kustannuskysely on monimutkaisempi, sillä kustannusraportissa tarvitaan lähes kaikki tiedot, mitä tietokantaan on syötetty. Kaikki muut taulukot, paitsi toimittajanyhteyshenkilö-taulukko, sisältävät tietoa, jota tarvitaan raportissa. Kustannuskyselyn avulla tehtiin myös matemaattisia toimenpiteitä, joiden tulokset tarvitaan raportissa. Esimerkiksi tiettyjen tuotteiden kapalemäärä piti kertoa yksikköhinnalla, jotta saadaan tietyn tuotteen kustannukset. Nämä eritellyt tuotekustannukset laskettiin yhteen, jotta saatiin selville koko projektin kokonaiskustannus.

5.2 Yhteyden luominen tietokannan ja käyttöliittymän välille

Yhteys tietokantaan muodostettiin automaattisella Data Source Configuration Wizard-toiminnolla. Data Source Configuration Wizard on Visual Studio 2008 -ominaisuus, jonka avulla ohjelman tietokannan tiedot vastaanotetaan automaattisesti. Toiminnossa määriteltiin tietokannan tyyppi, jonka jälkeen muodostettiin yhteys tietokantaan ja luotiin ohjelmaan tietojoukko. Tietojoukkoon määriteltiin tietokannan halutut taulut ja kentät. Lopuksi toiminto aukaisi Data Sources -ikkunan, jossa esitetään tietokannan objektit, jotka ovat ohjelman käytettävissä. /9, s. 440-441/

5.3 Käyttöliittymän toteuttaminen

Tässä osiossa esitellään ohjelman käyttöliittymän toteutuksen tärkeimmät kohdat sekä ohjelmakoodin että kuvien avulla. Kaikkia toimintoja ei ole esitelty yksityiskohtaisesti, vain olennaiset ja mainitsemisenarvoiset. Tässä osiossa esiteltyjä ominaisuuksia käytetään ohjelmassa toistuvasti. Mahdolliset poikkeukset ja erikoisominaisuudet esitellään seuraavassa luvussa 6 ”Tulokset” kuhunkin ominaisuuteen liittyvän lomakkeen esittelyn yhteydessä.

Osiassa esiteltyt koodit on tiivistetty niin, että kyseisen toiminnon kannalta tärkeät kohdat ovat esillä ja esittelyn ominaisuuden kannalta vähemmän tärkeä koodi on korvattu kommentilla.

Edellä olevassa taulukossa 1 esitetään komponentit, joita käytettiin ohjelmassa. Taulukossa esitetään vain ne komponentit englanniksi ja suomeksi, jotka esiintyvät ”Toteutus”-luvun esimerkkikoodeissa. Komponentit on uudelleennimetty koodissa niin, että jokaisella komponentilla on oma kolmikirjaiminen lyhenteensä.

Taulukko 1. Esimerkkikoodeissa käytetyt komponentit

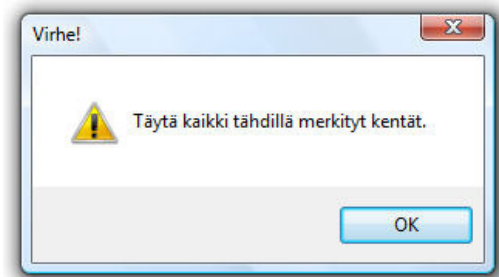
Nimi englanniksi	Nimi suomeksi	Koodissa käytetty nimi
Label	otsikko	lblNimi
ComboBox	vetovalikko	cbbNimi
Button	painike	cmbNimi
RadioButton	valintanappi	rdbNimi
TextBox	tekstikenttä	txtNimi

5.3.1 Virheentarkistus

Käyttöliittymän tekstikentissä on käytössä virheentarkistus, jonka avulla varmistetaan, että käyttäjä syöttää tiedon kaikkiin pakollisiin kenttiin. Jos pakollinen kenttä jätetään tyhjäksi ja tieto yritetään tallettaa, ruutuun ilmestyy virheilmoitus, joka kehottaa käyttäjää täyttämään kaikki tähdillä merkityt kentät. Lomakkeen pakolliset kentät on merkitty tähdellä, kun taas kentässä, jonka tieto ei ole välttämätön, tähteä ei ole.

Oma tuotekoodi	<input type="text"/>
Yksikköhinta *	<input type="text"/>

Kuva 8. Kentät



Kuva 9. Virheilmoitus

Seuraavaksi esitellään virheentarkistus-ohjelmakoodi. Esimerkkikoodi tarkistaa, ovatko tekstikentät txtNimi, txtOsoite tai txtPuhelin tyhjiä. Jos jokin kysesistä teksikentistä on tyhjä, virheilmoitus ilmestyy ruutuun. Jos kaikissa kentissä on tekstiä, virheilmoitusta ei tule.

```
If txtNimi.Text = "" Or txtOsoite.Text = "" Or txtPuhelin.Text = ""
Then
MsgBox("Täytä kaikki tähdillä merkityt kentät.",
MsgBoxStyle.Exclamation, "Virhe!")
```

5.3.2 Vetovalikkojen ominaisuudet

Ohjelman käyttöliittymässä on useita ComboBoxoja eli vetovalikoita. Vetovalikosta valitaan haluttu vaihtoehto napsauttamalla sen oikeassa reunassa olevaa mustaa nuolta.

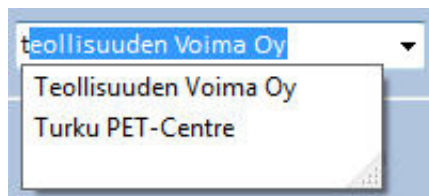


Kuva 10. Vetovalikko

Osa ohjelman vetovalikoista on linkitetty toiseen vetovalikkoon. Kun vetovalikosta valitaan jokin arvo, vaihtuu toisen vetovalikon arvo sen mukaan, mitä ensimmäisestä vetovalikosta valittiin. Jos esimerkiksi toimittaja-vetovalikosta valitaan jokin arvo, toimittaja-vetovalikkoon linkitettyssä toimittajan yhteyshenkilö -vetovalikossa näytetään vain kyseisen toimittajan yhteyshenkilöt.

Osassa vetovalikoista on käytössä ominaisuus, joka toimii siten, että kun vetovalikkoon aletaan kirjoittaa tekstiä, syötetyllä kirjaimella tai kirjainyhdistelmällä alkavat vaihtoehdot näytetään vetovalikon alla. Tämä toiminto toteutettiin asettamalla vetovalikon omi-

naisuus `AutoCompleteMode` arvoon `SuggestAppend` ja ominaisuus `AutoCompleteSource` arvoon `ListItems`. Ominaisuutta ei voitu asettaa vetovalikoihin, jotka on linkitetty toiseen vetovalikkoon. Näiden vetovalikkojen ominaisuus `DropDownStyle` vaihdettiin arvoon `DropDownList`, jolloin vetovalikko on niin sanotusti ”lukittu”, eikä siihen voida syöttää tekstiä.

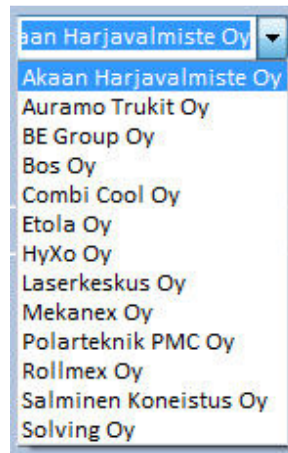


Kuva 11. Vetovalikkoon on syötetty t-kirjain



Kuva 12. ”Lukittu” vetovalikko

Kaikkien vetovalikkojen vaihtoehdot esitetään aakkosjärjestyksessä. Tämä helpottaa oikean vaihtoehdon löytämistä. Alla olevassa kuvassa 13 esitetään toimittaja-vetovalikko ja sen vaihtoehdot.



Kuva 13. Vaihtoehdot aakkosjärjestyksessä

Vaihtoehtojen aakkosjärjestys saatiin aikaan ohjelmakoodilla, joka on esitetty alla. Esimerkkikoodissa `Tilaaja-` tai `ToimittajaBindingSource`ä käyttävien valikoiden vaihtoehdot esitetään aakkosjärjestyksessä `Yritys-`kentän mukaan lajiteltuna. Koodi on kirjoitettava kyseessä olevan lomakkeen `Load-`osaan. Tässä esimerkissä oli kyse `NäytäTilaaja-Toimittaja-`lomakkeesta, jossa esitellään tilaajien ja toimittajien tiedot.

```

Private Sub NäytäTilaaajaToimittaja_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    TilaaajaBindingSource.Sort = "Yritys ASC"
    ToimittajaBindingSource.Sort = "Yritys ASC"
End Sub

```

5.3.3 Tilamuuttuja-toiminto

Useissa käyttöliittymän lomakkeiden Tallenna-painikkeissa on kaksi toimintoa. Tätä toimintoa kutsutaan tässä työssä tilamuuttuja-toiminnoksi. Kun painiketta napsautetaan ensimmäisen kerran, tekee se tietyn toiminnon, ja painikkeeseen vaihtuu eri teksti. Kun painiketta napsautetaan uudelleen, tekee se jonkin muun toiminnon ja teksti vaihtuu takaisin alkuperäiseksi. Otetaan esimerkiksi ”Lisää tilaaja tai toimittaja” -lomake. Lomakkeelle tullessa lomake näyttää tältä:

The screenshot shows a window titled "Lisää tilaaja tai toimittaja". At the top, there is a label "Valitse lisätäänkö tilaaja vai toimittaja." followed by two radio buttons: "Tilaaaja" (which is selected) and "Toimittaja". Below this is the instruction "Syötä tiedot ja paina 'Tallenna'.". There are eight text input fields, each with a label and an asterisk indicating it is required: "Yritys *", "Osoite *", "Postinumero *", "Postitoimipaikka *", "Puhelinnumero *", "Faksi", "Sähköposti", and "WWW-sivu". At the bottom of the window, there are two buttons: "Tallenna" on the left and "Takaisin" on the right.

Kuva 14. Lomake sivulle tullessa

Kaikki tekstikentät sekä Tallenna-painike ovat epäaktiivisia. Kun valintanapeista valitaan jompikumpi vaihtoehto, tekstikentät sekä Tallenna-painike aktivoituvat. Kun tekstikenttiin on syötetty halutut tiedot ja napsautetaan Tallenna-painiketta, näyttöön ilmestyy ilmoitus, joka kertoo, että tiedot tallennettiin tietokantaan. Ilmoitus kuitataan OK-painikkeella, jonka jälkeen lomake näyttää tältä:

Kuva 15. Lomake, kun Tallenna-painiketta on painettu kerran

Tallenna-painikkeeseen vaihtui teksti ”Luo uusi” ja sekä valintanapit että tekstikentät ovat epäaktiivisia. Kun ”Luo uusi” -painiketta napsautetaan, palaa lomake siihen muotoon, mitä se oli alussa. Tällöin ”Luo uusi” -painikkeen tekstiksi vaihtuu jälleen ”Tallenna”, valintanapeista voi valita, syötetäänkö tilaajan vai toimittajan tiedot ja tekstikentät tyhjenevät.

Seuraavaksi esitellään tilamuuttuja-toiminnon ohjelmakoodi TilaaajaToimittaja-lomakkeella. Muuttuja tilamuuttuja on esitelty heti ohjelmakoodin alussa, että se on koko lomakkeen käytettävissä. Muuttuja tilamuuttuja on alustettu Load-osassa arvoon 0. cmbTallenna on Tallenna-painike. Esimerkkikoodissa esitetään vain tilamuuttuja-toiminnon kannalta olennaiset asiat eli esimerkiksi virheentarkistuksen koodi on korvattu kommentilla.

```
Private Sub cmbTallenna_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmbTallenna.Click
If tilamuuttuja = 0 Then
    'virheentarkistus
    'luodaan uusi tietorivimuuttuja, joka tallennetaan joko
    tilaaja- tai toimittajataulukkoon, riippuen siitä, kumpi
    valintanappi valitaan
    If rdbTilaaaja.Checked = True Then
        'käyttäjä valitsi tilaaja-valintanapin, joten
        tiedot tallennetaan tilaaja-taulukkoon
```

```

ElseIf rdbToimittaja.Checked = True Then
    'käyttäjä valitsi toimittaja-valintanapin, joten
    tiedot tallennetaan toimittaja-taulukkaan
End If
'valintanapit ja tekstikentät asetetaan epäaktiivisiksi
cmbTallenna.Text = "&Luo uusi"
tilamuuttuja = 1
ElseIf tilamuuttuja = 1 Then
    'valintanapit aktivoidaan ja tekstikentät tyhjennetään
    cmbTallenna.Text = "&Tallenna"
    tilamuuttuja = 0
End If
End Sub

```

Tilamuuttuja-toiminto on käytössä lomakkeissa ”Lisää tilaaja tai toimittaja”, ”Lisää yhteyshenkilö”, ”Lisää tuote”, ”Lisää projekti” ja ”Lisää yksikkö”.

5.3.4 Tietojen tallentaminen tietokantaan

Käyttöliittymässä on useita lomakkeita, joiden avulla tietoja talletetaan tietokantaan. Tässä osiossa on käytetty esimerkkinä lomakkeen ”Lisää tilaaja tai toimittaja” koodia. Kyseisessä esimerkkikoodissa toimittajan tiedot tallennetaan toimittaja-taulukkaan.

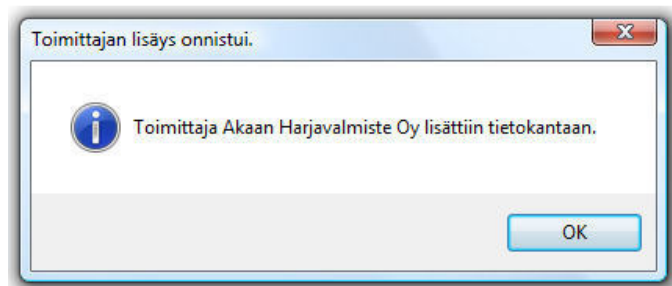
```

Dim uusi_tieto As DataRow
uusi_tieto = TietokantaDataSet.Tables("toimittaja").NewRow
uusi_tieto.Item("Yritys") = txtYritys.Text
uusi_tieto.Item("Osoite") = txtOsoite.Text
uusi_tieto.Item("Postinumero") = txtPostinumero.Text
uusi_tieto.Item("Postitoimipaikka") = txtPostitoimipaikka.Text
uusi_tieto.Item("Puhelinnumero") = txtPuhelin.Text
uusi_tieto.Item("Faksi") = txtFaksi.Text
uusi_tieto.Item("Sähköposti") = txtEmail.Text
uusi_tieto.Item("WWW-sivu") = txtWWW.Text
TietokantaDataSet.Tables("toimittaja").Rows.Add(uusi_tieto)
Me.Validate()
Me.ToimittajaBindingSource.EndEdit()
Me.ToimittajaTableAdapter.Update(Me.TietokantaDataSet.toimittaja)
Dim yritys As String
yritys = txtYritys.Text
MsgBox("Toimittaja " & yritys & " lisättiin tietokantaan.",
MsgBoxStyle.Information, "Toimittajan lisäys onnistui.")

```

Ensin luodaan uusi tietorivimuuttuja nimeltä uusi_tieto. Tämän jälkeen tehdään määrittely, jonka mukaan uusi_tieto on toimittaja-taulukon uusi rivi. Tämän jälkeen luodaan kaikki uusi_tieto-muuttujan tiedot. Tietorivimuuttuja lisätään toimittaja-taulukkaan ja tietokantaan. Tiedon kelpoisuus tarkistetaan ja tietojen muokkaus

lopetetaan. Lopuksi TableAdapter päivitetään ja luodaan ilmoitus, joka kertoo, että toimittajan tietojen lisäys onnistui.



Kuva 16. Toimittajan lisäys onnistui

5.3.5 Kustannusraportti

Page Header									
Kustannusraportti								="Sivu "	
Body									
Projektin nimi	Projektin numero	Tilaaaja			Aloituspv		Lopetuspv		
=Fields!projekti_Nimi.Valu	=First(Fields!Projektinro.)	=Fields!tilaaja_Yritys.Valu	=Format(First(Fields!Aloit	=Format(First(F					
Tuote	Toimittaja	Oma	Tomittajan	Määrä	Yksikköhinta	Yksikkö	Yhteensä:		
=Fields!tuote_Nimi.V	=Fields!toimit	=Fields!Oma	=Fields!Toimi	=Fields!Määrä	=Fields!Yksikkö	=Fields!Yksikkö	=Fields!yhteer	€	
							Yhteensä:	=sum(Fields	€
Page Footer									
								=format(now())	

Kuva 17. Kustannusraportti Report Designer -näkyssä

Kustannusraportti luotiin Visual Studio 2008:aan kuuluvalla Report Designer -toiminnolla. ReportViewer on työkalu, joka mahdollistaa raporttien tarkastelun. Raportin toiminnallisuuteen sisältyy myös ohjelmakoodia, joka esitellään tässä:

```

Public Class Kustannukset

Private Sub Kustannukset_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
Me.kustannuskyselyTableAdapter.Fill(Me.TietokantaDataSet.kustannuskysely, cbbProjektinnimi.SelectedValue)
Me.ReportViewer1.RefreshReport()
Me.ReportViewer1.SetDisplayMode(Microsoft.Reporting.WinForms.DisplayMode.PrintLayout)
End Sub

Private Sub cbbProjektinumero_SelectedValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cbbProjektinumero.SelectedValueChanged
Me.kustannuskyselyTableAdapter.Fill(Me.TietokantaDataSet.kustannuskysely, cbbProjektinumero.SelectedValue)
Me.ReportViewer1.RefreshReport()
End Sub

End Class

```

“Kustannukset”-lomakkeella on kaksi vetovalikkoa, projektin nimi - ja projektinumero-vetovalikot, joista voidaan valita, minkä projektin raportti näytetään ReportViewerissä. Vetovalikolla cbbProjektinnimi valitaan näytettävä projekti nimen mukaan ja vetovalikolla cbbProjektinumero projektinumeron mukaan. ReportViewer on päivitettävä aina lomakkeen Load-osassa, kun jommankumman vetovalikon arvo vaihtuu. ReportViewerin ominaisuuksia muokataan niin, että raportti näytetään tulostusmuodossa lomakkeelle tultaessa.

Kustannusraportin teossa käytettiin kustannuskyselyä. Kustannusraportin toteuttaminen esitellään tarkemmin osioissa 5.1.2 ”Kyselyt” ja 6.6 ”Kustannusraportti”.

5.3.6 Tietojen poistaminen

Ohjelmassa on tietoja, jotka voidaan poistaa suoraan, ja tietoja, joiden poisto edellyttää ensin niihin liittyvien tietojen poistoa. Projektin poistaminen ei edellytä siihen liittyvien tietojen poistoa etukäteen, vaan kun projekti poistetaan, poistetaan suoraan myös kaikki tuotteet, jotka liittyivät kyseiseen projektiin. Tuotteita ei poisteta tuotteet-tilukosta vaan projektintuotteet-tilukosta, eli tuotteet poistuvat vain tietyn projektin alta, eivätkä koko tietokannasta. Tässä esitellään ”Projektin poisto” -toiminnon ohjelmakoodi:

```

Private Sub cmbPoistaprojekti_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmbPoistaprojekti.Click

Dim nimi As String
Dim ID As Integer

ID = cbbProjekti.SelectedValue
nimi = cbbProjekti.Text

If MsgBox("Haluatko varmasti poistaa projektin " & nimi & "?" &
ControlChars.CrLf & "HUOM! Tämä toiminto poistaa samalla kaikki
projektin tiedot!", MsgBoxStyle.OkCancel, "Varmistus!") =
MsgBoxResult.Ok Then

    For Each kohderivi As DataRow In
TietokantaDataSet.Tables("projektintuotteet").Rows
        If kohderivi.Item("Projekti") = ID Then
            kohderivi.Delete()
        End If
    Next

    Me.Validate()
    Me.ProjektintuotteetBindingSource.EndEdit()
    Me.ProjektintuotteetTableAdapter.Update(Me.TietokantaDataSe
t.projektintuotteet)

    For Each kohderivi2 As DataRow In
TietokantaDataSet.Tables("projekti").Rows
        If kohderivi2.Item("ID") = ID Then
            kohderivi2.Delete()
        End If
    Next

    Me.Validate()
    Me.ProjektiBindingSource.EndEdit()

Me.ProjektiTableAdapter.Update(Me.TietokantaDataSet.projekti)

    MsgBox("Projekti " & nimi & " poistettiin tietokannasta.",
MsgBoxStyle.Exclamation, "Projektin poisto onnistui.")

End If
End Sub

```

Toisessa tapauksessa jotain tietoja on poistettava ennen, kuin halutun tiedon poisto onnistuu. Tällainen tilanne on esimerkiksi tilaajan poistossa. Ennen kuin tilaajan tiedot voidaan poistaa, on poistettava kaikki projektit, joissa kyseinen tilaaja on tilaajana. Kun tilaajan tiedot poistetaan, poistetaan samalla myös kaikki kyseisen tilaajan yhteyshenkilöt. Tässä esitellään ”Tilaajan poisto” -toiminnon ohjelmakoodi:

```

Private Sub cmbPoistatilaaja_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmbPoistatilaaja.Click
Dim nimi As String
Dim ID, apu As Integer
ID = cbbTilaaja.SelectedValue
nimi = cbbTilaaja.Text

```

```

apu = 0
For Each kohderivi3 As DataRow In
TietokantaDataSet.Tables("projekti").Rows
    If kohderivi3.Item("Tilaaaja") = ID Then
        apu = 1
    End If
Next
If apu = 0 Then
If MsgBox("Haluatko varmasti poistaa tilaajan " & nimi &
"?" & ControlChars.CrLf & "HUOM! Tämä toiminto poistaa
samalla kaikki tilaajan tiedot!", MsgBoxStyle.OkCancel,
"Varmistus!") = MsgBoxResult.Ok Then
    For Each kohderivi As DataRow In
TietokantaDataSet.Tables("tilaajanyhteyshenkilo")
.Rows
        If kohderivi.Item("Tilaaaja") = ID Then
            kohderivi.Delete()
        End If
    Next
    Me.Validate()
    Me.TilaajanyhteyshenkiloBindingSource.EndEdit()
    Me.TilaajanyhteyshenkiloTableAdapter.Update(Me.Ti
etokantaDataSet.tilaajanyhteyshenkilo)

    For Each kohderivi2 As DataRow In
TietokantaDataSet.Tables("tilaaja").Rows
        If kohderivi2.Item("ID") = ID Then
            kohderivi2.Delete()
        End If
    Next
    Me.Validate()
    Me.TilaajaBindingSource.EndEdit()
    Me.TilaajaTableAdapter.Update(Me.TietokantaDataSe
t.tilaaja)

    MsgBox("Tilaaja " & nimi & " poistettiin
tietokannasta.", MsgBoxStyle.Exclamation,
"Tilaajan poisto onnistui.")
End If
Else
    MsgBox("Poista ensin kaikki projektit, joiden tilaaja on "
& nimi & ".", MsgBoxStyle.Exclamation, "Virhe!")
End If
End Sub

```

”Poista tietoja” -lomake esitellään kokonaisuudessaan osiossa 6.5. ”Tietojen poistaminen”.

5.3.7 Projektinumero

Projektinumero on jokaisen projektin yksilöllinen numero. Se lisätään projektin tietoihin samanaikaisesti muiden tietojen kanssa ”Lisää projekti” -lomakkeella. Työssä esitel-

lyissä esimerkkiprojekteissa projektinumeron muodoksi on valittu 09001 siten, että 09 viittaa kuluvaan vuoteen ja kolmea viimeistä numeroa käytetään yksilöimään kukin projekti niin, että ensimmäisen projektin numero on 09001, seuraavan 09002 ja niin edelleen. Viishanke Oy:lla projektien vuotuinen määrä on korkeintaan 999. Viimeisin lisätty projektinnumero näytetään projektin tietojen lisäys lomakkeella. Viimeisimmän projektinumeron näyttämiseen luotiin seuraava ohjelmakoodi:

```
If lblProjektinnumero.Text <> "" Then
    pnro = lblProjektinnumero.Text + 1
    If pnro < 10000 Then
        txtProjektinnumero.Text = "0" & pnro.ToString()
    Else
        txtProjektinnumero.Text = pnro
    End If
End If
```

Edellinen projektinnumero näytetään lblProjektinnumero otsikko-objektissa. Kenttään, johon syötetään uusi projektinnumero, tuodaan valmiiksi edellistä projektinumeroa yhden arvon suurempi luku. Jos esimerkiksi lblProjektinnumero otsikko-objektissa edellisen projektinumeron arvo on 09001, tuodaan tekstikenttään txtProjektinnumero arvo 09002. Projektinumeroa voi silti halutessaan itse muuttaa. Etunolla tuodaan kaikkiin projektinumeroihin, jotka ovat pienempiä kuin 10000, jotta projektinumeroiden järjestys säilyisi oikeana. Muuten projektinnumero 1000 menisi projektinumeron 9000 edelle.

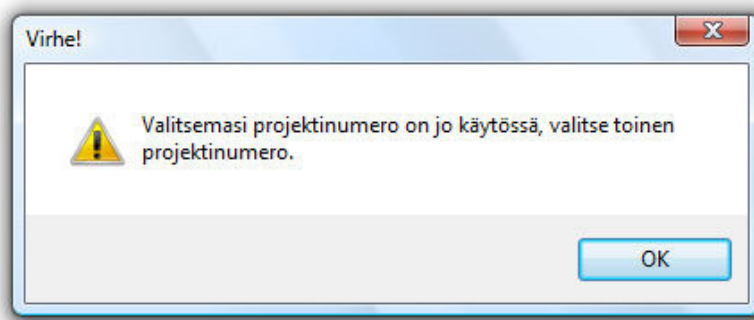
Projektin tietojen syötössä virheentarkistusta on laajennettu siten, että ohjelma tarkistaa, onko käyttäjän syöttämä projektinnumero jo entuudestaan käytössä jollain toisella projektilla. Näin varmistetaan, että jokaisella projektilla on yksilöllinen projektinnumero. Edellä esitetään virheentarkistuksen ohjelmakoodi:

```
Dim apu As Integer
apu = 0
For Each kohderivi As DataRow In
TietokantaDataSet.Tables("projekti").Rows
    If kohderivi.Item("Projektinro") = txtProjektinnumero.Text
    Then
        apu = 1
    End If
Next
If txtNimi.Text = "" Or txtProjektinnumero.Text = "" Or
txtAloituspvm.Text = "" Then
    MsgBox("Syötä kaikki tähdillä merkityt tiedot.",
    MsgBoxStyle.Exclamation, "Virhe!")
```

```

ElseIf apu = 1 Then
MsgBox("Valitsemasi projektinnumero on jo käytössä, valitse
toinen projektinnumero.", MsgBoxStyle.Exclamation, "Virhe!")

```



Kuva 18. Valittu projektinnumero on jo käytössä

5.3.8 Projektin aloitus- ja lopetuspäivämäärä-kentät

”Lisää projekti” -lomakkeen ”Projektin aloituspäivämäärä” - ja ”Projektin lopetuspäivämäärä” -kenttiin on ohjelmoitu toiminto, joka varmistaa sen, että kenttiin on kirjoitettu päivämäärä oikeassa muodossa. Kentistä ei pääse ulos, ennen kuin niihin on syötetty oikeanmuotoinen päivämäärä. Kyseinen toiminto toteutettiin ”Projektin aloituspäivämäärä” -kentän kohdalla seuraavalla ohjelmakoodilla:

```

Private Sub txtAloituspvm_Validating(ByVal sender As System.Object,
ByVal e As System.ComponentModel.CancelEventArgs) Handles
txtAloituspvm.Validating
    If txtAloituspvm.Text <> "" Then
        Try
            Dim paiva As Date = txtAloituspvm.Text

            Catch ex As Exception
                e.Cancel = True
                txtAloituspvm.SelectAll()
            End Try
        End If
    End Sub

```

”Projektin lopetuspäivämäärä” -kentän toiminto toteutettiin vastaavalla koodilla. Jos kenttään on kirjoitettu jotain, tehdään tarkistus. Tarkistuksessa varmistetaan, että kenttään on kirjoitettu päivämäärä. Kentästä päästään vielä ulos, jos siihen ei ole kirjoitettu mitään. ”Lisää projekti” -lomake esitellään osiossa 6.3.4 ”Projektin lisääminen”.

6 TULOKSET

Tässä osiossa esitellään valmis ohjelma ja sen toiminnot lomake kerrallaan. Osiossa 5.3. ”Käyttöliittymän toteutus” käytiin läpi ohjelman toteutuksen kannalta tärkeät perusominaisuudet, joita ei enää kerrata tässä osiossa.

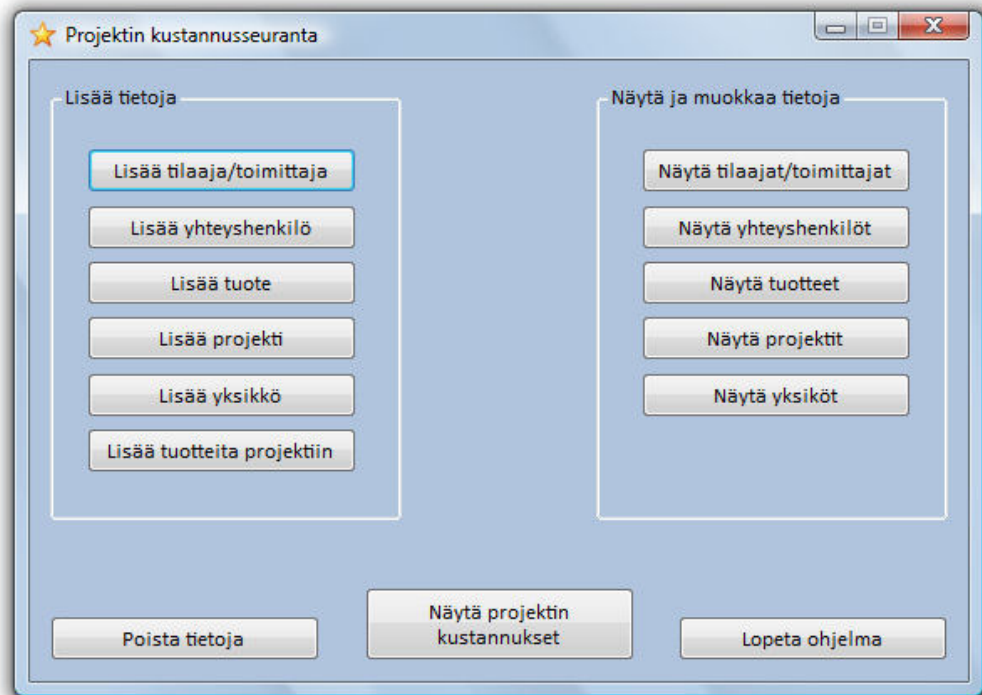
6.1 Ohjelman yleiset ominaisuudet

Jokainen lomake aukeaa keskelle näyttöä. Lomakkeet voidaan sulkea oikean yläkulman X:stä. Etusivun X:ää napsauttamalla koko ohjelma sulkeutuu, kun taas muissa lomakkeissa X:n napsautus sulkee kyseisen lomakkeen ja ohjelma palaa etusivulle. Lomakkeiden kokoa ei voi muuttaa. Useimpia painikkeita voi käyttää myös valintakirjaimen avulla, esimerkiksi jokaiselta lomakkeelta, etusivua lukuun ottamatta, löytyvä Tallenna-painike toimii valintakirjaimesta T ja Takaisin-painike valintakirjaimesta A. Kaikissa lomakkeissa taustaväri on LightSteelBlue ja fonttina Calibri.

Takaisin-painiketta voi painaa milloin vain, esimerkiksi kesken tietojen syötön. Tällöin palataan takaisin edelliselle lomakkeelle, eivätkä tiedot tallennu tietokantaan.

DataGridView-objektien joka toinen rivi on erivärinen, jotta tietojen tarkastelu olisi selkeämpää.

6.2 Aloitus-/etusivu



Kuva 19. Ohjelman aloitussivu

Ohjelman aloitussivulla on kuusi painiketta tietojen lisäämiseen sekä viisi painiketta tietojen tarkastelua ja muokkaamista varten. Aloitussivulla on myös painike tietojen poistamista varten, projektin kustannusten tarkastelua varten sekä ”Lopeta ohjelma” -painike. Osioissa 6.3.1-6.3.6, 6.4.1-6.4.5 sekä 6.5 ja 6.6 esitellään lomakkeet, jotka aukeavat etusivun painikkeita napsauttamalla.

6.3 Tietojen lisääminen

Ohjelmassa on kuusi eri lomaketta tietojen lisäämistä tietokantaan varten. Lomakkeen kenttiin syötetyt tiedot tallennetaan tietokantaan napsauttamalla Tallenna-painiketta. Aina tietojen lisäämisen onnistuessa näytölle aukeaa ”Tiedon lisäys onnistui” -ilmoitus, joka kertoo siitä, että tieto tallennettiin oikein ja mikä tieto lisättiin tietokantaan. Tietojen tallennus onnistuu vain, jos kaikki pakolliset tähdellä merkityt tiedot on syötetty kenttiin. ”Tiedon lisäys onnistui” -ilmoitus kuitataan painamalla OK-painiketta.

Täysin tyhjää tietokantaa täyttäessä tietojen tallennus on hyvä suorittaa tietyssä järjestyksessä. Ensin syötetään tarvittavien toimittajien ja tilaajien tiedot. Vasta tämän jälkeen toimittajille ja tilaajille lisätään yhteyshenkilöt. Ennen tuotteiden lisäämistä tietokannassa on oltava toimittajien tiedot sekä tuotteiden yksiköt. Projektiin voidaan luonnollisesti lisätä tuotteita vasta kun itse projekti ja sen tuotteet on luotu.

Tässä osiossa esiteltiin ”tietojen lisäys” -lomakkeiden yleiset ominaisuudet eli kaikille näille lomakkeille yhteiset piirteet. Osioissa 6.3.1-6.3.6 esitellään jokaisen lomakkeen ominaispiirteet eikä kaikille lomakkeille yhteisiä ominaisuuksia enää mainita.

6.3.1 Tilaajien ja toimittajien lisääminen

Lisää tilaaja tai toimittaja.

Valitse lisätäkö tilaaja vai toimittaja.

Tilaaja

Toimittaja

Syötä tiedot ja paina "Tallenna".

Yritys *

Osoite *

Postinumero *

Postitoimipaikka *

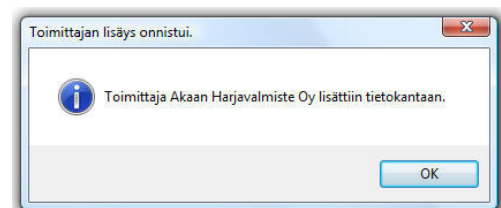
Puhelinnumero *

Faksi

Sähköposti

WWW-sivu

Tallenna Takaisin

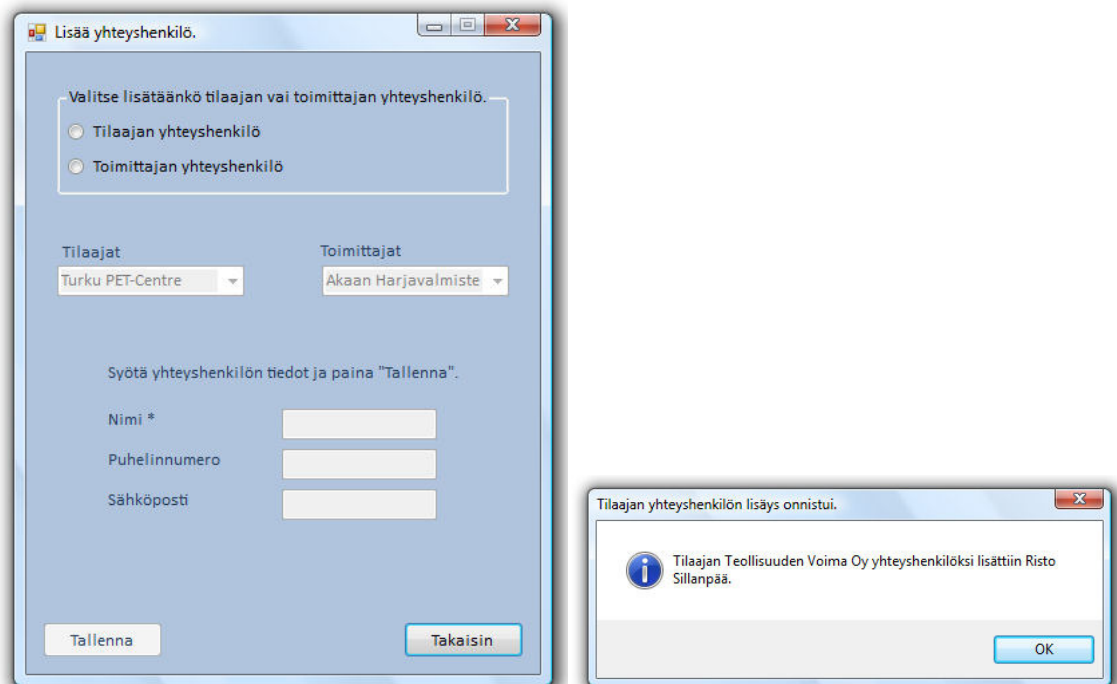


Kuva 20. Tilaajan tai toimittajan lisääminen Kuva 21. Toimittajan lisäys onnistui

”Tilaajien tai toimittajien lisäys” -lomakkeessa on sille tullessa aktiivisina ainoastaan kaksi RadioButtonia eli valintanappia sekä Takaisin-painike. Ensin valitaan, halutaanko lisätä tilaajan vai toimittajan tiedot napsauttamalla valintanappia. Näin kaksi toimintoa on saatu yhdistettyä yhdelle lomakkeelle. Kun jompikumpi vaihtoehto valitaan, kentät aktivoituvat tietojen syöttöä varten ja Tallenna-painike tietojen tallennusta varten.

Pakollisia tietoja ovat yrityksen nimi, osoite, postinumero, postitoimipaikka ja puhelinnumero. Faksi, sähköpostiosoite ja WWW-sivu ovat vapaaehtoisia tietoja jo siksi, että niitä ei kaikilta yrityksiltä löydy.

6.3.2 Yhteysenkilöiden lisääminen



Kuva 22. Yhteysenkilön lisääminen Kuva 23. Yhteysenkilön lisääminen onnistui

”Lisää yhteysenkilö” -lomakkeella lisätään tietokantaan joko tilaaajan tai toimittajan yhteysenkilön tiedot. ”Lisää tilaaja tai toimittaja” -lomakkeella syötetään ainoastaan yrityksen tiedot, kuten yrityksen nimi ja puhelinnumero. Tällä lomakkeella jokaiselle yritykselle yksilöidään tietty yhteysenkilö, kenen kanssa konkreettisesti ollaan yhteydessä.

”Lisää yhteysenkilö” -lomakkeella valitaan ensin valintanappien avulla, syötetäänkö tilaaajan vai toimittajan yhteysenkilön tiedot. Kentät ja Tallenna-painike aktivoituvat tietojen syöttöä varten vasta, kun valinta on suoritettu. Ainoa pakollinen tieto on yhteysenkilön nimi, sillä yhteysenkilön puhelinnumero voi olla myös sama kuin yrityksen puhelinnumero eikä kaikilla ole käytössään sähköpostitiliä.

6.3.3 Tuotteiden lisääminen

Lisää tuote.

Syötä tuotteen tiedot ja paina "Tallenna".

Nimi *

Toimittaja * Akaan Harjavalviste ▼

Toimittajan tuotekoodi

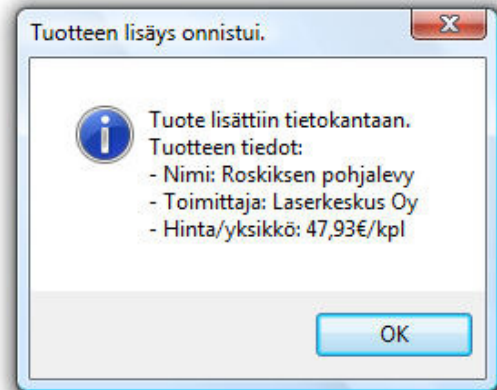
Oma tuotekoodi

Yksikköhinta *

Yksikkö * €/kpl ▼

Tallenna Takaisin

Kuva 24. Tuotteen lisääminen



Kuva 25. Tuotteen lisäys onnistui

”Lisää tuote” -lomakkeella tietokantaan syötetään tuotteiden tietoja. Ennen tuotteen tietojen lisäystä sen yksikkö sekä toimittaja pitää olla syötettynä tietokantaan. Tuotteen toimittaja ja yksikkö valitaan valmiista vaihtoehdoista vetovalikkojen avulla.

Tuotteen nimi, toimittaja, yksikköhinta sekä yksikkö ovat pakollisia tietoja, sillä nämä tiedot löytyvät jokaiselta tuotteelta. Yksikköhinta on pelkkä luku, esimerkiksi 47,93, ja vasta Yksikkö-kentästä valitaan yksikkö. Toimittajan tuotekoodi ja oma tuotekoodi ovat vapaaehtoisia tietoja.

6.3.4 Projektien lisääminen

Lisää projekti.

Syötä projektin tiedot ja paina "Tallenna".

Projektin nimi *

Projektinumero *

Edellisen projektin numero on 09005

Tilaaaja *

Aloituspäivämäärä *

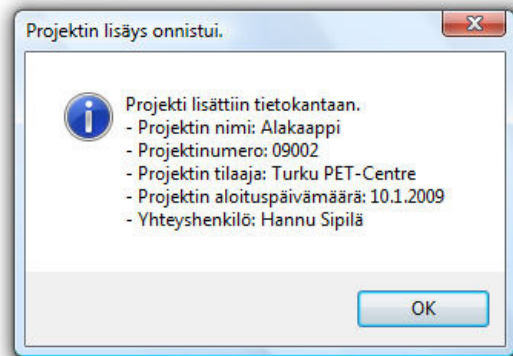
Lopetuspäivämäärä

Yhteyshenkilö *

Tallenna Takaisin

Lisää tuotteita projektiin

Kuva 26. Projektin lisääminen

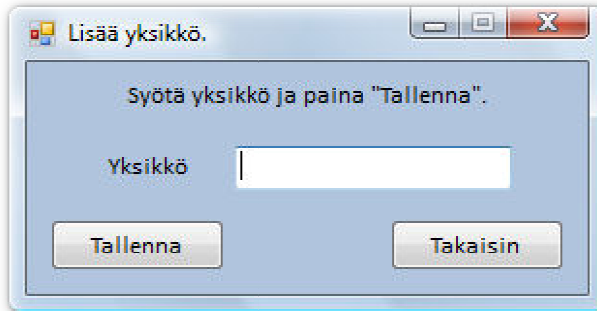


Kuva 27. Projektin lisäys onnistui

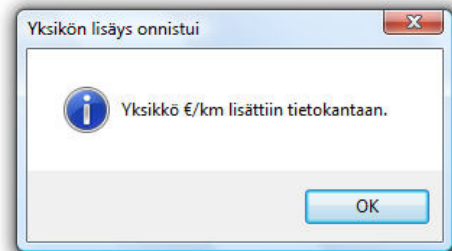
”Lisää projekti” -lomakkeella lisätään tilauksen eli projektin tiedot. Tilaajan ja tilaajan yhteyshenkilön tiedot on syötettävä ennen projektin tietojen syöttöä tietokantaan. Jokaisella projektilla on yksilöllinen projektinumero, sillä samannimisiä projekteja voi olla monta. Tässä esimerkissä on käytetty projektinumeroa 09002, joka viittaa siihen, että kyseessä on vuoden 2009 toinen projekti. Viimeisin projektinumero näkyy ”Lisää projekti” -lomakkeella. Jos yritetään käyttää jo olemassa olevaa projektinumeroa, ohjelma ilmoittaa, että projektinumero on jo käytössä. Projektinumeron ominaisuudet esitellään tarkemmin osiossa 5.3.7. ”Projektinumero”.

Projektin nimi, projektinumero, tilaaja, aloituspäivämäärä ja yhteyshenkilö ovat pakollisia tietoja. Projektin lopetuspäivämäärä ei ole pakollinen tieto, sillä aina projektin tietoja syötettäessä ei tiedetä, milloin projekti on valmis. Projektin lopetuspäivämäärän voi syöttää projektin tietoihin myös jälkikäteen. Tallenna-painiketta painaessa Lisää tuotteita projektiin -painike aktivoituu, jolloin aukeaa lomake, jossa voidaan lisätä tuotteita eri projekteihin. Kyseinen lomake esitellään osiossa 6.3.6 ”Tuotteiden lisääminen projektiin”. Tähän lomakkeeseen on painike myös etusivulla.

6.3.5 Yksiköiden lisääminen



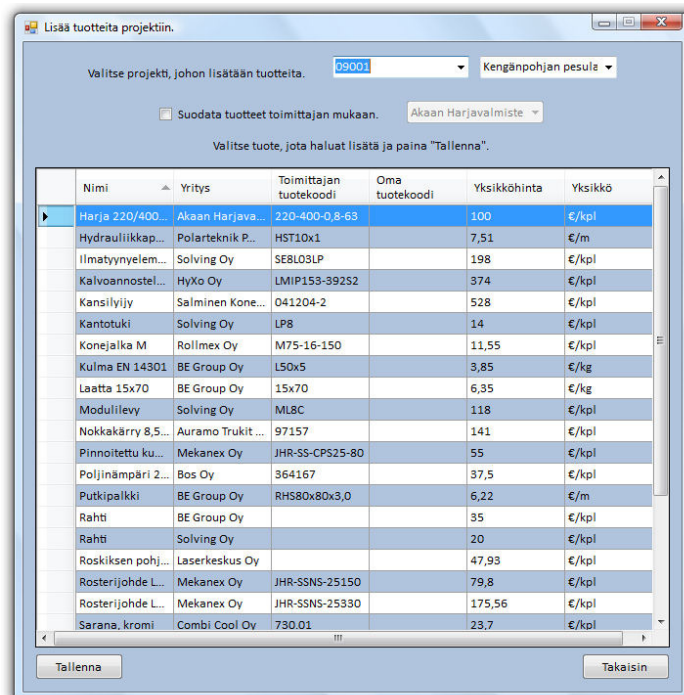
Kuva 28. Yksikön lisääminen



Kuva 29. Yksikön lisäys onnistui

”Lisää yksikkö” -lomakkeella Yksikkö-kenttään syötetään haluttu yksikkö. Yksikkö on muotoa valutta/yksikkö eli esimerkiksi €/km, kuten yllä olevassa kuvassa 29. Yksiköt on syötettävä tietokantaan ennen tuotteiden tietojen syöttämistä.

6.3.6 Tuotteiden lisääminen projektiin



Kuva 30. Tuotteiden lisääminen projektiin

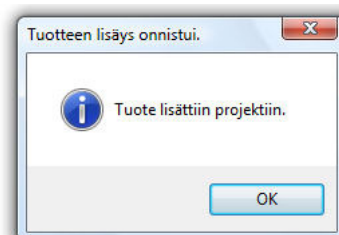
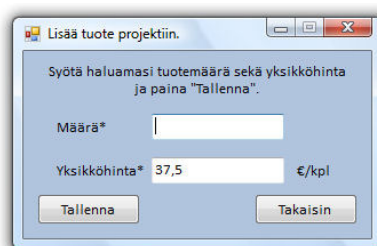
”Lisää tuotteita projektiin” -lomakkeella projekteihin lisätään tuotteita. Jos projekti on esimerkiksi kengänpohjan pesulaite, tällä lomakkeella projektiin lisätään kaikki ne tuotteet, jotka sen valmistamiseen tarvitaan. Kaikkien muiden tietojen, kuten projektin ja siihen liittyvien tuotteiden tietojen, pitää olla syötettyinä tietokantaan ennen, kuin tuotteet voidaan lisätä projektin tietoihin.

Lomakkeen yläreunassa on kaksi vetovalikkoa, joista valitaan, mihin projektiin tuotteita lisätään. Projektin voi valita joko projektinumeron tai projektin nimen mukaan. Kun projektinnumero-vetovalikosta valitaan jokin projektinnumero, vaihtuu projektin nimi-vetovalikkoon automaattisesti vastaavan projektin nimi.

Lomakkeella näkyy sille tultaessa kaikkien tietokantaan syötettyjen tuotteiden tiedot. Valintaruudusta ”Suodata tuotteet toimittajan mukaan” napsautettaessa toimittaja-vetovalikko aktivoituu, jolloin toimittajaa vaihtamalla näytetään ainoastaan kyseisen toimittajan tuotteet. Kun valintaruutua napsautetaan uudelleen, suodatus poistuu ja DataGridView-objektissa näytetään taas kaikkien tuotteiden tiedot.

DataGridView-objektin ominaisuuksia on muutettu siten, että yksittäisten tietueitten sijasta valinta kohdistuu aina koko riviin. Tässä tapauksessa ei ole mitään syytä, miksi tietueet pitäisi päästä valitsemaan erikseen, sillä projektiin lisätään aina tuotteen kaikki tiedot.

Tuotteiden lisääminen projektiin aloitetaan valitsemalla haluttu tuoterivi DataGridView-objektista. Tämän jälkeen napsautetaan Tallenna-painiketta, joka aukaisee ”Lisää tuote projektiin” -lomakkeen ”Lisää tuotteita projektiin” -lomakkeen päälle. Näin taustalla olevasta lomakkeesta on yhä mahdollista tarkistaa, mitä tuotetta oltiin lisäämässä projektiin, sillä valittu rivi pysyy edelleen aktiivisena.



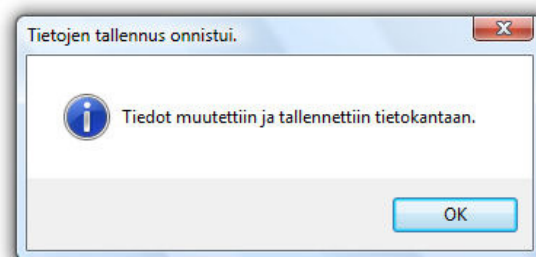
Kuva 31. Tuotteen lisääminen projektiin Kuva 32. Tuotteen lisäys projektiin onnistui

”Lisää tuote projektiin” -lomakkeella syötetään määrä, kuinka paljon valittuja tuotteita halutaan tallentaa projektin tietoihin. Jos yksikkö on esimerkiksi €/kpl ja Määräkenttään syötetään luku 12, projektiin tallentuu 12 kappaletta kyseistä tuotetta. Jos yksikkö on €/m ja määräksi syötetään 12, projektiin tallentuu 12 metriä tuotetta.

Määrän lisäksi pakollinen tieto on yksikköhinta, jonka voi vielä vaihtaa halutessaan. Yksikköhinnan vaihtaminen mahdollistettiin siksi, että joissain projekteissa hinta voi vaihdella esimerkiksi alennuksen takia. Jos yksikköhinnan vaihtaa ”Lisää tuote projektiin” -lomakkeella, uusi yksikköhinta tulee vain kyseiseen projektiin ja vanha yksikköhinta säilyy tietokannassa ja muissa projekteissa. Alkuperäisen yksikköhinnan voi vaihtaa ”Tarkastele ja muokkaa tuotteiden tietoja” -lomakkeella, joka esitellään osiossa 6.4.3. Tallenna-painiketta napsautettaessa ”Tuotteen lisäys onnistui” -lomake ilmoittaa, että tuote lisättiin projektiin ja ohjelma palaa ”Lisää tuotteita projektiin” -lomakkeelle.

6.4 Tietojen tarkasteleminen ja muokkaaminen

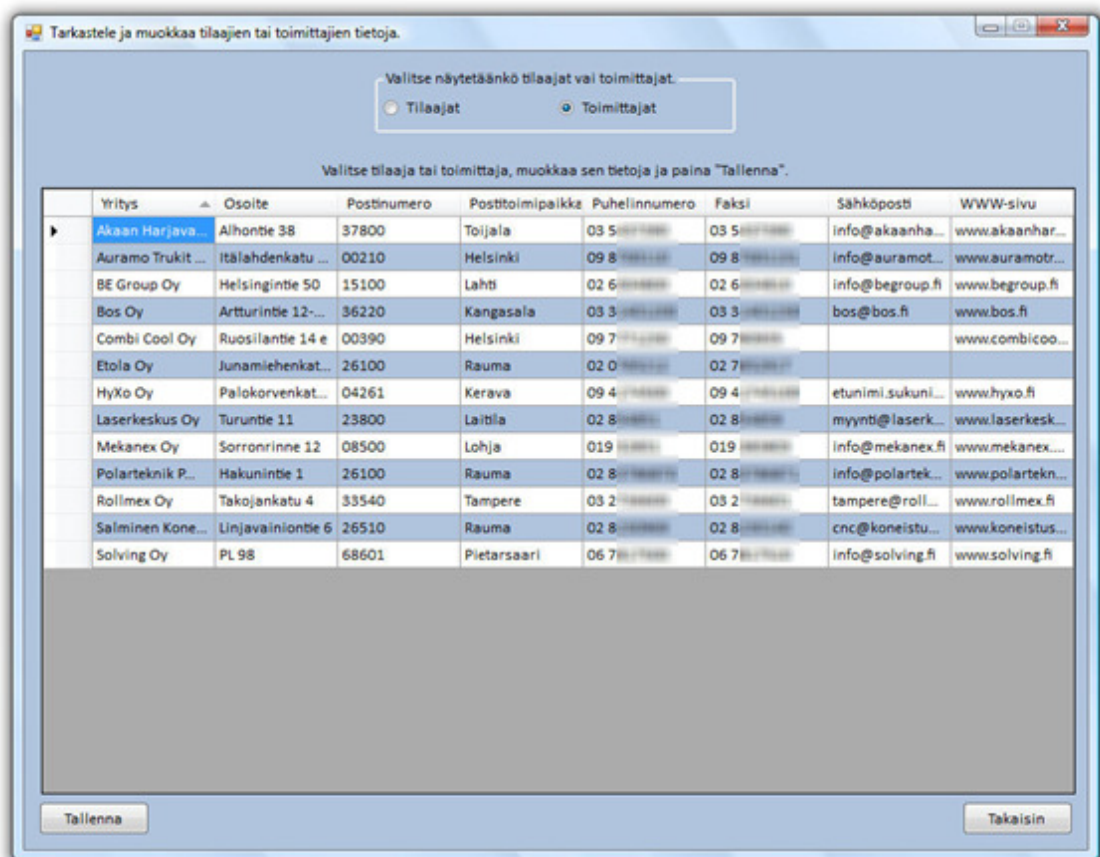
Ohjelmassa on viisi eri lomaketta tietojen tarkastelemista ja muokkaamista varten. Jokaisella lomakkeella on DataGridView-objekti, jonka tietueiden tietoja voidaan muokata. Jokainen tietue voidaan valita erikseen sitä napsauttamalla. Vanhoja tietoja voidaan muokata, mutta myös tyhjiin tietueisiin voidaan lisätä puuttuva tieto. Kun halutut muokkaukset on tehty, tiedot tallentuvat tietokantaan Tallenna-painiketta napsauttamalla. Jos lomakkeelta poistutaan ilman Tallenna-painikkeen napsautusta, tiedot eivät tallennu tietokantaan. Tallenna-painiketta napsautettaessa ruutuun ilmestyy jokaisella lomakkeella samanlainen ilmoitus, joka ilmoittaa, että tiedot muutettiin ja tallennettiin tietokantaan onnistuneesti. Ilmoitus kuitataan painamalla OK-painiketta.



Kuva 33. Tietojen tallennus onnistui

Osiossa 6.4 esiteltiin ”tietojen tarkasteleminen ja muokkaaminen” -lomakkeiden yleiset ominaisuudet eli kaikille näille lomakkeille yhteiset piirteet. Osioissa 6.4.1-6.4.5 esitellään jokaisen lomakkeen ominaispiirteet eikä tässä osiossa esiteltyjä kaikille lomakkeille yhteisiä ominaisuuksia enää mainita.

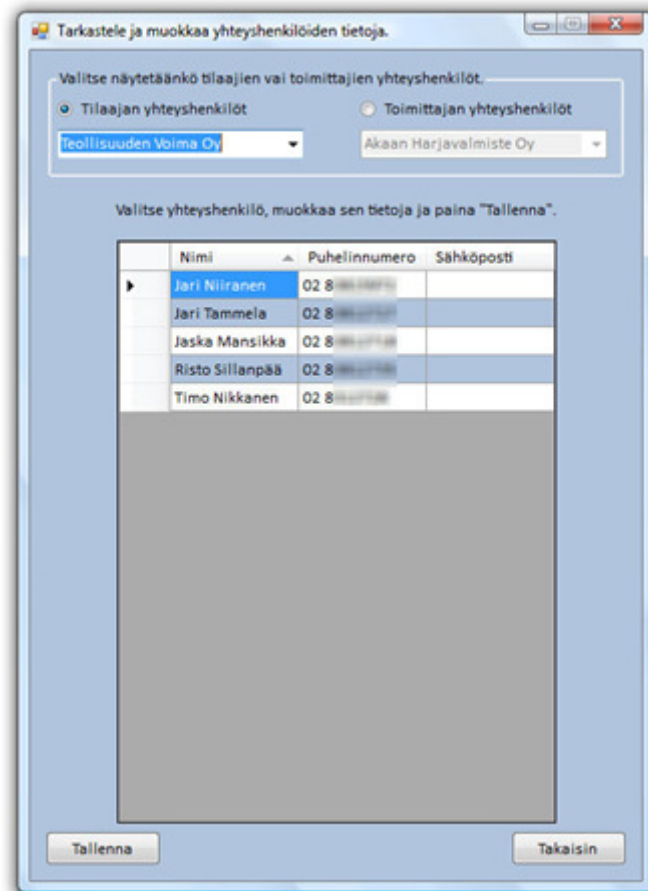
6.4.1 Tilaajien ja toimittajien tarkasteleminen ja muokkaaminen



Kuva 34. Tilaajien ja toimittajien tietojen tarkasteleminen ja muokkaaminen

”Tarkastele ja muokkaa tilaajien ja toimittajien tietoja” -lomakkeen yläreunassa on kaksi valintanappia, joiden valintaa vaihtamalla voi valita, näytetäänkö tilaajien vai toimittajien tiedot. Valinnan vaihtaminen onnistuu valintanappia napsauttamalla.

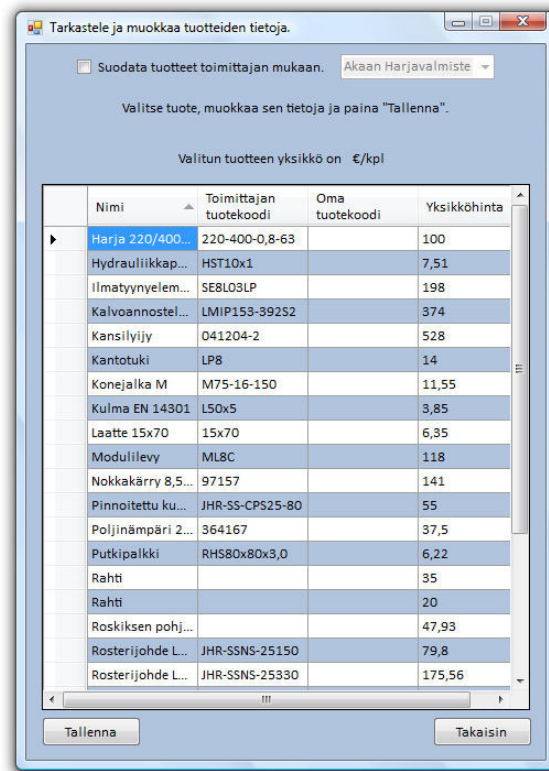
6.4.2 Yhteysenkilöiden tarkasteleminen ja muokkaaminen



Kuva 35. Yhteysenkilöiden tietojen tarkasteleminen ja muokkaaminen

”Tarkastele ja muokkaa yhteysenkilöiden tietoja” -lomakkeella voidaan katsella ja muokata sekä tilaajien että toimittajien yhteysenkilöiden tietoja. Valintanapeilla valitaan, näytetäänkö tilaajien vai toimittajien yhteysenkilöiden tiedot. Kun jompikumpi valintanappi on valittuna, sitä vastaava vetovalikko valintanapin alapuolella aktivoituu. Vetovalikosta valitaan minkä yrityksen yhteysenkilöt näytetään DataGridView-objektissa. Kuvassa 35 valintanappi ”Tilaaajan yhteysenkilöt” on valittuna, jolloin sen alla oleva tilaaja-vetovalikko on aktiivinen. Tilaaajista on valittu Teollisuuden voima Oy, jolloin DataGridView-objektissa näytetään kaikki kyseisen yrityksen yhteysenkilöt. Kenelläkään yrityksen yhteysenkilöistä ei ole sähköpostiosoitetta syötettynä tietokantaan, joten esimerkiksi sen tiedon lisääminen onnistuu tältä lomakkeelta.

6.4.3 Tuotteiden tarkasteleminen ja muokkaaminen



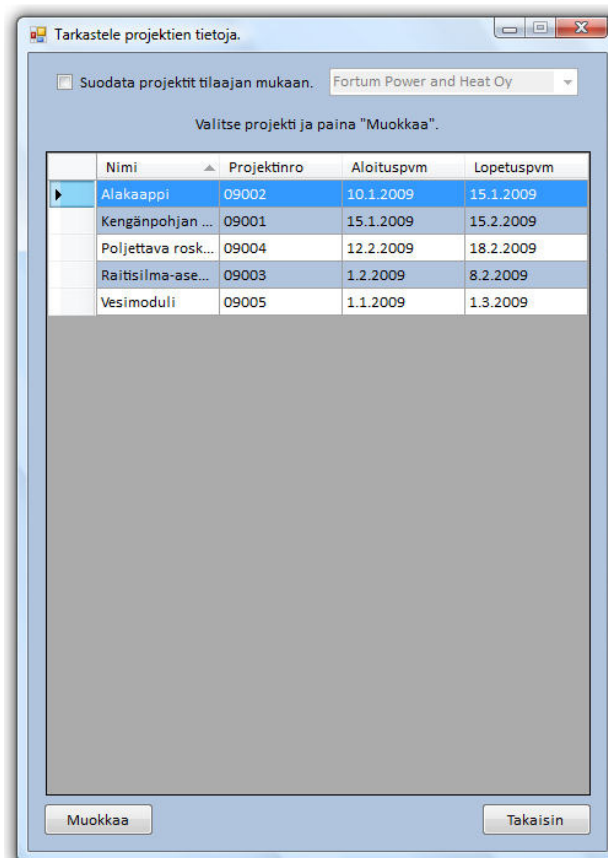
Kuva 36. Tuotteiden tietojen tarkasteleminen ja muokkaaminen

”Tarkastele ja muokkaa tuotteiden tietoja” -lomakkeella esitellään kaikkien tietokannasta löytyvien tuotteiden tiedot. Lomakkeelle tullessa DataGridView-objektissa näytetään kaikkien tuotteiden tiedot. Lomakkeen yläreunassa on ”Suodata tuotteet toimittajan mukaan” -valintaruutu, jota napsautettaessa toimittaja-vetovalikko aktivoituu. Kun vetovalikosta valitaan jokin toimittaja, ainoastaan kyseisen toimittajan tuotteet näytetään DataGridView-objektissa. Tämä toiminto helpottaa tuotteiden selausta, kun halutaan tarkastella vain tietyn toimittajan tietoja. Kun valintaruutua napsautetaan uudelleen, suodatus poistuu ja kaikkien toimittajien kaikki tuotteet näytetään taas DataGridView-objektissa.

Tuotteen yksikkö ilmoitetaan DataGridView-objektin yläpuolella. Yksikkö vaihtuu aina sen mukaan, mikä tuoterivi on valittuna. Yksikköä ei voida esittää samassa DataGridView-objektissa muiden tietojen rinnalla, sillä yksikkö on tallennettu Tuote-tauluun pelkkänä ID:nä, joka on siis luku. Jos yksikkö olisi haluttu esittää DataGridView-objektissa, olisi se onnistunut vain tekemällä halutuista tiedoista kysely, eikä kyselyn

tietoja voidaan muuttaa DataGridView-objektissa. Siksi päädyttiin ratkaisuun, jossa yksikkö esitetään erikseen DataGridView-objektin ulkopuolella eikä sitä näin ollen voi muuttaa. Jos yksikkö halutaan vaihtaa, on koko tuote poistettava ja lisättävä tietokantaan uudelleen. Viishanke Oy:n kanssa tultiin kuitenkin siihen tulokseen, että tuotteen yksikköä ei tarvitse missään tapauksessa vaihtaa, joten tämä toiminto nähtiin tarpeettomaksi.

6.4.4 Projektien tarkasteleminen ja muokkaaminen



Kuva 37. Projektien tietojen tarkasteleminen

”Tarkastele projektien tietoja” -lomakkeella näytetään projekti ja sen tiedot. DataGridView-objektissa näytetään projektin nimi, projektinnumero sekä projektin aloitus- ja lopetuspäivämäärä. Lomakkeelle tultaessa DataGridView-objektissa näytetään kaikkien projektien tiedot. Lomakkeen yläreunassa olevan ”Suodata projektit tilaajan mukaan” -valintaruudun avulla aktivoidaan tilaaja-vetovalikko, josta voidaan valita haluttu tilaaja.

Näin projektit suodatetaan tilaajan mukaan ja vain yhden tilaajan projektit näytetään DataGridView-objektissa kerrallaan.

Projektin tietoja ei voida muokata suoraan DataGridView-objektissa samanlaisesti kuin muilla lomakkeilla. Kun halutaan esittää projektin kaikki tiedot, joudutaan tietoja ottamaan monesta taulusta eli tekemään kysely. Samanlainen tilanne on ”Tarkastele ja muokkaa tuotteiden tietoja” -lomakkeessa, joka esitellään osiossa 6.4.3. Kyselyn tietoja ei kuitenkaan voida muuttaa DataGridView-objektissa. Tämän vuoksi projektin tietojen muuttamiseen on luotu oma lomakkeensa, jossa haluttuja tietoja muutetaan tekstikentissä. Projektin tietoja pääsee muokkaamaan valitsemalla haluttu projektirivi ja painamalla Muokkaa-painiketta. Tällöin aukeaa ”Muokkaa projektin tietoja” -lomake.

Muokkaa projektin tietoja.

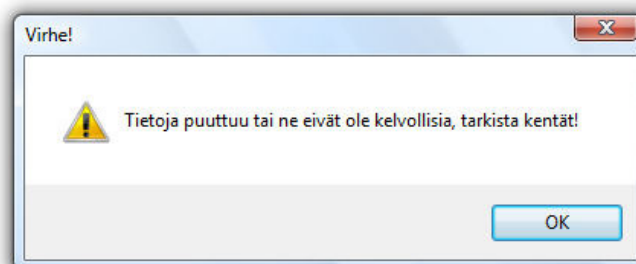
Muokkaa projektin tietoja ja paina "Tallenna".

Projektin nimi	Kengänpohjan pesulaite
Projektinumero	09001
Aloituspäivämäärä	15.1.2009
Lopetuspäivämäärä	15.2.2009
Tilaaaja	Teollisuuden Voima C
Yhteyshenkilö	Timo Nikkanen

Tallenna Takaisin

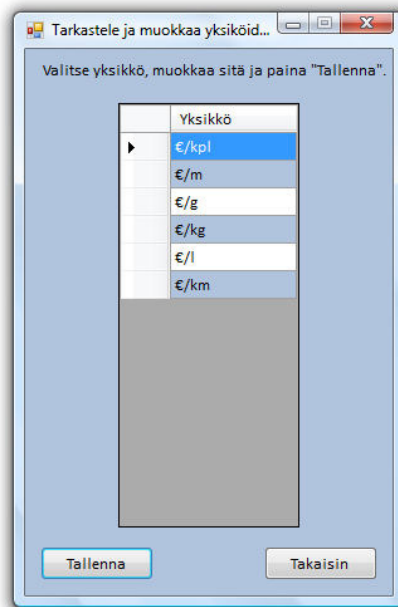
Kuva 38. Projektin tietojen muokkaaminen.

”Muokkaa projektin tietoja” -lomakkeella projektin tiedot esitetään tekstikentissä. Tekstikenttien sisältöä pääsee muokkaamaan kenttää napsauttamalla. Jos Tallenna-painiketta napsautettaessa kaikkiin kenttiin ei ole syötetty tietoja tai jos projektinumeroksi on syötetty jo käytössä oleva projektinumero, näyttöön ilmestyy virheilmoitus. Ilmoitus kuitataan OK-painiketta napsauttamalla, jonka jälkeen palataan ”Muokkaa projektin tietoja” -lomakkeelle.



Kuva 39. Virhe projektin tietojen muokkaamisessa

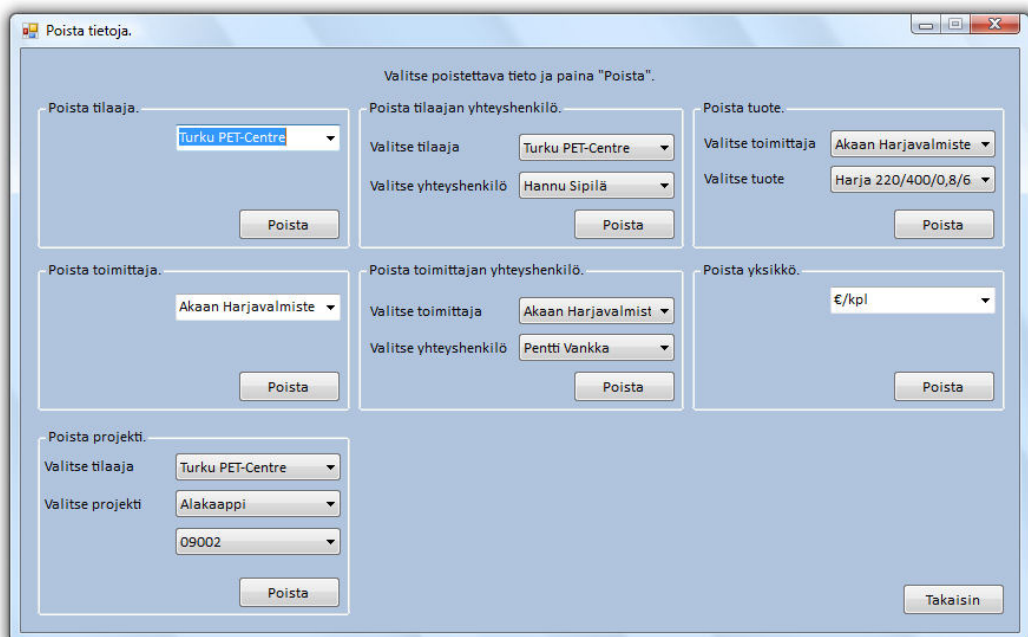
6.4.5 Yksiköiden tarkasteleminen ja muokkaaminen



Kuva 40. Yksiköiden tietojen tarkasteleminen ja muokkaaminen

”Tarkastele ja muokkaa yksiköiden tietoja” -lomakkeella on DataGridView-objekti, jossa näytetään kaikki tietokantaan tallennetut yksiköt. Yksiköitä voidaan muokata napsauttamalla haluttua tietuetta ja kirjoittamalla siihen halutut korjaukset.

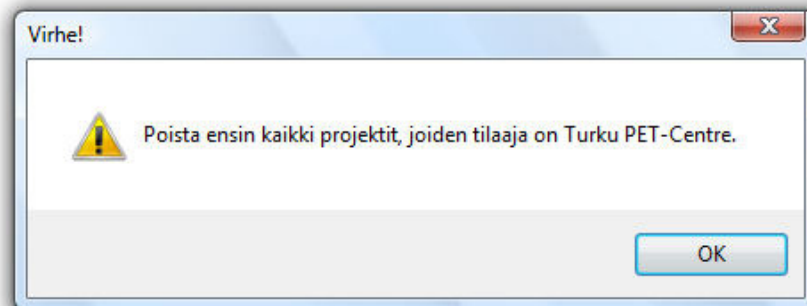
6.5 Tietojen poistaminen



Kuva 41. Tietojen poistaminen

”Poista tietoja” -lomakkeella voidaan poistaa tietoja tietokannasta. Poistettavia tietoja ovat tilaajat, toimittajat, projektit, tilaajan ja toimittajan yhteyshenkilöt, tuotteet ja yksiköt. Vetovalikosta valitaan ensin haluttu poistettava tieto, jonka jälkeen painetaan samassa ryhmässä olevaa Poista-painiketta. Projektien poistossa valitaan ensin tilaaja, jonka projekti halutaan poistaa ja sen jälkeen poistettava projekti joko nimen tai projektinumeron perusteella. Yhteyshenkilöiden poistossa valitaan ensin kenen tilaajan tai toimittajan yhteyshenkilö poistetaan ja sitten poistettava yhteyshenkilö. Tuotteen poistossa valitaan ensin tuotteen toimittaja ja sitten itse poistettava tuote.

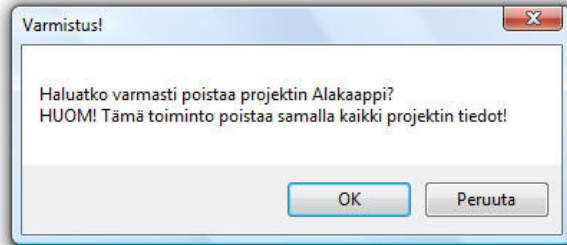
Jos halutaan poistaa tilaaja, on ensin poistettava kaikki kyseisen tilaajan tilaamat projektit. Tämä siksi, että jokaisella projektilla on oltava jokin tilaaja ja jos tilaaja poistetaan, puuttuu projektilta yksi pakollinen tieto. Sama tilanne on toimittajien poiston kohdalla. Kaikki kyseisen toimittajan tiedot on poistettava ennen kuin itse toimittaja voidaan poistaa. Myös tilaajan yhteyshenkilöt voidaan poistaa vasta sen jälkeen, kun kaikki projektit, joissa kyseinen henkilö on yhteyshenkilönä, on poistettu. Jos tieto yritetään poistaa ennen kuin se on mahdollista, ilmestyy näytölle virheilmoitus. Eli esimerkiksi jos yritetään poistaa tilaaja, jolla on tilattuja projekteja, ilmestyy ruutuun virheilmoitus, jossa kehoitetaan käyttäjää poistamaan ensin kaikki projektit, jonka tilaaja on kyseinen yritys.



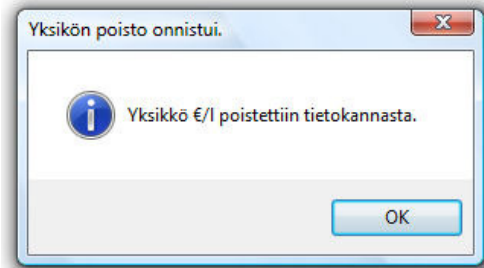
Kuva 42. Virhe tilaajan poistossa

Muiden tietojen eli projektien, toimittajien yhteyshenkilöiden, tuotteiden ja yksiköiden poistossa Poista-painiketta napsautettaessa näyttöön ilmestyy Varmistus-ikkuna, jossa käyttäjältä kysytään, halutaanko tieto varmasti poistaa. Esimerkiksi projektia poistaessa varmistetaan, että käyttäjä on ymmärtänyt, että kaikki projektin tiedot katoavat, kun se poistetaan. OK-painiketta napsautettaessa tieto poistuu ja näytölle ilmestyy ilmoitus,

jossa kerrotaan, mikä tieto poistettiin tietokannasta. Peruuta-painiketta napsautettaessa tietojen poisto peruutetaan ja palataan takaisin ”Poista tietoja” -lomakkeelle.



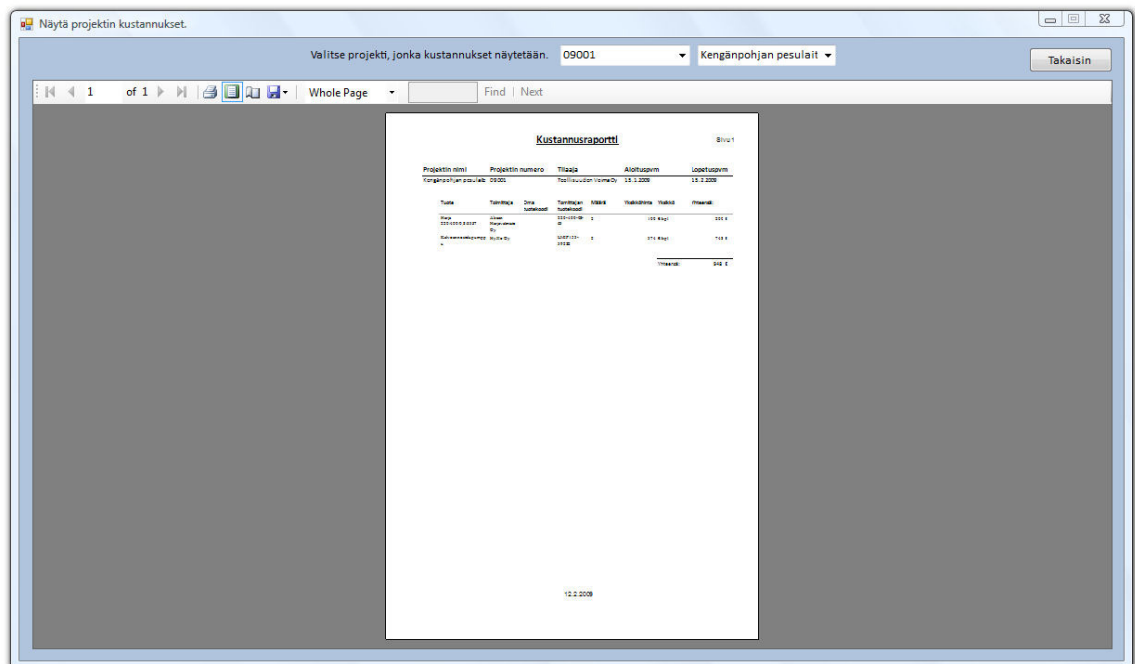
Kuva 43. Projektin poiston varmistus



Kuva 44. Yksikön poisto onnistui

6.6 Kustannusraportti

Kustannusraportti on koko ohjelman olennaisin toiminto ja perusidea. Raportin avulla Viishanke Oy pystyy dokumentoimaan yritykseen tulleet tilaukset ja niiden kustannukset sekä tarkastelemaan niitä myöhemmin. Kustannusraportista voidaan tulostaa haluttaessa myös paperiversio. Kustannusraportti esitetään liitteessä 1.



Kuva 45. Kustannusraportti lomakkeelle tultaessa

”Näytä projektin kustannukset” -lomakkeella päästään tarkastelemaan projektien tuotteita ja niiden kustannuksia. Lomakkeelle tullessa raportti näytetään niin, että koko sivu näkyy. Lomakkeen yläreunasta valitaan minkä projektin tiedot halutaan näyttää. Projekti voidaan valita joko projektinumeron tai projektin nimen perusteella. Työkalurivillä on erilaisia työkaluja raportin katselua varten.



Kuva 46. Työkalurivi

Työkalurivin eri toiminnot on eriteltyä alla. Find-Next -toiminto ei ole käytössä tällä lomakkeella.



Vaihda sivua, jos raportti on pidempi kuin yksi sivu.



Tulosta raportti.



Vaihda näkymää lomakkeella.

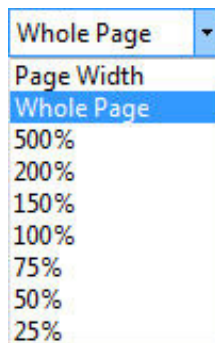


Vaihda sivun asetuksia.



Excel
Acrobat (PDF) file

Tallenna raportti joko Excel- tai PDF-tiedostona.



Määrittele, kuinka suurena raportti näytetään lomakkeella.

Näytä projektin kustannukset.

Valitse projekti, jonka kustannukset näytetään. 09001 Kengänpohjan pesulaite Takaisin

1 of 1 Page Width Find | Next

Kustannusraportti

Sivu 1

Projektin nimi	Projektin numero	Tilaaaja	Aloituspvm	Lopetuspvm
Kengänpohjan pesulaite	09001	Teollisuuden Voima Oy	15.1.2009	15.2.2009

Tuote	Toimittaja	Oma tuotekoodi	Tomittajan tuotekoodi	Määrä	Yksikköhinta	Yksikkö	Yhteensä:
Hajja 220-400/0,8/63/37	Akaan Harjavalmist Oy		220-400-0,8- 63	2	100 €/kpl		200 €
Kalvoannostelupumppu	HyXo Oy		LMIP153- 392S2	2	374 €/kpl		748 €
Yhteensä:							948 €

Kuva 47. Kustannusraportti, kun raportin kooksi on määritelty sivun leveys.

Kustannusraportin yläreunassa on otsikko ”Kustannusraportti” ja sivunumero. Projektin nimi, numero, tilaaja sekä aloitus- ja lopetuspäivämäärä on eritelty raportin alussa. Projektin tuotteet on listattu Tuote-otsikon alle. Projektissa Kengänpohjan pesulaite on kaksi eri tuotetta ja molempia kaksi kappaletta. Raportissa näytetään yksikköhinnat sekä tuotteen määrästä riippuva kokonaishinta. Myös koko projektin kokonaishinta on määritelty raportin lopussa. Kulloinenkin päivämäärä tulostuu aivan raportin alareunaan.

7 YHTEENVETO

7.1 Yleistä

Kaikki ominaisuudet, joista sovittiin aloituspalaverissa, saatiin toteutetuksi ohjelmassa. Kustannusseurantaohjelman molemmat osiot, suunnittelu ja toteutus, onnistuivat. Ohjelman päätarkoitus, kustannusraportti, voidaan luoda helposti ja nopeasti. Kustannusraportista selviävät kaikki halutut tiedot yksinkertaisesti ja selkeästi. Myös myöhemmin toivotut ja tarpeellisiksi huomautetut lisäominaisuudet onnistuttiin toteuttamaan.

Sain itse suunnitella ohjelman toiminnallisuuden ja ulkoasun. Onnistuin tässä mielestäni hyvin, sillä käyttöliittymä on selkeä ja johdonmukainen. Kaikki toiminnot löytyvät helposti, sillä vaihtoehdot ovat aina esillä. Toiminnot ovat yhteneväiset joka lomakkeella, jolloin ohjelman käyttäminen on mielekästä. Tietojen syötöstä on tehty mahdollisimman yksinkertaista ja käyttäjäystävällistä.

Haastavinta opinnäytetyön teossa oli ohjelmakoodin toteutus. Tiesin, mitä toimintoja halutaan, mutta niiden toteutus ei aina ollut yksinkertaista. Monet toiminnot toteutettiin kokeilemalla ja soveltamalla. Yksinkertaistenkin toimintojen toteutus tuntui välillä haastavalta.

Tämän opinnäytetyön tekeminen oli mielekkäällä tavalla haastavaa ja vaativaa. Opin myös sen, että säännöllinen yhteydenpito työn tilaajan kanssa on välttämätöntä, jotta ohjelmasta tulee täysin toiveita vastaava. Työn onnistuminen vaati paljon itseopiskelua ja asioiden selvittelyä, joten pääsin tutustumaan ohjelmointiin aivan uudella tavalla.

7.2 Kehitysehdotukset

Projektien kustannusseurantaohjelma tehtiin Viishanke Oy:n tarpeita vastaavaksi. Ohjelmassa on kuitenkin kehitysmahdollisuuksia, jotka on syytä ottaa huomioon varsinkin silloin, jos ohjelma päättyy jonkin suuremman yrityksen käyttöön. Niitä ei kuitenkaan nähty tarpeellisiksi Viishanke Oy:ssä tällä hetkellä eikä myöskään työn laajuutta rajatessa.

Nyt ohjelmassa ei ole virheentarkistusta tietojen tallennusprosessissa. Ainoastaan käyttäjän syötteen virheettömyys tarkistetaan, mutta itse tietojen tallennuksessa tietokantaan ei ole varmistusta, jos tietojen tallennus epäonnistuu. Virheentarkistus tallennusprosessissa on lähes välttämätön toiminto suuremmissa yrityksissä ja kun ohjelmalla on monta käyttäjää.

Ohjelmassa ei ole toimintoa, jonka avulla käyttöliittymän ulkoasua, kuten värejä ja fontteja, voitaisiin muuttaa. Toimintoa ei tarvittu nyt, kun ohjelmalla on vain yksi käyttäjä, joka hyväksyi ulkoasun. Jos ohjelmalla on tulevaisuudessa monia käyttäjiä, ulkoasun muutosmahdollisuus lisäisi käyttäjäystävällisyyttä.

Yksikkö-taulukon toteutus olisi voitu tehdä niin, että yksikköä ei olisi tarvinnut näyttää erillisenä tietona DataGridView-objektin yläpuolella, kuten ”Tarkastele ja muokkaa tuotteiden tietoja” -lomakkeella kuvassa 36. Päätin kuitenkin toteuttaa yksikön esittämisen näin, sillä se oli yksinkertaisin ja työn tilaajan hyväksymä tapa. Ohjelman jatkokehityksessä yksikkö-taulukon toteutus voidaan muuttaa, jos niin halutaan.

LÄHTEET

1. Lambert, S., Dow Lambert, M. & Preppernau, J. Microsoft Office Access 2007: Tehokas hallinta. 1. painos. Jyväskylä: Readme.fi Oy, 2008. 308 s.
2. Wikipedia, suomenkielinen, hakusana: Access [verkkodokumentti].[Viitattu 16.2.2009] Saatavissa: <http://fi.wikipedia.org/wiki/Access>
3. Verkkopedagogi [verkkodokumentti].[Viitattu 17.2.2009] Saatavissa: <http://verkkopedagogi.net/vanhat/fi/sisalto/materiaalit/access2003/luku0375c6.html?C:D=419702&selres=419702>
4. Polvinen, T. Tietokannat käytännön työssä. 1. painos. Porvoo: Teknolit, 1999. 286 s.
5. Järvinen, J. Visual Studio 2008 -käsikirja. Porvoo: WSOY, 2008. 425 s.
6. Wikipedia, suomenkielinen, hakusana: Visual Basic [verkkodokumentti].[Viitattu 16.2.2009] Saatavissa: http://fi.wikipedia.org/wiki/Visual_Basic
7. Wikipedia, suomenkielinen, hakusana: Visual Studio [verkkodokumentti].[Viitattu 16.2.2009] Saatavissa: http://fi.wikipedia.org/wiki/Visual_Studio
8. Franklin, K. VB.NET. 1. painos. Jyväskylä: WSOY, 2002. 342 s.
9. Halvorson, M. Microsoft Visual Basic 2008: Tehokas hallinta. 1. painos. Jyväskylä: Readme.fi Oy, 2008. 550 s.
10. Kuivalahti, M. & Peltoniemi, J. Tietojärjestelmän sisällön suunnittelu. 2. painos. Helsinki: VAPK-kustannus, 1992. 107 s.
11. Microsoft Office Online [verkkodokumentti].[Viitattu 18.2.2009] Saatavissa: <http://office.microsoft.com/fi-fi/access/HA012242471035.aspx>
12. Hovi, A., Ylinen, J. & Koistinen, H. Tietovarastot liiketoiminnan tukena. Jyväskylä: Talentum Media Oy, 2001. 276 s.
13. Jyväskylän yliopisto [verkkodokumentti].[Viitattu 18.2.2009] Saatavissa: <http://www.mit.jyu.fi/opiskelu/seminarit/ohjelmistotekniikka/kayttoliittyma/index.html#luku3>
14. Kalimo, A., Pettersson, M., Sinkkonen, I. & Suikola, E. Graafisen käyttöliittymän suunnittelu: Opas ohjelmistojen käytettävyyteen. Jyväskylä: Suomen ATK-kustannus Oy, 1996. 225 s.

Kustannusraportti

Sivu 1

Projektin nimi	Projektin numero	Tilaaaja	Aloituspvm	Lopetuspvm			
Vesimoduli	09005	Hidex Oy	1.1.2009	1.3.2009			
Tuote	Toimittaja	Oma tuotekoodi	Tomittajan tuotekoodi	Määrä	Yksikköhinta	Yksikkö	Yhteensä:
Hydrauliikkaputki	Polarteknik PMC Oy		HST10x1	12	7.51 €/m		90.12 €
Ilmatyynyelementti	Solving Oy		SESL03LP	4	198 €/kpl		792 €
Kansilyijy	Salminen Koneistus Oy		041204-2	1	528 €/kpl		528 €
Kantotuki	Solving Oy		LP8	4	14 €/kpl		56 €
Laatta 15x70	BE Group Oy		15x70	14.5	6.35 €/kg		92.075 €
Modulilevy	Solving Oy		ML8C	4	118 €/kpl		472 €
Pinnoitettu kuormalevy	Mekanex Oy		JHR-SS- CPS25-80	4	55 €/kpl		220 €
Putkipalkki	BE Group Oy		RHS80x80x3,0	12	6.22 €/m		74.64 €
Rahti	BE Group Oy			1	35 €/kpl		35 €
Rahti	Solving Oy			1	20 €/kpl		20 €
Rosterijohde L=150	Mekanex Oy		JHR-SSNS- 25150	2	79.8 €/kpl		159.6 €
Rosterijohde L=330	Mekanex Oy		JHR-SSNS- 25330	1	175.56 €/kpl		175.56 €
Suojaputki	Salminen Koneistus Oy		041205-2	1	77 €/kpl		77 €
Venttiili	Solving Oy		EAR4000F04	1	24.16 €/kpl		24.16 €
Vesimodulin kansi	Salminen Koneistus Oy		041201-2	1	330 €/kpl		330 €
V-johdepyörä 25 kiinteä	Mekanex Oy		JHR-SS- SJ25CNS	4	43 €/kpl		172 €
V-johdepyörä 25 säädettävä	Mekanex Oy		JHR-SS- SJ25ENS	12	44 €/kpl		528 €
						Yhteensä:	3846.155 €

1.4.2009