

Antti Uusi-Seppä

TYÖKALU KIRPPUTORIN PÄIVITTÄISTEN ASIOIDEN
HALLINTAAN

Tietojenkäsittelyn koulutusohjelma

2012



TYÖKALU KIRPPUTORIN PÄIVITTÄISTEN ASIOIDEN HALLINTAAN

Uusi-Seppä, Antti

Satakunnan Ammattikorkeakoulu

Tietojenkäsittelyn koulutusohjelma

Helmikuu 2012

Ohjaaja: Grönholm, Jukka

Sivumäärä: 34

Asiasanat: C#, Microsoft Sharepoint 2010, Microsoft SQL Server 2008

Opinnäytetyön tarkoituksena oli kehittää sovellus, joka tulee kirpputorien käyttöön helpottamaan niiden päivittäistä toimintaa huomattavasti. Sovelluksen avulla vältetään jokaisen tuotteen tietojen kirjoittamiselta käsin sekä siltä, että tuotteen ostaja vaihtaa tuotteen hinnan jostain halvemmasta tuotteesta. Sovelluksen avulla asiakas voi varata looseja sekä muuttaa tuotteen tietoja kotoa käsin.

Sovellus tulee testauskäyttöön Ellinoora-kirpputorille. Heidän palautteensa avulla kehitetään sovellusta jolloin se voidaan ottaa käyttöön myös muilla kirpputoreilla.

Opinnäytetyön toteutuksessa käytettiin Sharepointtia, Visual Studiota sekä Microsoft SQL Serveriä. Ohjelmointikielenä käytettiin C#-kieltä.

A TOOL FOR HANDLING DAILY BUSINESS AT FLEA MARKET

Uusi-Seppä Antti

Satakunta University of Applied Sciences

Degree Programme in Business Information Systems

February 2012

Supervisor: Grönholm, Jukka

Number of pages: 34

Keywords: C#, Microsoft Sharepoint 2010, Microsoft SQL Server 2008

The purpose of this thesis was to develop software that helps daily business at flea markets. With this software flea markets don't have to handwrite information to each product. Customers can't change product information to cheaper price because every product has unique id-number. With this software you can reserve lodge and change your product information at home.

This software will come in use of flea market called Ellinoora. With their feedback we can improve software and maybe someday this software is in use of other flea markets.

The technique used in this thesis was Sharepoint, Visual Studio and SQL Server. Programming language was C#.

Sisällysluettelo

1 JOHDANTO	5
2 KÄYTETYT TYÖKALUT	6
2.1 Microsoft Visual Studio	6
2.2 Microsoft Sharepoint 2010	7
2.2.1 Uusi Sharepoint-sivu.....	8
2.2.2 Sharepoint webpartin lisääminen sivulle	10
2.3 Microsoft SQL Server.....	11
2.4 Relaatiotietokannat.....	11
3 SUUNNITTELU	14
3.1 Käyttöliittymän suunnittelu	14
3.2 Tietokannan suunnittelun merkitys	15
3.3 Tietokannan suunnittelu	16
4 SOVELLUS PROJEKTINA	18
5 TOTEUTUS.....	20
5.1 Tietokannan taulut.....	20
5.2 Sovelluksen toteutus	21
5.2.1 Uuden käyttäjän lisäys.....	21
5.2.2 Sovellukseen kirjautuminen.....	22
5.2.3 Uuden sopimuksen luonti.....	23
5.2.4 Käyttäjän sopimukset.....	24
5.2.5 Uuden tuotteen lisäys	25
5.2.6 Lista tuotteista ja niiden muokkaus ja poisto.....	26
5.2.7 Tuotteen myyntitapahtuma	27
5.2.8 Viestin lähettäminen.....	29
5.2.9 Ilmoitus uusista viesteistä	30
5.2.10 Lista käyttäjän viesteistä	30
5.2.11 Käyttäjän myydyt tuotteet.....	32
6 YHTEENVETO	33
LÄHTEET	34

1 JOHDANTO

Opinnäytetyöni aiheena oli kehittää sovellus helpottamaan kirpputorin päivittäistä toimintaa. Sovellus jakaantuu pääkäyttäjiin eli kirpputorin työntekijöihin sekä käyttäjiin eli kirpputorin asiakkaisiin. Pääkäyttäjät pääsevät ylläpitosivulle jossa lisätään uudet asiakkaat sekä tehdään sopimukset. Ylläpitosivulla myös aloitetaan uusi myyntitapahtuma. Asiakkaat voivat tehdä loosivaraustoiveita sekä lisätä tuotteita pääkäyttäjän tekemään sopimukseen sekä muokata niitä kotoa käsin. Sovelluksessa on myös viestintäjärjestelmä, jonka avulla asiakas voi tehdä varaustoiveita tai vaikkapa kysyä liikkeen aukioloaikoja.

Sain aikaa sovelluksen tekoon noin kolme kuukautta. Aluksi meni hieman aikaa uusien työkalujen opettelemiseen. Kun työkalut olivat hieman paremmin hallinnassa, aloin suunnittelemaan sovelluksen tietokantaa sekä käyttöliittymää. Käsittelen tietokannan sekä käyttöliittymän suunnittelua luvussa 3. Tämän jälkeen aloin toteuttamaan itse sovellusta, jota käsittelen luvussa 5.

2 KÄYTETYT TYÖKALUT

2.1 Microsoft Visual Studio

Visual Studio on Microsoftin tärkein Windows-kehitin. Sillä kehitetään pääasiassa .NET-pohjaisia sovelluksia valmistajan omiin käyttöjärjestelmäympäristöihin, eli työasemiin, palvelimiin sekä nykyisin myös viihde- ja mobiililaitteisiin. Visual Studiosta on tarjolla useita erilaisia kokonaisuuksia alkaen opiskelijoille ja harrastajille tarkoitetuista ilmaisista versioista aina laajoihin ammattilaisille tarkoitettuihin versioihin.

Visual Studio tukee erilaisia sovellustyyppejä, joista tärkeimpiä ovat:

- Perinteiset, graafiset Windows-sovellukset (Windows Forms)
- Uudentyyppiset, graafiset sovellukset WPF-tekniikkaa käyttäen
- Konsolisovellukset
- Windowsin palvelut (service application)
- Web-sovellukset
- Web Services-sovellukset eli verkkopalvelut
- Microsoft Office-tuotteisiin integroidut sovellukset
- Mobiilisovellukset
- Asennusohjelmat
- Visual Studion kehitysympäristön laajennukset ja makrot

(Järvinen 2008, 2-5.)

2.2 Microsoft Sharepoint 2010

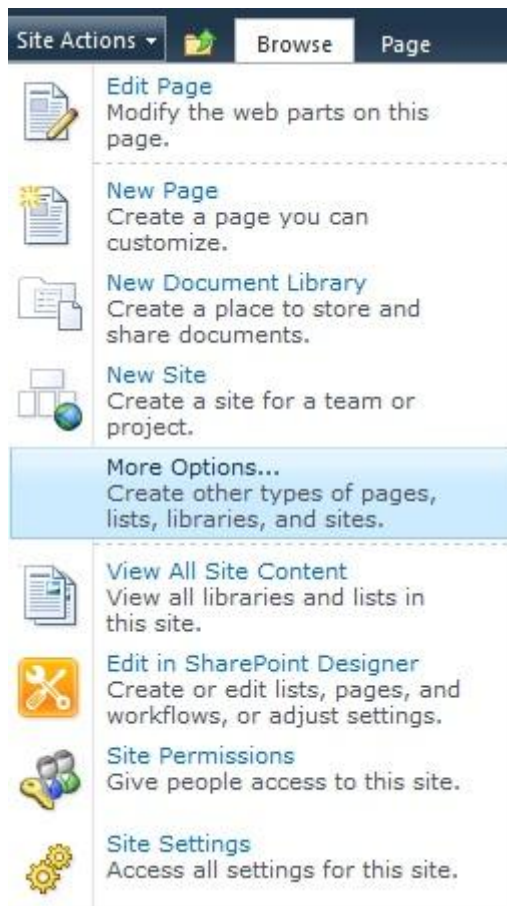
Microsoft Sharepoint 2010 on pienille ja suurille organisaatioille kehitetty monipuolinen web-sovellusalusta. Se on suosittu, koska se tarjoaa alustan, jota voidaan muokata organisaatiolle sopivaksi. Sharepoint tarjoaa myös liiketoiminnan kannalta tärkeiden tietojen esittämisen yhdessä keskeisessä paikassa. Käyttöliittymä on tuttu muista Office-tuotteista. Microsoft Sharepoint 2010 sisältää mm. seuraavia toiminnallisuuksia:

- Internet-sivustojen julkaisun ja sisällöntuotannon
- Dokumenttien hallinnan
- Työtilojen hallinnan
- Raportoinnin työkaluja
- Kalenterien- ja tehtävienhallinnan
- Laajat haut

Sharepoint tarjoaa siis jo sinällään paljon toiminnallisuutta mutta usein asiakkaiden vaatimukset ovat niin yksityiskohtaisia, että sivustojen yksityiskohtainen räätälöinti on välttämätöntä. (Noel 2010, 5-7.)

2.2.1 Uusi Sharepoint-sivu

Sovellusta kehittäessäni piti testata webpartien toimivuutta ja näin ollen piti tehdä Sharepoint-sivu johon laittaa webpartit. Uuden sivun luominen on suhteellisen helppoa yksinkertaisen käyttöliittymän vuoksi.

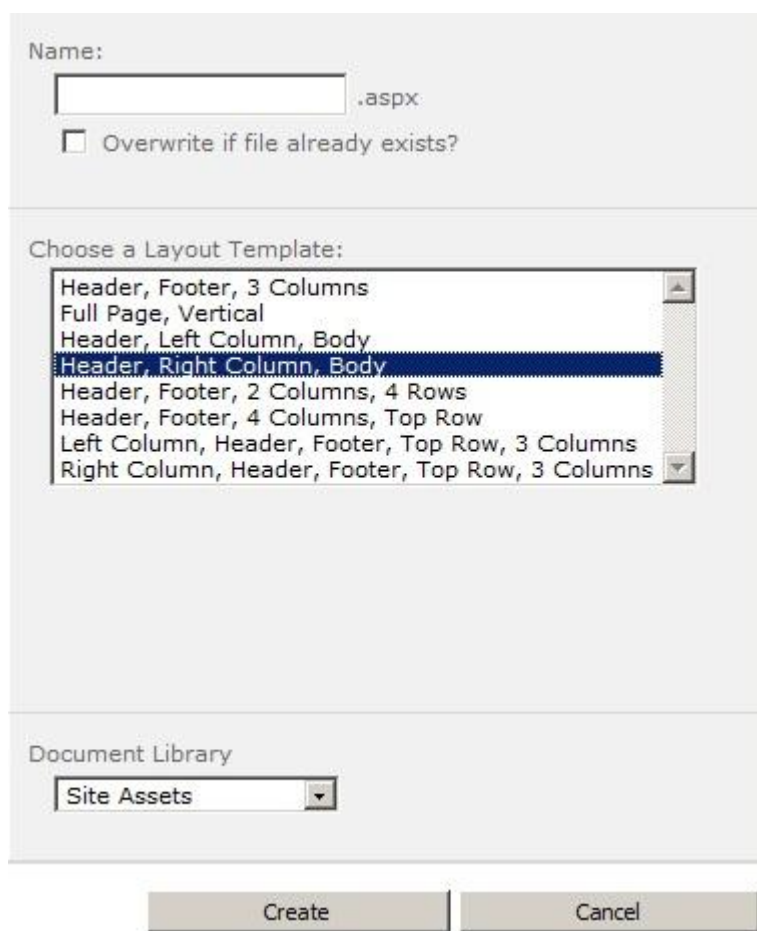


Kuva 1. Uuden Sharepoint-sivun luonti

Sivua valittaessa voidaan valita tavallinen sivu tai webpart-sivu. Tavallisellekin sivulle voi lisätä webparteja, mutta minusta sivu on helpompi rakentaa pudottamalla webpartit valmiiksi määrättyihin kohtiin. Sivulle kannattaa antaa mahdollisimman kuvaava nimi.



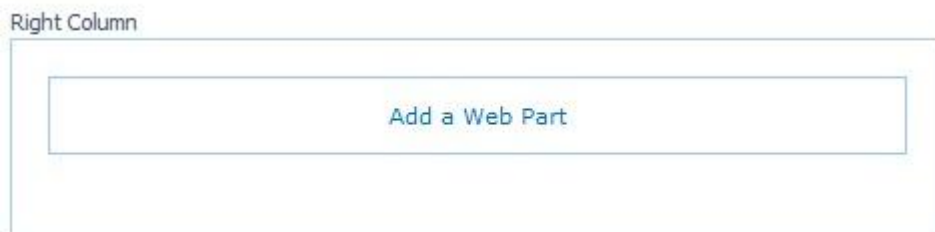
Kuva 2. Tavallinen sivu ja webpart-sivu.



Kuva 3. Sivun rakenteen valitseminen.

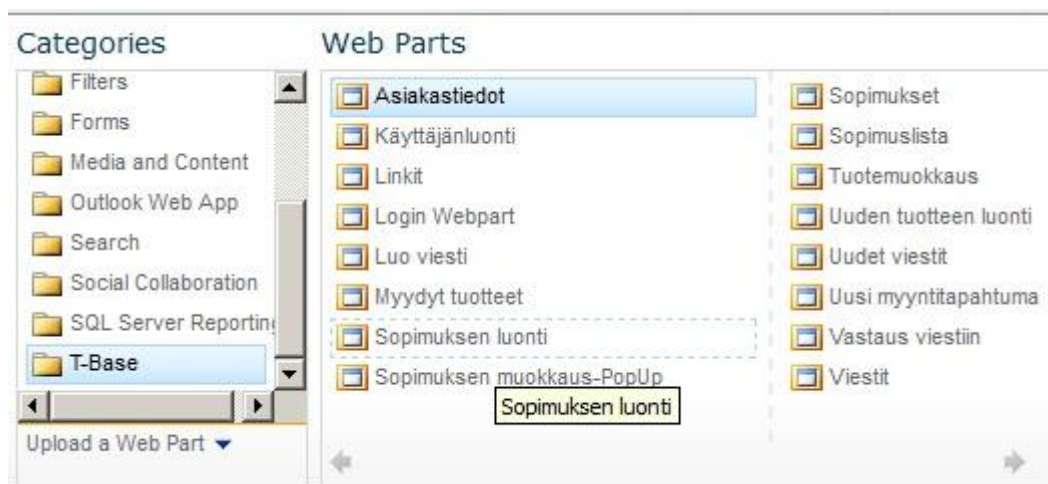
2.2.2 Sharepoint webpartin lisääminen sivulle

Webpart on www-osa joka voidaan lisätä sivuille, missä on webpartin mahdollistava alue. Sekä ilmaisia että maksullisia valmiita webpartteja on paljon olemassa. Valmiit webpartit ovat käteviä joissakin tapauksissa mutta ne eivät ole kovin muokattavia. Tämän takia on hyvä tehdä itse webpartteja asiakkaan yksityiskohtaisten toiveiden mukaan. Webpartteja voidaan käyttää yhdessä toisten webpartien kanssa. Tässä sovelluksessa esimerkiksi Login-webpart toimii kaikkien muiden webpartien kanssa yhteistyössä, koska siitä saadaan kirjautuneena olevan henkilön CustomerID.



Kuva 4. Webpartin mahdollistava alue

Yllä olevasta kuvasta päästään kohtaan, josta valitaan haluttava webpart. Seuraavassa kuvassa on lista valmiita webpartteja ja valittuna itsetehdyt webpartit.



Kuva 5. Webpartin valinta

2.3 Microsoft SQL Server

Microsoft SQL Server on monipuolinen ja kustannuksiltaan edullinen tietokantaluista joka toimii erinomaisesti erilaisten tietojärjestelmien kanssa skaalautuvuutensa ansiosta. Microsoft SQL Server on tehokas ja nopea, ja se tarjoaa monipuoliset raportointityökalut. Microsoft SQL Serverissä on myös yksinkertainen käyttöliittymä ja se toimii moitteetta yhdessä muiden Microsoft-tuotteiden kanssa.

SQL Serverillä voidaan myös itse tutkia tietokantoja ja etsiä uusia asiakasmahdollisuuksia. Yrityksen työntekijät voivat tehdä itse erilaisia analyyseja. Microsoft SQL Server on luotettava ja turvallinen ratkaisu tietokantojen hallitsemiseen. (Petkovic 2008, 3-5)

2.4 Relaatiotietokannat

Yleisesti ottaen tietokanta on joukko tietoja, joita voidaan käsitellä tietokantakielellä (esimerkiksi SQL). Tietojen hallintaan tarvitaan erityinen ohjelmisto, tietokannan hallintajärjestelmä. Esimerkkeinä Oracle, DB2, Microsoft SQL Server, MySQL ja Access.



Kuva 6. Esimerkki kahdesta taulusta

Relaatiotietokanta koostuu useista tauluista, kuten yllä Yritys ja Henkilö. Tauluissa on sarakkeita, joilla on toisistaan poikkeavat nimet, sekä rivejä. Sarakkeita ovat Yrtunnus, Nimi, Postinro, Postitmp sekä Puhelin.

Jokaisessa taulussa on tunnisteena perusavain (primary key). Kuvassa yritystaulun perusavain on Yrtunnus ja henkilötaulun perusavain on Hnro. Perusavaimen on oltava yksilöivä eli uniikki joka tarkoittaa sitä, että sarakkeessa ei riveillä saa olla samaa arvoa. Perusavain voi koostua useammastakin sarakkeesta.

Kuvassa henkilöt riippuvat yritykseen joka tarkoittaa sitä, että niiden välillä on riippuvuus. Yrityksellä on monta henkilöä, mutta henkilö kuuluu aina yhteen yritykseen. Tällaista tilannetta kutsutaan yksi-moneen-yhteydeksi. Tämän vuoksi tarvitaan viiteavaimia (foreign key). Henkilötaulussa on viiteavain Yrtunnus joka viittaa yritystaulun perusavaimeen (Yrtunnus) . On myös olemassa yksi-yhteen- ja moni-moneen yhteys.

Kuvasta näkyy, että Systeemityö Oy:lla ei ole puhelinnumeroa. Jos tietokantaan syötettäessä dataa sarakkeella ei ole arvoa, tulee sarakkeeseen merkintä NULL eli tyhjä arvo. NULL ei ole välilyönti eikä nolla vaan tuntematon arvo. (Hovi 2005, 5-10.)

Tietokannan tauluissa jokaisella kentällä on oma tyyppinsä. On tärkeää valita oikea tietotyyppi, koska se voi vaikuttaa koko tietokannan toimivuuteen ja ohjelmointiin. Taulukossa 1 esitellään yleisimmät tietotyypit.

Taulukko 1. Taulun yleisimmät tietotyypit

Tietotyyppi	Selitys
char(n)	merkkimuotoinen, kiinteämittainen
varchar(n)	merkkimuotoinen, vaihtuvamittainen
decimal (n,m)	pakattu desimaaliluku, n = kokonaispituus, m = desimaalien määrä
number (n,m)	pakattu desimaaliluku
integer, int	suuri kokonaisluku
smallint	pieni kokonaisluku
date, datetime	päivämäärä, voi sisältää myös kellonajan
boolean	arvona tosi tai epätosi

3 SUUNNITTELU

3.1 Käyttöliittymän suunnittelu

Suunnitellessa käyttöliittymää pitäisi olla jonkinlainen käsitys siitä, millaiset ihmiset tulevat tuotetta käyttämään. On hyvä tietää tulevien käyttäjien tavoitteet, tehtävät, rajoitukset, motiivit ja toimintaolosuhteet. Muuten tuotteen menestys on sattumanvaraista. Käyttäjien eroina ovat esimerkiksi ikä, ja sukupolviasiat, kokeneisuus ja toimintaolosuhteet.

Käyttäjät voidaan jakaa seuraavasti eri kohderyhmiin.

- Noviisit eli aloittelijat jotka eivät tiedä tietokoneiden käytöstä tai tietotekniikasta yleensäkin kovinkaan paljoa. Heille sovelluksen kaikkien toimintojen pitää olla mahdollisimman yksinkertaisia ja helposti opittavia. Ei myöskään saa olla liikaa vaihtoehtoja.
- Edistyneemmillä käyttäjillä on jo jonkin verran kokemusta tietokoneista ja he pärjäävät jo hieman paremmin monimutkaistenkin käyttöliittymien kanssa.
- Experteillä eli asiantuntijoilla on jo vahva kokemus tietokoneista ja he haluavat käyttöliittymässä olevan paljon vaihtoehtoja ja esimerkiksi haluavat kustomoida käyttöliittymää mieleisekseen.

(Sinkkonen 2009, 66-70.)

Tämän sovelluksen käyttäjät tulevat olemaan suurilta osin joko noviiseita tai edistyneempiä käyttäjiä. Suurin osa on kuitenkin noviiseita jonka takia käyttöliittymä on suunniteltu mahdollisimman yksinkertaiseksi ja selväksi. Ei ole liikaa vaihtoehtoja eikä nappeja. Kun käyttäjä esimerkiksi tallentaa uuden tuotteen, ilmoitetaan selvästi, että tuote on tallennettu onnistuneesti.

3.2 Tietokannan suunnittelun merkitys

Tietokanta muodostaa nykyaikaisten sovellusten perustan ja siksi on tärkeää, että tietokanta suunnitellaan ja rakennetaan hyvin. Jos edellä mainitut asiat ovat huonosti toteutettu, ei sovelluksesta tule kovinkaan onnistunutta. Jos suunnittelu ja rakentaminen toteutetaan huolella, välttyy myöhemmiltä korjauksilta ja helpottaa suuresti ohjelmointia.

Tietokantaa suunniteltaessa olisi hyvä pohtia, minkälaiseen rakenteeseen pyritään. Hyvän tietokannan tärkeitä ominaisuuksia ovat

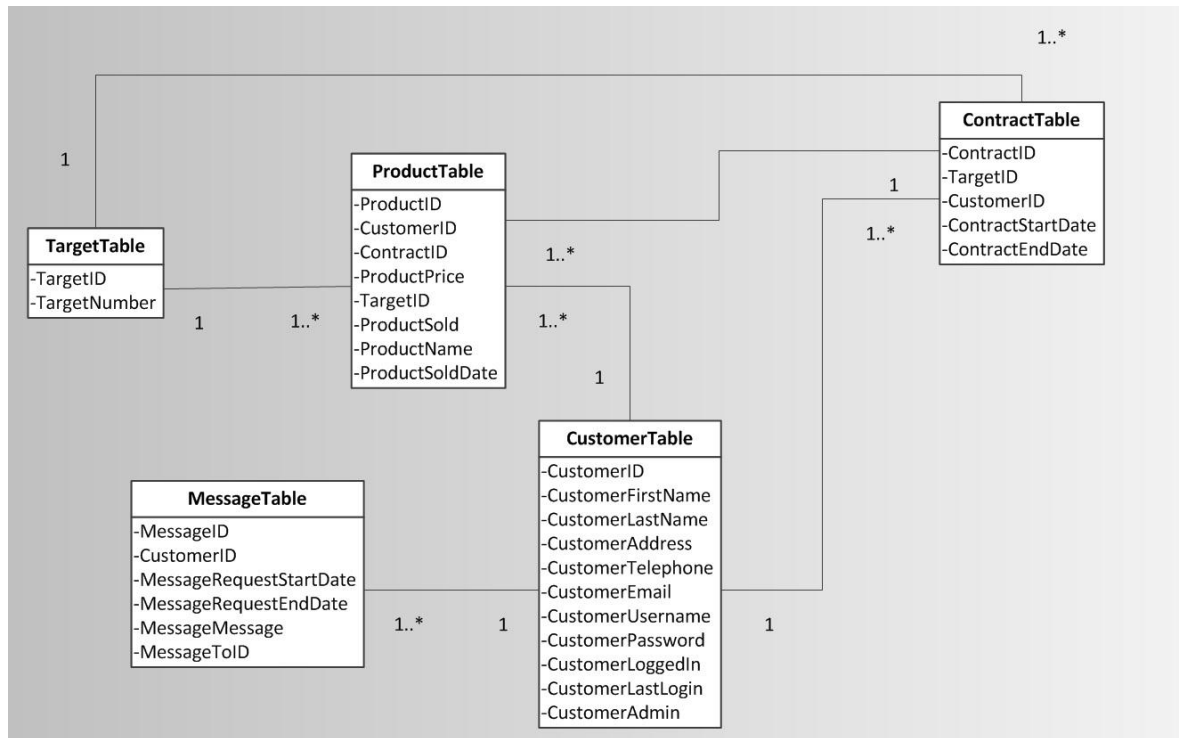
- kattavuus, eli sisältää kaikki kyselyissä tarvittavat tiedot ja yhteydet
- selkeys ja ymmärrettävyys, eli mahdollisimman yksinkertainen rakenne jolloin on helppo tehdä kyselyitä
- muutosjoustavuus, eli mahdollisimman helppo laajennettavuus minimoiden muutokset
- yleiskäyttöisyys, eli soveltuu eri asiakkaille tarvitsematta muuttaa itse rakennetta
- eheys, eli ei samaa dataa varastoida tauluihin
- suorituskyky eli tehokkuus on myös hyvin tärkeää

3.3 Tietokannan suunnittelu

Tietokantaa lähdettiin suunnittelemaan asiakkaan toiveiden mukaisesti.

Keskustelujen pohjalta koottiin lista vaadituista toiminnoista ja ominaisuuksista:

- Käyttäjätunnistus ja sisään kirjautuminen
- Pääkäyttäjän toiminnot ja ominaisuudet
 - Uuden käyttäjän lisäys
 - Uuden sopimuksen luonti
 - Tuotteiden myyntitapahtuma
 - Viestin luonti ja vastaanottaminen
 - Asiakkaan myytyjen tuotteiden yhteissumma
- Asiakkaan toiminnot ja ominaisuudet
 - Omien tietojen muokkaus
 - Uuden tuotteen lisääminen
 - Tuotteen muokkaus
 - Viestin luonti ja vastaanottaminen



Kuva 7. Tietokannan uml-kaavio.

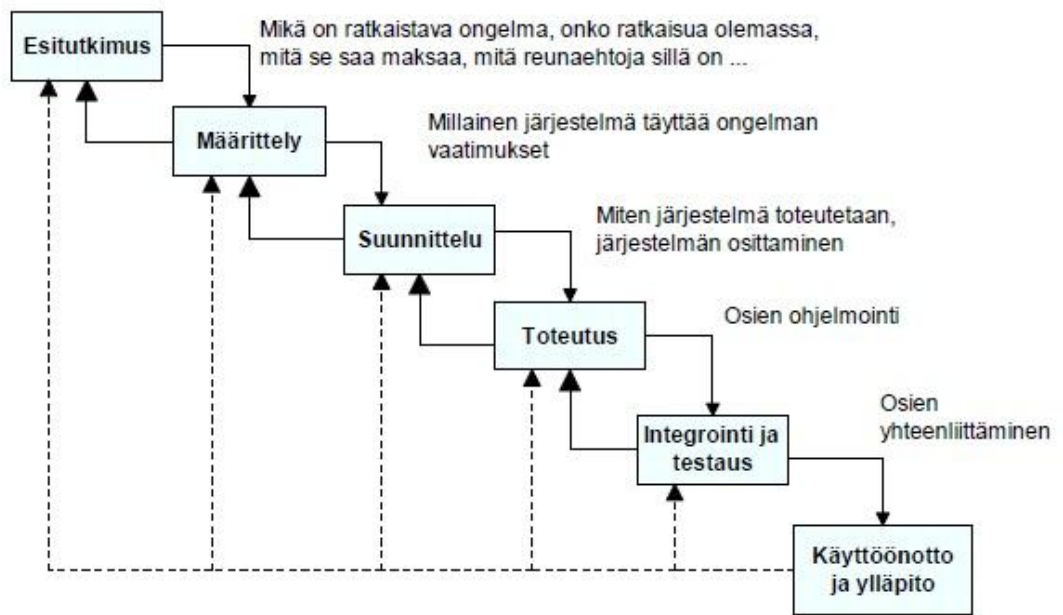
4 SOVELLUS PROJEKTINA

Lähtiessäni toteuttamaan sovellusta minulle annettiin aikaa noin kolme kuukautta. Arvio oli tehty sen mukaan, mitä kokeneempi kehittäjä arvioi itsellään menevän aikaa vastaavanlaisen sovelluksen tekemiseen. Tähän arvioon lisättiin aikaa huomioiden kokemattomuuteni sekä se, että minun pitää opetella Sharepointin käyttöä. Sovelluksen etenemistä seurattiin päivittäisillä palavereilla joissa käsiteltiin seuraavat asiat; mitä olen saanut aikaan ja mitä teen seuraavaksi. Palavereissa käsiteltiin myös kohdattuja ongelmia ja niihin ratkaisuja. Projektin aikana käytiin sovelluksen tilaajan kanssa muutama palaveri jossa käytiin läpi toimintoja ja tilaaja antoi myös joitain parannusehdotuksia. Samalla tulevat käyttäjät opettelivat hieman käyttämään sovellusta.

Sovelluksen prosessimallina käytettiin Winston W. Roycen kehittämää vesiputousmallia. Vesiputousmalli koostuu seitsemästä eri prosessivaiheesta. Vesiputousmallia noudattaessa edetään järjestyksessä vaiheesta toiseen eli kun esimerkiksi esitutkimus on suoritettu, siihen ei enää palata vaan siirrytään määritteluvaiheeseen.

Vesiputousmalli soveltuu hyvin sellaisiin projekteihin, jossa määritelmät ovat alusta asti hyvin tiedossa. Ongelmana on se, että se on hieman joustamaton uusien määritelmien tullessa esiin. Tämän vuoksi se ei ehkä ole paras vaihtoehto laajojen sovellusten tekoon.

Työssä käytin hieman muunneltua vesiputousmallia, koska tilaaja halusi uusia toimintoja ja ominaisuuksia toteutusvaiheessa. Tämän vuoksi minun piti palata takaisin prosessivaiheissa.



Kuva 8. Muunnellun vesiputousmallin prosessivaiheet

5 TOTEUTUS

5.1 Tietokannan taulut

Toteutus alkoi tietysti laittamalla ohjelmistot kuntoon, eli asennettiin ja konfiguroitiin Microsoft SQL Server. Toteutus on sujuvaa käyttämällä SQL-lauseita taulujen perustamiseen.

CustomerTablessa säilytetään asiakkaan sekä pääkäyttäjän tietoja. CustomerTablen kenttiä ovat CustomerID, joka on pääavain ja kasvaa aina yhdellä uuden asiakkaan/pääkäyttäjän luonnissa. CustomerFirstName on asiakkaan/pääkäyttäjän etunimi ja CustomerLastName sukunimi. CustomerAddress on osoite, CustomerTelephone puhelinnumero ja CustomerEmail E-mail. CustomerUsername on käyttäjän haluama käyttäjänimi ja CustomerPassword salasana joilla kirjaututaan järjestelmään. CustomerLoggedInistä nähdään, onko käyttäjä kirjautuneena sisään. CustomerLastLoginistä nähdään, milloin käyttäjä on viimeksi kirjautunut järjestelmään. CustomerAdmin määrittää, onko käyttäjä asiakas vai pääkäyttäjä.

ProductTablessa säilytetään tuotteen tietoja. ProductTablen kenttiä ovat ProductID joka on pääavain ja kasvaa automaattisesti aina yhdellä. CustomerID ja ContractID ovat vierasavaimia, jotka liittyvät CustomerTableen sekä ContractTableen. ProductPrice on tuotteen hinta, TargetID tuotteen loosi, ProductSoldista nähdään onko tuote myyty, ProductSoldDatesta tuotteen mahdollinen myyntipäivä sekä ProductName on tuotteen nimi.

ContractTablessa säilytetään sopimusten tietoja. ContractID on taulun pääavain ja kasvaa aina yhdellä lisättäessä uutta sopimusta. CustomerID on vierasavain,

joka liittyy CustomerTableen. ContractStartDate sekä ContractEndDate ovat sopimuksen aloituspäivämäärä sekä lopetuspäivämäärä.

MessageTablessa säilytetään pääkäyttäjien ja asiakkaiden viestejä. MessageID on taulun pääavain ja kasvaa aina yhdellä uuden viestin saapuessa. CustomerID on vierasavain joka liittyy CustomerTableen. MessageRequestDate ja MessageEndDate ovat asiakkaan toiveita pääkäyttäjälle sopimuksen kestosta. MessageMessage on itse viesti ja MessageToID on viestin vastaanottaja.

TargetTablessa säilytetään loosinumeroa. TargetID on pääavain joka kasvaa aina yhdellä ja TargetNumber tarkoittaa loosia.

5.2 Sovelluksen toteutus

Sovellus toteutettiin tekemällä jokaisesta halutusta toiminnosta webpart, jotka yhdistettiin myöhemmin. Eri webparteja tuli yhteensä 11 kappaletta.

5.2.1 Uuden käyttäjän lisäys

CreateNewUser-web-osalla tehdään uusi käyttäjä. Lisääjän tarvitsee olla pääkäyttäjä. Lomakkeeseen syötetään perustiedot jonka jälkeen painetaan "Tallenna"-nappia. Jos lisätään rasti "Pääkäyttäjä"-kohdassa, lisätystä käyttäjästä tulee pääkäyttäjä. Jos käyttäjänimi ei ole vielä käytössä, lisätään uusi asiakas/pääkäyttäjä tietokantaan CustomerTableen.

Etunimi:	<input type="text"/>
Sukunimi:	<input type="text"/>
Osoite:	<input type="text"/>
Puhelinnumero:	<input type="text"/>
E-mail:	<input type="text"/>
Käyttäjänimi:	<input type="text"/>
Salasana:	<input type="text"/>
Vahvista salasana:	<input type="text"/>
<input type="checkbox"/> Pääkäyttäjä	
<input type="button" value="Tallenna"/>	

Kuva 9. Uuden käyttäjän lisäys

5.2.2 Sovellukseen kirjautuminen

Sovellukseen kirjautuminen tapahtuu Login-web-osassa. Kenttiin syötetään edellä syötetyt käyttäjänimi ja salasana.

Käyttäjätunnus:	<input type="text"/>	Kirjautuneena: Antti Uusi-Seppä
Salasana:	<input type="text"/>	Kirjaudu ulos
<input type="button" value="OK"/>		

Kuva 10. Oikealla käyttäjän/pääkäyttäjän kirjautuminen ja vasemmalla käyttäjä/pääkäyttäjä kirjautuneena.

Login-web-osassa myös määritellään se, että mitä tietokantaa käytetään. Kun sovellus tulee käyttöön niin tietysti eri kirpputoreilla on eri tietokannat käytössä. Ongelma ratkaistiin tekemällä custom property web-osaan. Merkittyyyn kohtaan syötetään halutun tietokannan nimi.

← Login Webpart

[-] Appearance

Title
Login Webpart

Height
Should the Web Part have a fixed height?
 Yes Pixels
 No. Adjust height to fit zone.

Width
Should the Web Part have a fixed width?
 Yes Pixels
 No. Adjust width to fit zone.

Chrome State
 Minimized
 Normal

Chrome Type
Default

[+] Layout
[+] Advanced
[-] T-Base

InitialCatalog
 ...

OK Cancel Apply

Kuva 11. Käytetyn tietokannan valitseminen.

5.2.3 Uuden sopimuksen luonti

Kun lisätään uutta sopimusta, luojan tarvitsee olla pääkäyttäjä.

Alasvetovalikosta valitaan asiakas, jolle sopimus luodaan. Tämän jälkeen valitaan sopimuksen aloitus- ja lopetuspäivämäärä ja haetaan vapaat loosit kohdasta "Hae vapaat kohteet".

Asiakas:

Sopimuksen aloituspäivämäärä:

Sopimuksen päättymispäivämäärä:

[Hae vapaat kohteet](#)

Kuva 12. Asiakkaan ja sopimuksen päivämäärien valinta.

Tämän jälkeen aukeaa kohdeluettelo, josta nähdään vapaat ja varatut loosit. Loosit on merkitty numeroin. Jos loosi on varattavissa, eli sitä ei ole vielä varattu, se varataan kohdasta "Varaa".

Kohdeluettelo

Numero	Varattu	Varaus
1	✓	
2	✓	
3		Varaa

Kuva 13. Loosin varaus.

5.2.4 Käyttäjän sopimukset

Tällä web-osalla näytetään kirjautuneena olevan käyttäjän voimassa olevat sopimukset. Kohdasta "Muokkaa" aukeaa ponnahdusikkuna, jossa voidaan muokata sopimuksen alkamis – ja päättymispäivämäärää. Kohdasta "Lopeta sopimus", sopimuksen päättymispäivämääräksi päivitetään meneillään oleva päivä.

Sopimuslistaus

Alkamispäivämäärä	Päätymispäivämäärä	Muokkaa	Lopeta sopimus
1/18/2012 12:00:00 AM	1/26/2012 12:00:00 AM	Muokkaa	Lopeta sopimus
1/19/2012 12:00:00 AM	1/30/2012 12:00:00 AM	Muokkaa	Lopeta sopimus

Kuva 14. Käyttäjän voimassa olevat sopimukset

Alkamispäivämäärä: Päätymispäivämäärä:

Kuva 15. Sopimuksen muokkauksen ponnahdusikkuna

5.2.5 Uuden tuotteen lisäys

Alasvetovalikossa on kirjautuneena olevan käyttäjän avoimena olevien sopimusten aloituspäivämäärät. Valitaan haluttava sopimus, johon lisätään uusi tuote kirjoittamalla kenttiin tuotteen nimi ja hinta. Painetaan tallennusnappia jolloin tuote tallentuu tietokantaan.

Sopimus:

Tuotteen nimi:

Tuotteen hinta:

Kuva 16. Uuden tuotteen lisäys.

5.2.6 Lista tuotteista ja niiden muokkaus ja poisto

Alasvetovalikkoon tulee kirjautuneen olevan käyttäjän sopimukset. Valikosta valitaan sopimus, jonka tuotteet halutaan nähdä. Tuotteen tiedot tulevat näkyviin.

Kohdasta "Muokkaa" aukeaa ponnahtusikkuna josta päästään muokkaamaan tuotteen hintaa ja nimeä. Kohdasta "Poista" voidaan poistaa haluttu tuote.

1/18/2012 12:00:00 AM

Tuotelistaus

Nimi	Hinta €	Myyty	Muokkaa	Poista
Taskulamppu	12.4	✓	Muokkaa	Poista
Pöytälamppu	22	✓	Muokkaa	Poista
Teline	5	✓	Muokkaa	Poista
Kypärä	50		Muokkaa	Poista
Kirja	10		Muokkaa	Poista

Kuva 17. Tuotteen tiedot.

Tuotteen nimi:	Pöytälamppu
Hinta: (€)	22
Tallenna muutokset	

Kuva 18. Valitun tuotteen muokkaus.

5.2.7 Tuotteen myyntitapahtuma

Kun tuote tuodaan myyntiin, saa se uniikin ProductID:n josta tehdään viivakoodi. Viivakoodi kiinnitetään tuotteeseen ja viedään hyllyyn. Kun joku tuo sen kassalle ostettavaksi, tarvitsisi se jotenkin merkitä myydyksi. Päädyttiin seuraavanlaiseen ratkaisuun. Aloitetaan uusi myyntitapahtuma jolloin näkyviin tulee tekstilaatikko. Tuotteen viivakoodin voi lukea viivakoodinlukijalla jolloin tuotteen ja tuotteen myyjän tiedot tulevat suoraan näkyviin tai syöttää viivakoodin käsin ja painaa nappia ”Hae tuote” jolloin samat tiedot tulevat esiin. Tuotteen myyjän tiedot tulevat sen takia, jos tulee esimerkiksi jotain kysyttävää tuotteeseen liittyen.

<u>Aloita uusi myyntitapahtuma</u>
16
Hae tuote

Kuva 19. Tuotteen viivakoodi voidaan lukea viivakoodinlukijalla tai syöttää käsin.

Tuotteen tiedot:	
Tuotteen nimi:	Kypärä
Tuotteen hinta:	50
Tuotteen loossi:	2
Asiakkaan tiedot:	
Asiakkaan nimi:	Erkki Esimerkki
Asiakkaan osoite:	Esimerkkikatu 23
Asiakkaan E-mail:	esim@esim.com
Asiakkaan puhelinnumero	040123456
<input type="button" value="Merkitse tuote myydyksi"/>	

Kuva 20. Kun tuotteen viivakoodi on luettu lukijalla tai kirjoitettu, tiedot tulevat näkyviin.

5.2.8 Viestin lähettäminen

Asiakas halusi sovellukseen jonkinlaisen viestintäominaisuuden, jotta asiakas voisi esittää toiveita sopimuksen ajankohdasta tai vaikkapa kysyä aukioloaikoja. Päädyttiin seuraavanlaiseen ratkaisuun. Alasvetovalikosta valitaan pääkäyttäjä, jolle viesti halutaan lähettää. Kirjoitetaan aihe ja halutessaan päivämäärät sopimuksen keston. Päivämääriä ei ole pakko valita, jos kyse ei ole loosivarauksesta. Päivämäärät voi kirjoittaa, tai valita kalenterista. Napista ”Lähetä” viesti tallentuu tietokantaan.

Vastaanottaja:	Antti Uusi-Seppä
Aihe:	Loosivuokra
Sopimuksen aloituspäivämäärä:	1/25/2012
Sopimuksen päättymispäivämäärä:	2/1/2012
Viesti:	
Haluaisin varata loosin kyseiseksi ajaksi.	
Lähetä	

Kuva 21. Viestin lähettäminen.

5.2.9 Ilmoitus uusista viesteistä

Ilmoitus uusista viesteistä sijoitetaan etusivulle, jotta käyttäjä näkee heti sovellukseen kirjautuessaan onko uusia viestejä. Uudet viestit saadaan vertailemalla viestin lähetyspäivämäärä ja sitä, koska käyttäjä on viimeksi kirjautunut sisään. Eli MessageTablen kenttä MessageSentDate sekä CustomerTablen kenttä CustomerLastLogin. Lasketaan ne ja sijoitetaan linkkiin (Sinulle on x uutta viestiä), josta pääsee viestilistaan.

5.2.10 Lista käyttäjän viesteistä

Linkistä päästään tälle sivulle, jossa on lista kirjautuneen käyttäjän viesteistä. Viestit on laitettu tulemaan kannasta laskevaan järjestykseen jolloin uusin viesti tulee aina ensimmäisenä listaan. Listasta näkyy lähettäjä, aihe ja lähetyspäivämäärä. Näitä painamalla voidaan listan järjestystä muuttaa. Kohdasta "Vastaa" aukeaa ponnahdusikkuna, josta voidaan vastata viestiin. Kohdasta "Poista" viesti poistetaan.


Viestit				
Lähettäjä	Aihe	Lähetetty	Vastaa	Poista
Erkki Esimerkki	Loosivuokra	1/24/2012 12:00:00 AM	Vastaa	Poista viesti


Kuva 22. Lista käyttäjän viesteistä.

Lähettäjä:

Aihe:

Viesti: Haluaisin varata loosin kyseiseksi ajaksi.

Alkamispäivämäärä: 

Päätymispäivämäärä: 

Vastaa:

Loosi on varattu teille.

Kuva 23. Viestiin vastaaminen.

5.2.11 Käyttäjän myydyt tuotteet

Nämä nähdäkseen tarvitsee olla pääkäyttäjä. Alasvetovalikosta valitaan asiakas, jonka myydyt tuotteet halutaan nähdä. Kun asiakas on valittu, ilmestyy lista myydyistä tuotteista ja alas niiden yhteissumma.

Asiakas:

Erkki Esimerkki

Myydyt tuotteet:

Nimi	Hinta €
Taskulamppu	12.4
Pöytälamppu	22
Teline	5

Yhteensä: 39.4 €

Kuva 24. Valitun asiakkaan myytyjen tuotteiden yhteissumma.

6 YHTEENVETO

Olen suhteellisen tyytyväinen opinnäytetyöni lopputulokseen. Tilaajan toivotut toiminnot ja ominaisuudet tulivat täytettyä hyvin ja sovellus toimii vakaasti. On kuitenkin mahdotonta sanoa tässä vaiheessa sitä, että suorittaako sovellus täydellisesti tehtävänsä sillä sovellus ei ole vielä tässä vaiheessa käytössä. Jos tarvitsisi tehdä jotain toisin, voisi mainita dokumentoinnin parantamisen. Dokumentointi laahasi jonkun verran perässä koko projektin ajan. Pysin aikataulussa kohtalaisesti, vaikka haluttuja toimintoja tuli projektin aikana jonkin verran lisää.

Olen erittäin tyytyväinen C#-kielen ohjelmointitaitojen kehittymisestä sekä SQL Serverin hallinnan kehittymisestä. Opin myös paljon Sharepointista ja yleisesti tietokannan hallitsemisesta sekä suunnittelusta. Opinnäytetyö oli kokonaisuudessaan mielenkiintoinen ja haastava toteuttaa.

LÄHTEET

Hovi 2005: Tietokantojen suunnittelu ja indeksointi, Docendo Finland Oy, Jyväskylä

Järvinen 2008: Visual Studio-käsikirja, Docendo Finland Oy, Jyväskylä

Noel 2010: Microsoft Sharepoint 2010 Unleashed, Pearson Education, Indianapolis

Petkovic 2008: Beginners guide to SQL Server 2008, The McGraw-Hill Companies, USA

Sinkkonen 2009: Helppokäyttöisen verkkopalvelun suunnittelu, Kariston kirjapaino Oy, Hämeenlinna 2009