

KYMENLAAKSON AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

Mikko Korpinen

ENERGIARAPORTOINTIPORTAALI

Opinnäytetyö 2011

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Tietotekniikka

KORPINEN MIKKO	Energia raportointi portaali
Insinööri työ	28 tekstisivua
Työn valvoja	Laboratorio insinööri Marko Oras
Työn ohjaaja	Toimitus johtaja Mikko Maja
Toimeksiantaja	Profimill Engineering Oy
Syyskuu 2011	
Avainsanat	energia, raportointi, asp.net, web-ohjelmointi

Työn tavoitteena oli kehittää Profimill Engineering Oy:lle energia raportointi palvelu, jonka avulla käyttäjä voi seurata jonkin kiinteistön energiankulutusta haluamallaan aikavälillä. Sovellus hakee tiedot tietokannasta ja näyttää ne käyttäjälle raporteina. Käyttäjä voi valita hakurajauksista muun muassa näyttötaajuuden ja haluamansa aikavälin.

Ohjelmisto toteutettiin ASP.NET-tekniikalla ja raporttien tekemiseen käytettiin Crystal Reports -ohjelmistoa. Sivuston dynaamiset osat on toteutettu Ajaxin avulla ja tiedot haetaan SQL-Serverin tietokannoista. Tietokannan kehittäminen ei kuulunut työn tekemiseen vaan se saatiin valmiina Profimill Engineering Oy:ltä.

Työn lopputuloksena saatiin toimiva ohjelmisto, johon yrityksessä oltiin tyytyväisiä. Kehitettävää jäi ulkoasuun, optimointiin ja täydellisempään testaukseen. Ohjelmistoon liittyvät muut asiat, esimerkiksi tiedon keräys kantaan ja tiedon jatkokäyttäminen ovat vielä kehityksen alla, joten ohjelmistoa ei päästy testaamaan sen oikeassa ympäristössä oikealla datalla.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Information Technology

KORPINEN MIKKO	Energy Reporting Portal
Bachelor's Thesis	28 pages
Supervisor	Marko Oras, Laboratory Engineer
Instructor	Mikko Maja, Managing Director
Commissioned by	Profimill Engineering Oy
September 2011	
Keywords	energy, reporting, asp.net, web programming

The goal was to develop energy reporting service for Profimill Engineering Oy. Energy reporting service allows users to monitor the energy consumption in a building for a desired time interval. The application retrieves the data from the database and displays them in reports. From the user interface, the user can select the desired time interval and some other search parameters.

The software was made the ASP.NET technology and the reports were made with the Crystal Reports software. The dynamic parts of the site were made by using Ajax and the data are in the SQL Server databases. Database development was not part of the work.

The result of the work was software which the company was pleased with. Future development parts in the software are layout, optimization and more complete testing. Software related to other issues, such as data collection and further data use, is still under development, so the software did not reach the right to test the real data.

ALKUSANAT

Tämä insinöörityö on tehty Profimill Engineering Oy:lle. Työn tavoitteena oli tehdä energiaraportointipalvelu.

Työn ohjaajana toimi laboratorioinsinööri Marko Oras ja yrityksen edustajana toimitusjohtaja Mikko Maja. Haluan kiittää heitä onnistuneesta yhteistyöstä. Lisäksi erityiskiitoksen ansaitsee Profimill Engineering Oy:n ohjelmistosuunnittelija Niko Sipilä, jolta sain paljon hyödyllistä tietoa ja ohjeita työtä tehdessä.

Kotkassa 6.9.2011

Mikko Korpinen

SISÄLLYS

1.	JOHDANTO.....	7
1.1	Taustaa.....	7
1.2	Tavoite.....	7
1.3	Energiaraportointi-ohjelmiston käyttötarkoitus	7
1.4	Yritysesittely	8
2.	TEORIA	8
2.1	Energian käyttö	8
2.2	Ohjelmiston määrittely	9
2.3	Sivuston suunnittelu ja käytettävyys	10
2.4	Ohjelmiston testaus	10
2.5	Ohjelmiston turvallisuus	11
2.6	Työkalut.....	11
2.6.1	Microsoft Visual Studio	12
2.6.2	Microsoft SQL Server.....	12
2.6.3	SAP Crystal Reports	13
2.6.4	Internetselaimet	14
2.7	Toteutustekniikat.....	14
2.7.1	.NET.....	14
2.7.2	Ajax.....	16
2.7.3	T-SQL.....	17
3.	TOTEUTUS	17
3.1	Työn aloitus ja haasteet	17
3.2	Ohjelmiston suunnittelu	18
3.3	Ohjelmiston toteutus.....	19
3.4	Testaus.....	19
4.	RAPORTOINTIOHJELMISTO	20
4.1	Käyttöliittymä	20
4.1.1	Menu	20
4.1.2	Energiankulutus yhteenveto -raportin hakuvalinnat	21
4.1.3	Energiankulutus lajeittain -raportin hakuvalinnat.....	21
4.1.4	Energiankulutus kohteittain -raportin hakuvalinnat.....	22
4.2	Energiaraportit.....	22
4.2.1	Energiankulutus-yhteenveto	23

4.2.2	Energiankulutus-lajeittain	23
4.2.3	Energiankulutus-kohteittain	24
4.3	SQL	25
4.4	Ohjelmisto.....	25
4.5	Käyttöohjeet	26
5.	YHTEENVETO JA LOPPUPÄÄTELMÄ.....	26

1. JOHDANTO

1.1 Taustaa

Projektin toteutuksen ohjauksesta vastasi Mikko Maja Profimill Engineering Oy:ltä sekä Marko Oras Kymenlaakson ammattikorkeakoulusta. Lisäksi apua tekniseen toteutukseen saatiin Profimill Engineering Oy:n tuotekehityksestä. Työn tuloksia tullaan hyödyntämään kiinteistöjen ja teollisuuden energiaraportointiratkaisuissa.

1.2 Tavoite

Työn tavoitteena oli kehittää teollisuuskohteiden ja kiinteistöjen energianseurantaan soveltuva raportointityökalu.

Työ jakaantuu seuraaviin osiin:

- Raportoinnin tietokantaliittynän, tietosisällön, käyttötoimintojen, käytettävyyden ja ulkoasun suunnittelu.
- Raportoinnin toteuttaminen ASP.NET-tekniikalla ja hyödyntämällä SAP Crystal Reports -raportointiohjelmistoa.

Tietokantarakenne ja muut määrittelyt, esimerkiksi raporteilla näytettävät tiedot, tulivat Profimill Engineering Oy:ltä.

1.3 Energiaraportointi-ohjelmiston käyttötarkoitus

Energianraportointiohjelmisto on tarkoitettu pääasiassa toimitiloihin ja sen tarkoituksena on helpottaa energiankulutuksen seuranta. Myöhemmässä vaiheessa ohjelmisto voidaan kytkeä mahdollisesti energiankulutuksen ohjausjärjestelmään, mikä mahdollistaa energiankulutuksen optimoimisen erittäin tarkasti. Nykymaailmassa energiankulutus on tärkeä asia ja huomioidaan useasti, joten tämän kaltaiselle ohjelmistolle on tarvetta ja markkinoita. (1)

Ohjelmistoa ei voida kytkeä suoraan mihin tahansa järjestelmään tai energiankulutuskohteeseen, koska se tarvitsee datankeruujärjestelmän ohjelmiston tietokantaan. Tässä työssä ei kuitenkaan keskitytty datankeräykseen vaan sen hoitavat yrityksen muut henkilöt.

Ohjelmiston eräs tärkeimmistä käyttöhyödyistä voi olla epäkuntoisen koneen huomaaminen raporteista. Jokin kohde voi alkaa kuluttaa energiaa täysin turhaan epäkuntoisuuden vuoksi. Tämä voidaan huomata raportista epätavallisena piikkinä.

1.4 Yritysesittely

Profimill Engineering Oy on kotkalainen vuonna 2010 perustettu yritys. Yritys toimittaa teollisuudelle tietojärjestelmäkokonaisuuksia tuotannon ja logististen prosessien tehostamiseen. Yritys on keskittynyt tuotannon- ja valmistuksenohjaukseen, tiedonkeruu- ja raportointiratkaisuihin, energiankäytön hallintaan, tietojärjestelmien integrointipalveluihin ja asiakaskohtaisten ohjelmistojen valmistamiseen. Profimill Engineering Oy työllistää neljä henkilöä.

2. TEORIA

2.1 Energian käyttö

Suomessa kulutetaan energiaa noin 408 TWh vuosittain (vuosi 2007). Energiankulutus voidaan jakaa muun muassa neljään osaan teollisuuden, rakennusten lämmittämisen, liikenteen ja muiden asioiden suhteen. Näistä teollisuus vie 50 %, rakennusten lämmittäminen 20 %, liikenne 14 % ja muut 16 %. Energiankulutus on kasvanut vuosien 1990 - 2007 aikana noin 1,7 % vuotta kohden. Vastuullinen kuluttaja pysyy kohtuudessa kaikkien energiamuotojen käytössä ja säästää sähköä sekä vertaa sitä muiden samanlaisten kohteiden kulutuksiin. Energian säästö on tarpeellista myös ympäristön kannalta. Esimerkiksi hiilidioksidipäästöjen vähentäminen on tarpeen ja onnistuu, kun puuttuu kaikkeen asumisen ja liikenteen energian käyttöön. (2, 6-16)

Tehokkaimpia lämmitysenergian säästökeinoja uudehkoissa toimistorakennuksissa ovat rakennusten ilmanvaihtojärjestelmien ja niiden säädön kehittäminen, rakennusten eristystason parantaminen ja rakennusten valokatteisten tilojen hyödyntäminen rakennusten ilmanvaihdossa. (3, 189) Näistä ilmanvaihtojärjestelmän kehittäminen ja erityi-

sesti säädön kehittäminen liittyvät tähän työhön. Energiaraporteista voidaan saada tärkeää tietoa ilmanvaihtojärjestelmän laitteista, jos ne ovat kytkettyinä raportointiohjelmistoon.

2.2 Ohjelmiston määrittely

Määrittely on ohjelmiston toimintojen ja ominaisuuksien kuvaamista. Pohjana määrittelylle toimivat ohjelmistolle annetut tavoitteet. Niiden tulisi olla mahdollisimman tarkkoja, ymmärrettäviä ja hyväksyttäviä. Ohjelmistoprojektissa määritellään, suunnitellaan, toteutetaan ja toimitetaan valmis ohjelmisto. Usein kyseiset vaiheet suoritetaan syklisesti, ohjelmisto jaetaan pienempiin osa-alueisiin ja ne toteutetaan vaiheittain. (4)

Määrittely voidaan jakaa esimerkiksi perinteiseen määrittelyyn ja minimaaliseen määrittelyyn. Perinteisellä määrittelyllä tarkoitetaan ohjelmiston tarkkaa määrittelyä pienimpienkin yksityiskohtien suhteen. Sen ongelmia ovat muun muassa turha määrittelytyö, vanhentuminen, tehottomuus ja liian rajoittunut suunnitelma. Määrittelyssä voidaan siis hukata aikaa ja sitoa ohjelmoitsijaa turhaan. Useasti ohjelmistoon tulee muutoksia toteutuksen yhteydessä ja nämä muutokset muuttavat määrittelydokumenttia nopeasti vanhentuneeksi, joten sen päivitys aiheuttaa joskus paljonkin työtä. Etuina tarkassa määrittelyssä on se, että kokoajan tiedetään, mitä tehdään ja miltä lopputulos tulee näyttämään. (5, 323 - 325)

Minimaalinen määrittely koostuu vähimmäisestä tietomäärästä, joka tarvitaan tuotteen kuvailuun mielekkäästi. Sen ongelmia voi olla avainvaatimusten unohtaminen, epäselvät tai mahdottomat tavoitteet ja yliviimeistelyn mahdollisuus. Etuina minimaalisessa määrittelyssä ovat parempi moraalinen ja motivaatio, tilannekohtainen tehokkuus, vähemmän turhaa työtä ja lyhyempi vaatimusvaihe. Minimaalinen määrittely vapauttaa ohjelmoitsijan toimimaan mahdollisimman nopeasti suunnittelun suhteen. (5, 326 - 327)

2.3 Sivuston suunnittelu ja käytettävyys

Käyttäjää kiinnostaa pääasiassa internetsivuston sisältö, joten sivuston navigointivälineet eivät saisi tuhlata suurinta osaa sivuston näkymästä. Käyttöliittymän tulee kuitenkin olla suunniteltu siten, että käyttäjä löytää sen avulla helposti haluamallensa sivulle tai haluamansa tiedon. Käyttäjän käyttämää näytön kokoa ja resoluutiota on vaikea arvata, joten sivusto tulisi suunnitella siten, että se toimii hyvin resoluutiosta riippumatta. Kirjasinkoko vaihtelee myös käyttäjien kesken, jonka takia sivusto tulisi testata erilaisilla kirjasinkoon arvoilla. (6, 18 - 29)

Eräs tärkeimmistä asioista www-palveluissa on vasteaika. Vasteajan tulisi olla mahdollisimman pieni, jotta käyttäjä kokee palvelun käyttämisen mielekkääksi. Monet käyttäjät haluavat tulostaa sivun, joten suunnittelussa täytyy pyrkiä siihen, että kaikista tärkeimmistä dokumenteista on tarjolla tulostettava versio. (6, 42 ja 6, 92 - 94)

Kehykset eivät ole hyvä ratkaisu sivustoille, joten niiden käyttämistä tulisi välttää. Kehyksien avulla sivu ei ole vain yksi www-sivu vaan se koostuu useammasta sivusta. Kehykset aiheuttavat muun muassa sen, että sivuston laittaminen kirjainmerkkeihin hankaloituu, URL-osoite ei toimi oletetulla tavalla ja kehykset voivat toimia suurella näytöllä hyvin, mutta pienellä ne taas vievät turhaa tilaa muulta sisällöltä. (6, 85 - 86)

2.4 Ohjelmiston testaus

Ohjelmiston virheiden eli bugien torjumiseksi vaaditaan testaamista. Jos ohjelmisto ei toimi oikein, voi tyytymätön asiakas vaihtaa parempaan. Testauksessa on kyse siitä, että tuotetta verrataan ohjelmiston vaatimusmäärittelyyn, joka kertoo mitä ohjelma tekee ja ketä se palvelee. Perinteisesti ohjelmoijat tekevät ohjelman ja kokeilevat sen toimintaa omilla koneillansa sekä laittavat kehitysversion testaukseen ulkopuolisille testaajille. Automatisoidulla testityökalulla voidaan löytää virheitä ja tarkastella ohjelmiston suorituskykyä. Testaus voi perustua myös monien vapaaehtoisten suorittamaan panokseen, esimerkiksi bugien keräykseen voidaan käyttää Mozilla-yhteisön webpohjaista Bugzilla-työkalua. (7)

Testaus aiheuttaa kustannuksia ja raha ratkaisee monessa projektissa. Virheetöntä tuotetta ei ole olemassa, joten raja joudutaan vetämään aina johonkin kohtaan testauksesakin. Käyttöympäristö vaikuttaa testaukseen, yrityksen sisäinen työkalu ei vaadi välttämättä niin suurta testausta kuin sovellus, jonka varassa on ihmishenkiä tai rahaliikennettä. Yksikin bugi voi vaikuttaa käyttökokemukseen ratkaisevasti, jos ohjelma ei toimi ollenkaan tai siinä on suuri tietoturvaongelma, se vaihtuu helposti ensimmäisellä käyttökerralla. (7)

Käytettävyys on osa ohjelmiston laatua ja se tulisikin testata jo ohjelmiston suunnitteluvaiheessa. Jos käytettävyyttä testataan ohjelmiston toteutuksen loppuvaiheessa, joudutaan useasti toteamaan, ettei muutoksien toteuttamiseen ole enää resursseja. (7)

2.5 Ohjelmiston turvallisuus

Käytännössä kaikki tietokoneet ovat yhteydessä toisiinsa nykyisellä internet-aikakaudella. Koneita ja ohjelmistoja vastaan tehdään useasti hyökkäyksiä, internet on niin sanotusti vihamielinen ympäristö. Ohjelmoitsijan tulisikin suunnitella kaikki koodi kestäväksi hyökkäyksiä. Sovellusta tehdessä tulisi ajatella, että ohjelma ajetaan mahdollisimman vihamielisessä ympäristössä. Koskaan ei saisi olettaa, että ohjelma ajetaan vain muutamassa tiedossa olevassa järjestelmässä. Sovelluksen turvallisuus tulisi ottaa huomioon alusta asti, jälkikäteen tehdyt turvallisuuskorjaukset ja turvallisuuden huomioiminen ovat aina kalliimpia ja haastavampia. (8, 3 - 19)

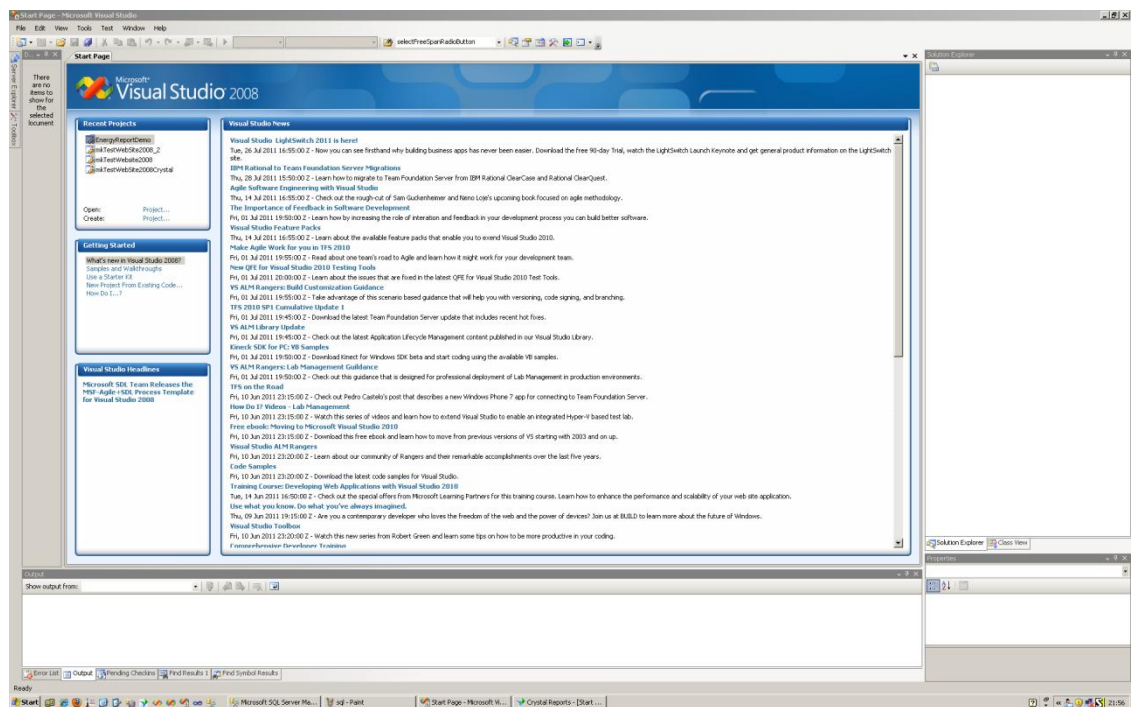
2.6 Työkalut

Työn toteutuksessa käytettiin Profimill Engineering Oy:n tarjoamaa kannettavaa tietokonetta Windows 7 käyttöjärjestelmällä. Ohjelmointityö tapahtui kuitenkin Windows XP -käyttöjärjestelmällä, jota ajettiin virtualisoinnin kautta. Kehitystyökaluina käytettiin seuraavia ohjelmistoja: Visual Studio 2008, SQL Server 2008 ja Crystal Reports 2008. Testaus tapahtui Internet Explorer 8.0 ja Mozilla Firefox 4 -selainten avulla.

2.6.1 Microsoft Visual Studio

Microsoftin Visual Studio on ohjelmointityökalu, joka yhdistää ohjelmiston kehityksen kaikki vaiheet suunnittelusta toteutukseen. Visual Studion avulla voi kehittää ohjelmia moniin eri tarkoituksiin ja ympäristöihin, esimerkiksi Windows Phone, SharePoint, Windows ja internet. Kehitystyökalusta löytyy paljon eri versioita erilaisille yrityksille tai opiskelijoille. (9 ja 10, 4)

Kuvassa 1. on työssä käytetyn Visual Studion perusnäkö.

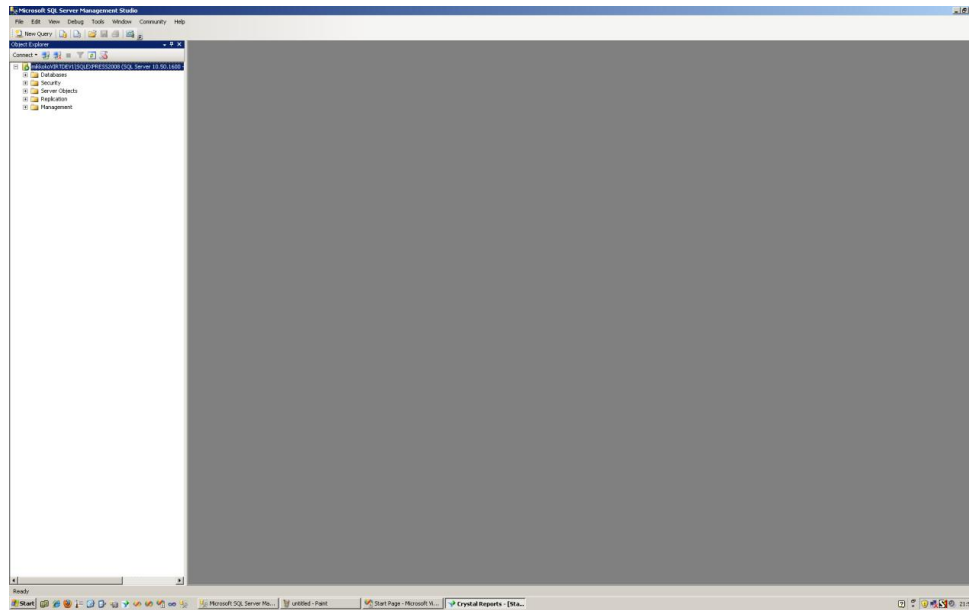


Kuva 1. Visual Studio 2008.

2.6.2 Microsoft SQL Server

Microsoft SQL Server on relaatiotietokantapalvelin, jossa käytetään T-SQL kieltä kyselyiden suorittamiseen. SQL Server 1.0 on julkaistu jo vuonna 1989 ja uusin versio SQL Server 2008 R2 vuonna 2010. (11)

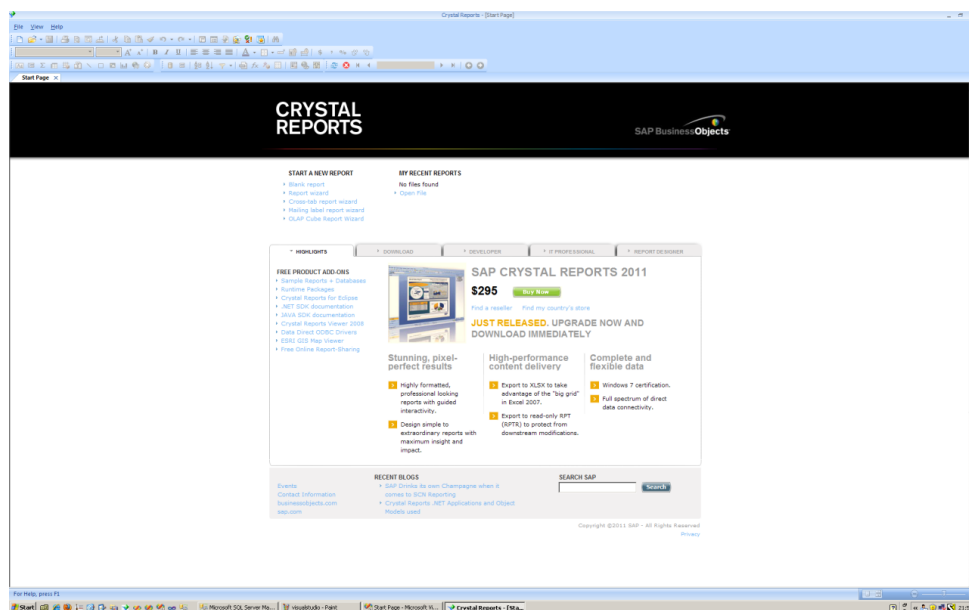
Kuvassa 2. on työssä käytetyn SQL Serverin perusnäkö.



Kuva 2. SQL Server 2008 R2.

2.6.3 SAP Crystal Reports

SAP Crystal Reports on ohjelmisto, jolla voi luoda ja katsella tehokkaasti raportteja, ja tiedot raportteihin voidaan hakea muun muassa tietokannoista (12). Crystal Reports on suosittu raportointiohjelmisto, koska se on pultattu jo Visual Studio 2003 ympäristöön (13). Kuvassa 3. on käyttämäni Crystal Reportsin perusnäky.



Kuva 3. Crystal Reports.

2.6.4 Internetselaimet

Ohjelmiston loppukäyttäjä katselee raportit internetselaimen avulla, joten käytännössä kunnollinen testaus tarkoittaisi jokaisella erilaisella selaimella testaamista. Tässä työssä tällaiselle testaamiselle ei kuitenkaan koettu tarvetta, koska tavoitteena oli, että palvelu toimii Internet Explorer selaimella. Työtä tehdessä käytettiin Internet Explorer 8.0 ja Mozilla Firefox 4 -selaimia.

Internet Explorer on Microsoftin tekemä selain, jonka historia yltää aina vuoteen 1995 asti. Silloin julkaistiin versio 1.0 Windows 95:n mukana. Nykyisin on käytössä vielä Internet Explorer 6.0 versiota, joka on julkaistu vuonna 2001. Uusin versio on vuoden 2011 Internet Explorer 9.0. Jokainen versio on tuonut uusia ominaisuuksia ja pitänyt selaimen käytetyimpien joukossa. (14)

Mozilla-projekti luotiin vuonna 1998 ja monien vuosien sekä versioiden jälkeen Mozilla 1.0 julkaistiin vuonna 2002. Mozilla Firefox 1.0 julkaistiin 2004 ja siitä tuli heti iso menestys. Uusin versio selaimesta on Firefox 6. (15 ja 16)

2.7 Toteutustekniikat

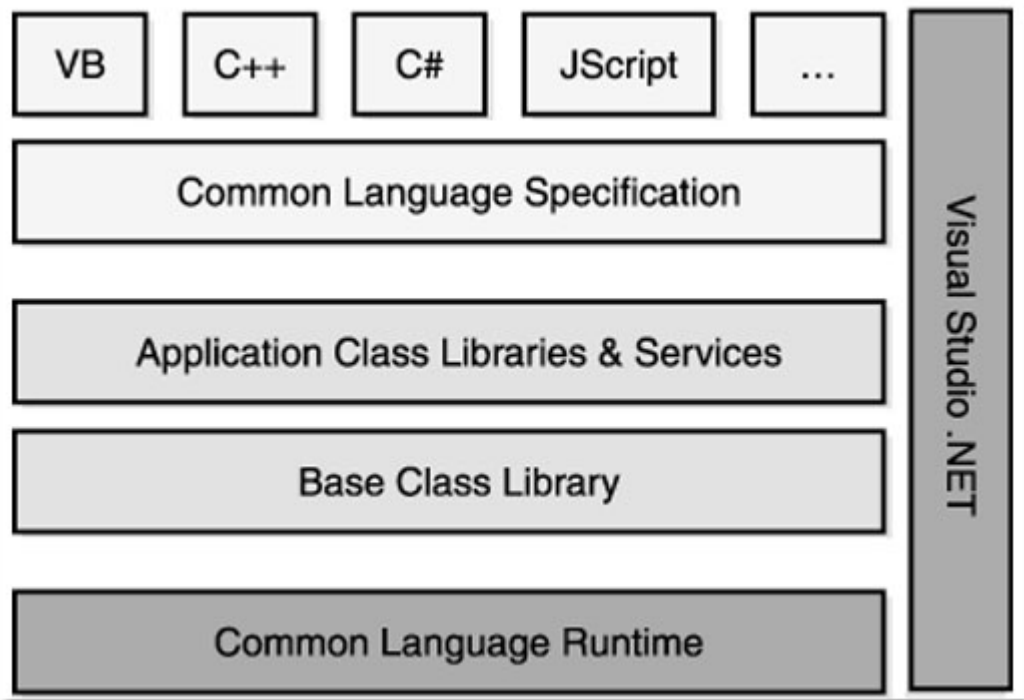
2.7.1 .NET

Microsoft .NET –arkkitehtuuri yhdistää eri sovellusalueiden tekniikat ja palvelut yhteisen katon alle ja sen tavoitteena on tarjota sovelluskehittäjille yhteinen ympäristö, jonka eri osat toimivat keskenään yhdessä ilman erillisiä rajapintoja. Ydin koostuu kahdesta pääosasta .NET-arkkitehtuurissa, Common Language Runtime (CLR) ja .NET Framework Class Library, joita on esitelty tarkemmin seuraavissa kappaleissa. (17, 4)

CLR toimii .NET-ympäristön perustana tarjoten palvelut ajettavien ohjelmien taustalla. CLR:n tehtävänä ovat muun muassa ohjelmasäikeiden hallinta, ajonaikainen turvallisuus ja huolehtiminen yleisestä käännetyn ohjelmakoodin ajamisesta. CLR irroitaa käytettävän ohjelmointikielen suorituksesta, joten se mahdollistaa monien eri kielten käyttämisen. Vaihtoehtoina ovat muun muassa VB.NET, C#, Perl, Python ja C++. Suosituimmat kielet ovat kuitenkin VB.NET ja C#. (17, 4)

.NET Framework Class Library on kokoelma yleiskäyttöisiä luokkia. Niiden avulla sovellukset saadaan kytkettyä .NET ympäristöön. Esimerkkinä voidaan mainita System.XML, josta löytyvät tarpeelliset metodit ja ominaisuudet XML-datan käsittelyyn. (17, 5)

Kuvassa 4. on yksinkertaistettu periaatekuva .NET –arkkitehtuurista.



Kuva 4. .NET-arkkitehtuurin yleiskuva. (18)

2.7.1.1 ASP.NET

ASP.NET on Microsoft .NET –arkkitehtuurin eräs osa-alue, joka on tarkoitettu web-sovelluksien kehittämiseen. ASP.NET voidaan asentaa .NET:n tukemiin palvelinympäristöihin. Ympäristön asentamisen jälkeen voidaan määrittellä yleiset asetukset, mutta myös sovelluskohtaisten asetusten määrittäminen on erittäin monipuolista. Sovelluskohtaiset asetukset määritellään Web.config-asetustiedostoihin ja näitä tiedostoja voi olla järjestelmässä monia. ASP.NET-ympäristö toimii www-palvelinohjelmiston kautta, joka on Windows-tuoteperheen sovelluspalettiin kuuluva IIS-palvelinohjelmisto. (17, 2 - 34)

2.7.1.2 Visual Basic.NET

Visual Basic.NET on ohjelmointikieli, joka on käytännössä täysin uudelleen kirjoitettu Visual Basic, mutta se säilyttää kuitenkin melko pitkälti samanlaisen syntaksin ja on helposti omaksuttava kieli. Visual Basic.NET tukee kaikkea sitä mitä .NET-alusta tarjoaa. (19, 30 - 32)

2.7.2 Ajax

Ajax (Asynchronous JavaScript and XML) ei ole ohjelmointikieli vaan tapa käyttää olemassa olevia standardeja, mikä mahdollistaa sivun osan päivittämisen ilman koko sivun uudelleen lataamista. Ajax mahdollistaa siis datan siirron sivuston ja serverin välillä, joka taas luo sivusta dynaamisen. (20 ja 21, 14)

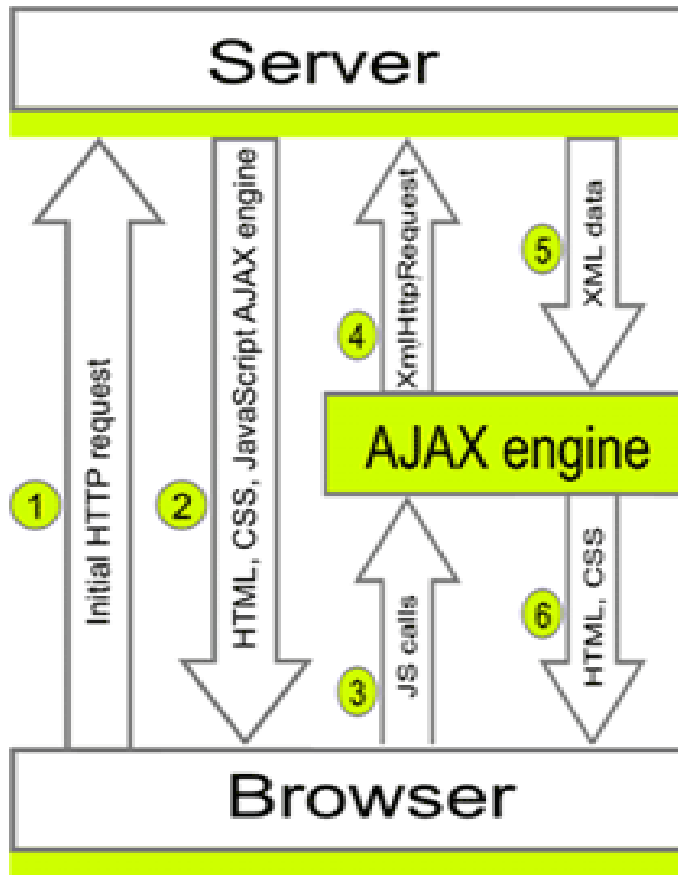
Ajaxin vahvuus on se, että se pohjautuu olemassa oleviin standardeihin ja toimii nykyaikaisissa selaimissa ilman plug-in-ohjelmia tai lisälaitteita. Merkittävin hyödyntäjä Ajaxin suhteen on ollut Google. Googlen palveluista muun muassa Gmail ja Google Maps käyttävät Ajaxia paljon hyödyksi. (21)

Ajax pohjautuu internetstandardeihin ja käyttää osaa tai kaikkia seuraavista: (20)

- XMLHttpRequest datan siirtoon
- JavaScript/DOM-informaation näyttämiseen ja vaikuttamiseen
- CSS datan tyylittelyyn
- XML yleensä siirretyn datan muotoiluun

Kuvassa 5. on Ajaxin perusidea. Seuraavassa selitykset numeroiduista kohdista: (22)

1. Pyydetään sivunlataus selaimelta, kun käyttäjä menee jollekin sivulle.
2. Koko sivu muodostetaan palvelimella ja lähetetään käyttäjälle.
3. Funktio kutsuu Ajax-moottoria.
4. Ajax-moottori tekee XMLHttpRequest-pyyntöä.
5. Palvelin käsittelee pyynnön ja lähettää vastauksen XML-muodossa. Vastaus sisältää datan vain elementeistä, joita tarvitsee muuttaa.
6. Ajax-moottori päivittää vain tietyn osan käyttäjän näkemästä sivusta.



Kuva 5. Ajaxin toiminta. (22)

2.7.3 T-SQL

Microsoft SQL Server käyttää tietokantaohjelmointiin Transact-SQL (T-SQL) –kieltä. Tyypillisesti ohjelmakoodi kirjoitetaan tallennettuihin prosedureihin tai herättimiin. T-SQL sisältää muun muassa tyypilliset kontrolli- ja silmukkarakenteet. (23)

3. TOTEUTUS

3.1 Työn aloitus ja haasteet

Työn toteutus alkoi toteutustekniikoiden opiskelulla. ASP.NET maailmasta tai SQL Serveristä ei ollut aikaisempaa kokemusta. Visual Studio oli tuttu ympäristö aikaisemman työharjoittelun ja opiskelujen vuoksi, joten siihen tutustuminen ei vienyt aikaa. Crystal Reportsista ei myöskään ollut aikaisempaa kokemusta ennen työn tekemistä.

Työn tekeminen aloitettiin ASP.NET -opiskelulla sivuston <http://www.asp.net> kautta, sivusto tarjosi loistavia ohjeita ja videoita alkutaipaleelle. Työn toteutukseen saatiin avuksi myös raportointityökaluprojekti, joka oli toteutettu Profimill Engineeringillä, joten olemassa olevaan koodiin pystyttiin tutustumaan ja tutkimaan toisen toteutuksen ratkaisuja.

ASP.NET tuli tutuksi ensimmäisen viikon aikana, joten seuraavaksi siirryttiin Crystal Reports -raporttien pariin. Raporttien luonti vaikutti helpolta lukuun ottamatta kaavioiden tekoa. Kaavioihin joutui hakemaan tiedon tietokannasta, joten tiedonhakuja kehitettiin ensin XML-tiedostosta. XML toimi kokeilussa tietovarastona, josta sai kätevästi tietoa kaavioihin.

SQL Server -ohjelmistoon tutustuminen jätettiin viimeiseksi aiheeksi. Siihen tutustuttiin aikaisemman raportointiesimerkin avulla ja demokanta luotiin energianraportointiohjelmistoa varten esimerkin kannan taulun pohjalta. Samaa taulua tulitaisiin käyttämään myös energianraportoinnissa, joten asiaan ei tarvinnut käyttää suuremmin suunnittelu-aikaa.

Viimeisenä suurena opiskelun aiheena oli kaikkien asioiden yhdistäminen ja parametrien välittäminen ASP.NET:n, Crystal Reports -raporttien ja SQL:n välillä.

3.2 Ohjelmiston suunnittelu

Raportointipalvelun suunnittelu alkoi ja eteni opiskeluvaiheessa. Soveltuvia hakuvalintalaatikoita kehitettiin useita, joista osa soveltui suoraan ohjelmistoon. Profimill Engineering Oy:ltä saatiin lisäksi periaatekuvat kuvaajista sekä selostukset hakuvalinnoista. Näitä käytettiin hyväksi toteutussuunnitelmassa ja toteutuksessa.

Ohjelman toteutuksesta tehtiin noin 20 sivua pitkä toteutussuunnitelma (Toteutussuunnitelma_EnergiaRaportointi.docx), jossa kuvataan tärkeimmät asiat ohjelman toteuttamisen kannalta ja ylläpitoa varten. Toteutussuunnitelma on Profimill Engineering Oy:n käytössä, joten sitä ei voida laittaa julkisesti nähtäville.

3.3 Ohjelmiston toteutus

Ohjelmiston toteutus tapahtui opettelu-, suunnittelun ja ohjelmoinnin aikana. Pohjana toteutukselle oli yrityksen aikaisempi raportointipalvelu, joten monia asioita löytyi valmiina tai asioihin sai ainakin hyvän mallin.

Ohjelmisto pyrittiin toteuttamaan suunnittelun pohjalta. Suunnittelu ja toteutus tapahtuivat kuitenkin osittain samaan aikaan ja suunniteltuihin asioihin tuli loppuvaiheessa vielä muutoksia. Joitakin asioita ei pystytty tekemään täysin suunnitelman mukaisesti, esimerkiksi yhteenveto-raportissa kuvaajan alle ei saatu listausta eri energiamuodoille vierekkäin, joten ne laitettiin allekkain. Kuvaajien akseleiden tekstien osalta jouduttiin tietyissä tapauksissa tekemään myönnytyksiä yksiköiden suhteen, koska kaikissa tapauksissa ei ollut mahdollista saada yksikköä tekstiin. Ongelma ratkaistiin kirjoittamalla kuvaajien akseleiden tekstien perään ”(energia)” tai ”(valuutta)”.

3.4 Testaus

Koska tässä työssä raportointipalvelun testausta ei suunniteltu etukäteen eikä toteutettu laajamittaisesti, olisi tärkeää tehdä testaus suunnitelma ja testata palvelun toiminta kattavasti. Testaus hoidettiin yksikkötestauksena toteutuksen yhteydessä ja sen jälkeen. Ohjelmisto testattiin eri hakuvaihtoehdoilla ja SQL-lauseisiin tehtiin yksinkertaiset tarkastukset tyhjän hakuparametrin varalta. Turvallisuusnäkökulmaa ei ole otettu huomioon, koska yritys ei nähnyt sille tarvetta. Ohjelmistojen turvallisuus on yksi tärkeimmistä huomioitavista asioista, joten se tulisi ottaa huomioon jatkossa.

Käyttöliittymä testattiin toiminnan osalta kokeilemalla erilaisia toimintoja ja toteamalla, että komento tuottaa halutun lopputuloksen. Raporttien summaukset halutulta aikaväliltä testattiin yksinkertaisella testidatalla minkä oikeat tulokset pystyttiin laskemaan Microsoftin Excel -ohjelmalla.

4. RAPORTOINTIOHJELMISTO

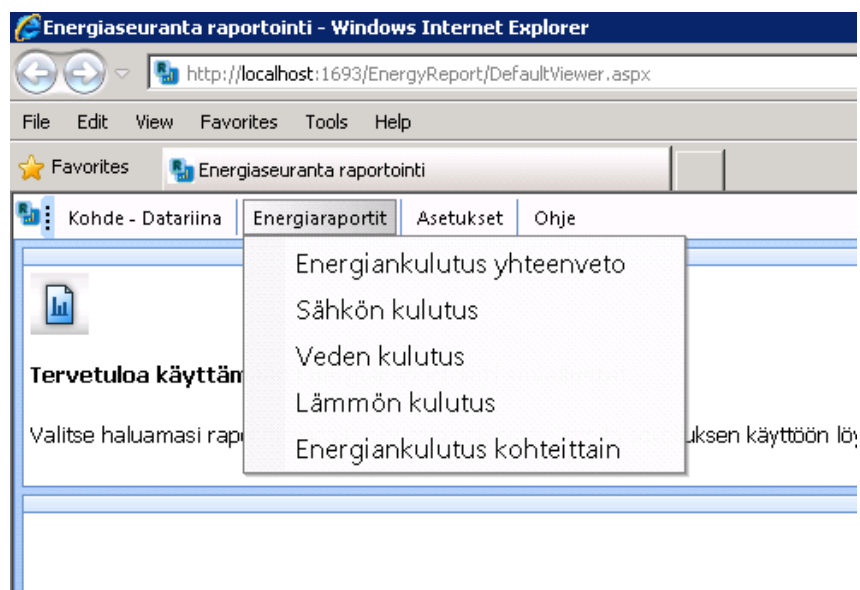
Tarkempi kuvaus ohjelmistosta löytyy dokumentista Toteutussuunnitelma_Energiaraportointi.docx. Toteutussuunnitelma on kuitenkin vain Profimill Engineering Oy:n käytettävissä, joten seuraavissa kappaleissa on selitetty ohjelmiston toiminta pääpiirteittäin.

4.1 Käyttöliittymä

Käyttöliittymä koostuu menusta ja hakurajauksista. Menun Energiaraportit-kohdasta päästään raportteihin ja eri raporteilla on erilaiset hakurajaukset. Hakurajausyhdistelmiä on kolmenlaisia: yhteenveto, lajikohtainen ja kohdekohtainen.

4.1.1 Menu

Menu on toteutettu sivuston yläosaan ja siitä voi valita kohteen, haluamansa raportin, asetukset ja ohjeet. Kuvassa 6. on avattu menu Energiaraportit-kohdasta. Valintoina ovat Energiankulutus-yhteenveto, energiankulutus jonkin energiamuodon kohdalla (lajikohtainen raportti) ja Energiankulutus-kohteittain.



Kuva 6. Menu.

4.1.2 Energiankulutus yhteenveto -raportin hakuvalinnat

Yhteenvetoraportin hakuvalinnoissa (kuva 7) on aikajaksovalinta, jolla valitaan raportin datan näyttötaajuus. Toisin sanoen, jos käyttäjä valitsee kuukausitason, näytetään raportissa palkit kuukausittain, jos valintana puolestaan on päivätaso, näytetään palkit päivittäin. Valinta vaihtaa aikarajaukset-laatikkoon tilanteeseen sopivan aikajaksovalinnan ja näyttää Ajaxin avulla tapahtumaosoittimen, mikä kertoo käyttäjälle, että valintoja päivitetään. Aikarajauksista käyttäjä voi valita esimerkiksi raportin aloittavan ja lopettavan kuukauden.

Näyttövalintalaatikosta käyttäjä voi vaihtaa raportin tilaksi joko energia tai kustannus. Energia-tila näyttää raportissa energiakohtaiset kulumat, kun taas kustannus-tila näyttää valuuttakohtaiset kulumat.

Energialajit-valinnasta voidaan valita mitä kaikkia energialajeja yhteenvedossa näytetään. Sen avulla voidaan valita yksi tai useampi energialaji näkymään raportissa.

Kuva 7. Yhteenvetoraportin hakurajaukset.

4.1.3 Energiankulutus lajeittain -raportin hakuvalinnat

Lajikohtaisen raportin hakuvalinnat ovat täysin samoja kuin yhteenvetoraportin hakuvalinnat, mutta viimeisenä valintana on kohderajaus. Kohderajauksilla voidaan valita, mitkä kohteet kyseisestä energialajista näytetään raportilla. Kuvassa 8. on näkymä hakurajauksista.

Kuva 8. Lajikohtaisen raportin hakurajaukset.

4.1.4 Energiankulutus kohteittain -raportin hakuvalinnat

Kohdekohtaisen raportin hakuvalinnat ovat täysin samoja kuin yhteenvetoraportin hakuvalinnat, mutta viimeisenä valintana on kohderajaus. Kohderajauksella voidaan valita yksi tietty kohde, joka näytetään raportissa. Kuvassa 9. on näkymä hakurajauksista.

Kuva 9. Kohdekohtaisen raportin hakurajaukset.

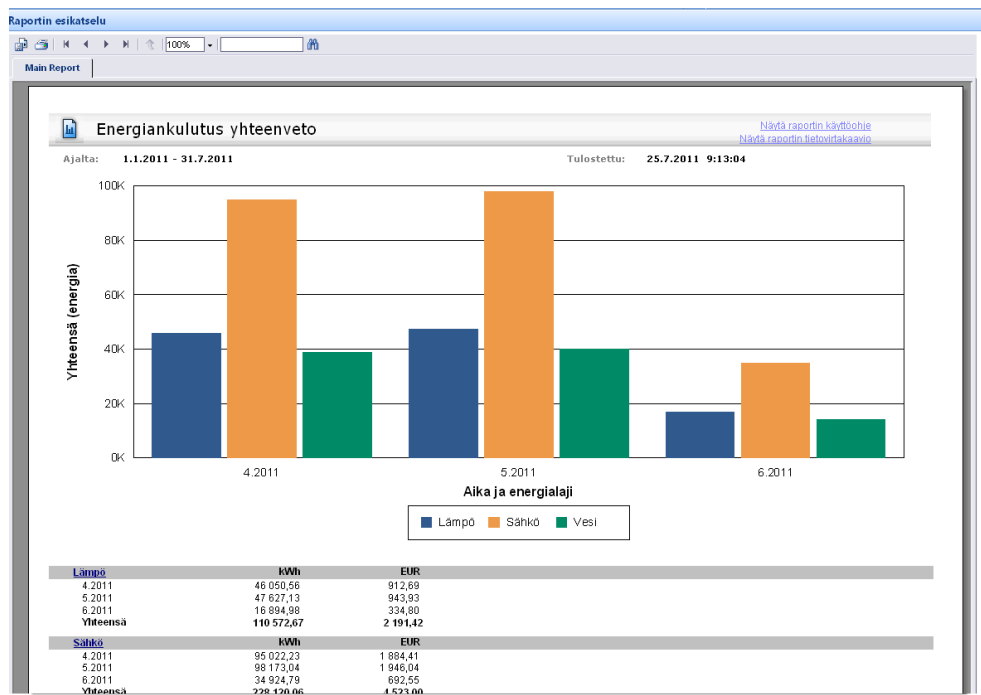
4.2 Energiaraportit

Raportit avautuvat upotettuun kehykseen (iframe), jonka ala on mahdollisimman suuri selainikkunassa. Käyttöliittymä on pyritty supistamaan pieneksi, jotta raporteille jää paljon tilaa. Raportin latautuessa näytetään Ajaxilla latautumista osoittava tapahtuma-osoitin, joka näyttää käyttäjälle, että raporttia ladataan. Crystal Reports -raporttien Ajax-toteutus aiheutti konflikteja hakurajauksien Ajaxin kanssa. Parhaimmaksi ratkaisuksi koettiin näiden asioiden erottaminen eri sivuillensa. Asioiden erottamiseen tarvittiin kehyksiä, joiden käytön haitoista ei koettu olevan tässä yhteydessä ongelmaa.

Crystal Reports -raporteissa on erillinen linkki raportin tulostamiseen. Käyttäjän ei tarvitse tulostaa raporttia selaimen tulosta-toiminnon avulla, vaan se onnistuu suoraan raportista löytyvän tulosta-painikkeen kautta.

4.2.1 Energiankulutus-yhteenveto

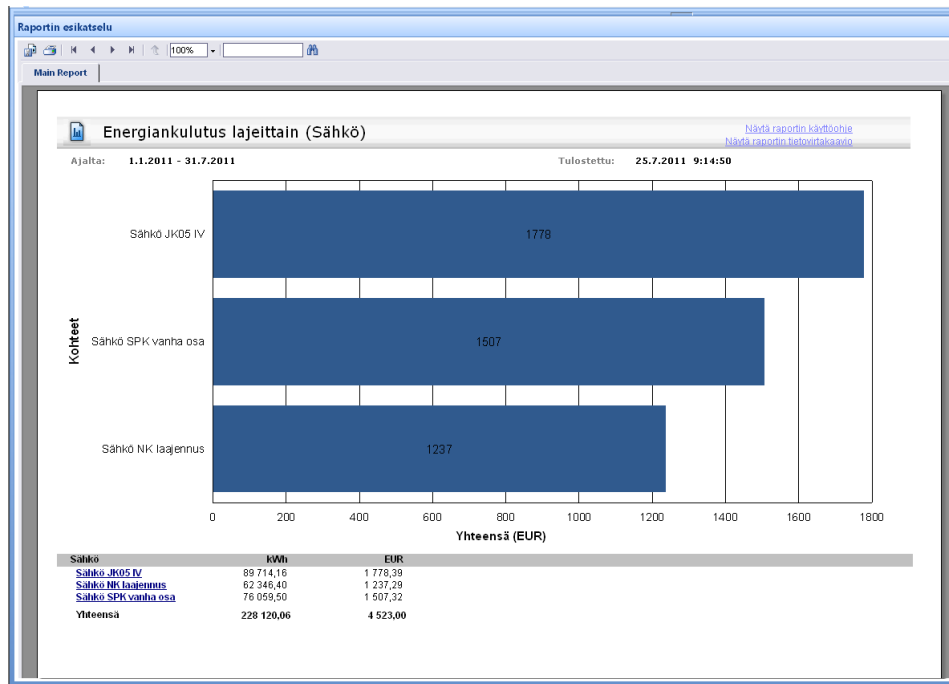
Yhteenvetoraportti (kuva 10) kerää kaikki valitut energiamuodot samaan raporttiin ja näyttää kulutuksen pystypalkkikaaviona sekä tekstimuodossa kaavion alapuolella. Kaavion alapuolisesta tekstiosuudesta löytyvät linkit eri energiamuotoihin, joten niistä voi porautua haluamaansa energiamuotoon. Linkki vie käyttäjän lajikohtaiseen raporttiin ja monistaa sinne käyttäjän valitsemat hakurajaukset.



Kuva 10. Yhteenveto-raportti.

4.2.2 Energiankulutus-lajeittain

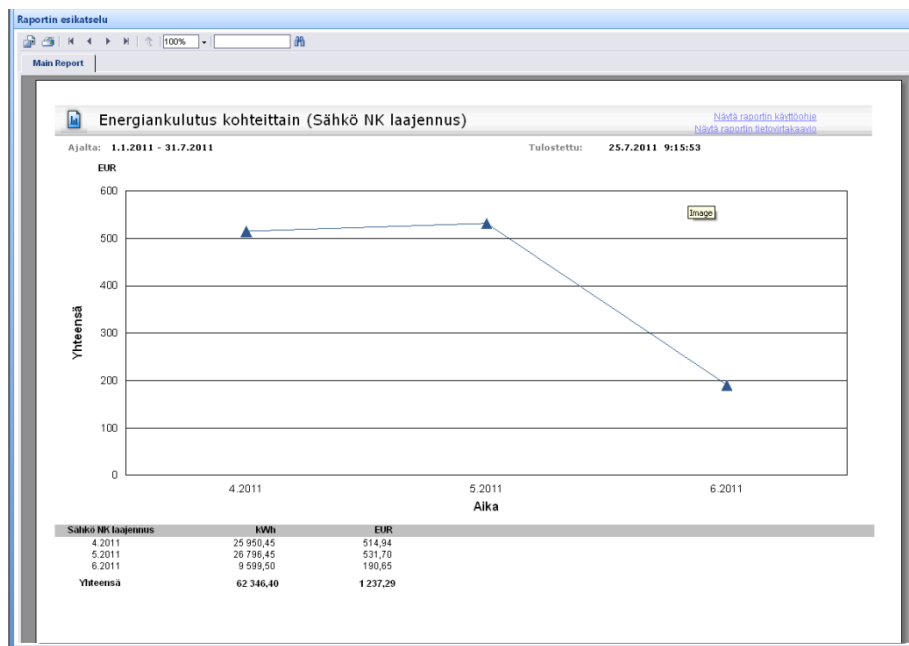
Lajeittain-raportti (kuva 11) kerää halutusta energiamuodosta valitut kohteet raporttiin ja näyttää summatun kulutuksen valitulta aikaväliltä vaakapalkkikaaviona sekä tekstimuodossa kaavion alapuolella. Kaavion alapuolisesta tekstiosuudesta löytyvät linkit kohteisiin, joten niistä voi porautua haluamaansa kohteeseen. Linkki vie käyttäjän kohdekohtaiseen raporttiin ja monistaa sinne käyttäjän valitsemat hakurajaukset.



Kuva 11. Lajikohtainen raportti.

4.2.3 Energiankulutus-kohteittain

Kohteittain-raportti (kuva 12) näyttää halutun kohteen kulutuksen trendikäyränä sekä tekstimuodossa kaavion alapuolella.



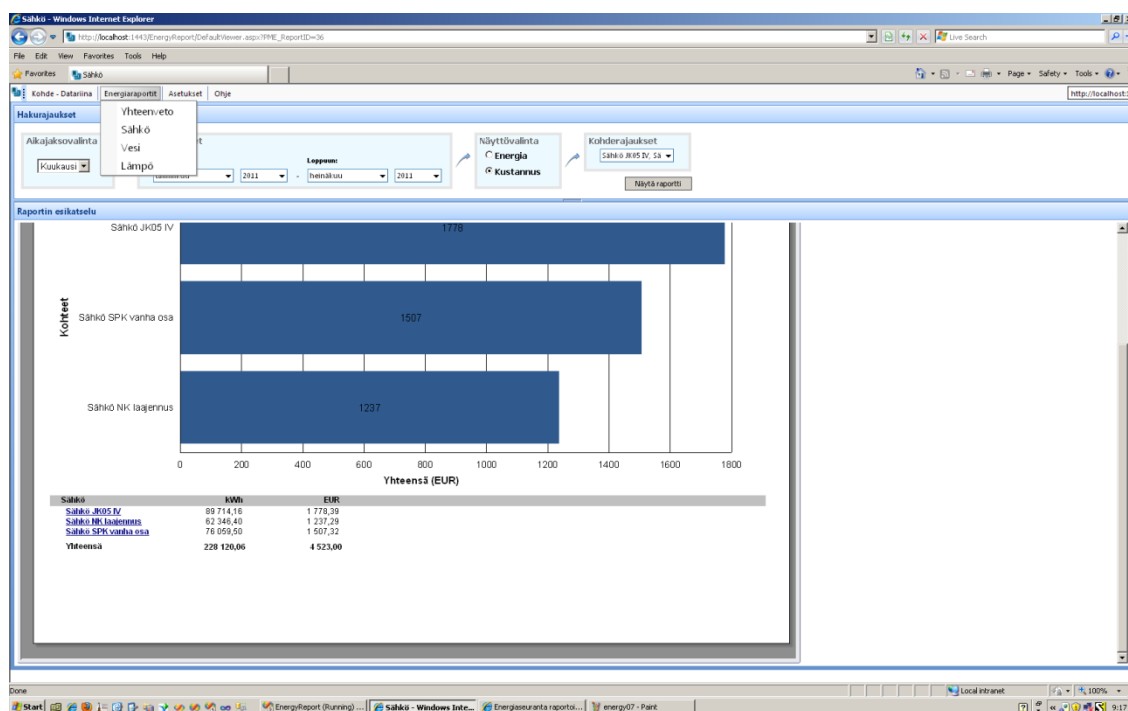
Kuva 12. Kohdekohtainen raportti.

4.3 SQL

SQL-kyselyillä hoidetaan datan hakeminen kannasta. Kaikki data on tallennettu kantaan, joten on tärkeää hakea juuri haluttu tieto ja näyttää se sitten käyttäjälle raporteissa. Myös käyttöliittymään liittyviä asioita on tallennettu kantaan, joten nämä joudutaan hakemaan käyttöliittymää generoidessa. Tietokantaosioon ei kuitenkaan keskitytä tässä enempää, vaan se on kuvattu tarkemmin Profimill Engineering Oy:n käyttöön tarkoitettussa toteutussuunnitelmassa.

4.4 Ohjelmisto

Ohjelmisto toimii yhdistäen käyttäjän valitsemat toiminnot, syöttäen haun SQL-hakuun ja yhdistämällä datan raporteihin sekä näyttämällä tulokset. Kuvassa 13. on ohjelmiston käytöstä kuva, missä käyttäjä on hakenut raportin ja on valitsemassa valikosta jonkin toiseen osioon menemistä.



Kuva 13. Yleiskuva ohjelmistosta.

4.5 Käyttöohjeet

Ohjelmistolle ja raporteille tehtiin käyttöohjeet, joista selviävät ohjelmiston peruskäyttöön tarvittavat asiat. Käyttöohjeiden kuvat on otettu testidatalla luoduista kuvaajista, joten ne tulisi päivittää paremmiksi. Oikeinkirjoitus ja käyttöohjeiden sisältö tulisi tarkistaa ja tarvittaessa kehittää eteenpäin. Raportointipalvelun käyttöohjeeseen pääsee sivuston menusta ja raporttikohtaisiin käyttöohjeisiin raporteista löytyvistä linkeistä. Käyttöohjeet on tallennettu pdf-muodossa ja ne ovat Profimill Engineering Oy:n omistuksessa, niitä ei kuitenkaan haluttu liittää tämän työn liitteisiin.

5. YHTEENVETO JA LOPPUPÄÄTELMÄ

Opinnäytetyön aihe ja toteutustekniikat eivät olleet ennestään kovin tuttuja, joten työn tekeminen osoittautui mielenkiintoiseksi uuden oppimisen kannalta. Toteutus ei mennyt suorinta tietä vaan monia asioita jouduttiin yrittämään erehdyksen kautta. Vaikeimmaksi tai haastavimmaksi asiaksi koettiin Crystal Reports sen laajuuden vuoksi.

Kehittämistä olisi raporttien ulkonäössä, SQL-lauseiden optimoinnissa, testauksessa ja turvallisuusnäkökulman huomioimisessa. Turvallisuus olisi tärkeää huomioida jokaisessa verkon palvelussa ja testauksella saataisiin varmuus, että raportointijärjestelmä toimii varmasti halutulla tavalla. SQL-lauseiden optimointi voisi parantaa tuloksien haku-aikaa huomattavasti ja raporttien miellyttävämpi ulkoasu parantaisi käyttökokemusta. Lisäksi raportointijärjestelmän yleisulkoasu tulisi uudistaa värimaailman osalta, koska tämän hetkinen värimaailma perustuu aikaisempaan raportointijärjestelmään. Vihreä väri voisi sopia hyvin energiaraportointiin. Raporttien ulkoasu sekä mahdollisesti ohjeet tulisi päivittää sivuston ulkoasun uudistamisen yhteydessä.

Työn toteutuksessa päästiin haluttuun lopputulokseen ja työ vastasi Profimill Engineering Oy:n tarpeita. Ohjelmisto ei tullut kuitenkaan vielä käyttöön, koska tietojen keruu järjestelmän tietokantaan on toteutusvaiheessa ja projekti on kehityksen alla.

Ohjelmiston hyötyinä ovat energiankulutuksen reaaliaikainen seuranta sekä mahdollisuus epäkuntoisen kohteen löytämiseksi tavallisuudesta poikkeavan kulutuksen vuoksi. Energiankulutus on nykypäivänä iso asia, joten sen seuraaminenkin on tärkeää.

LÄHTEET

1. Maja, Mikko. Keskustelut kokouksissa ja työn toteutuksen yhteydessä. Kesäkuu 2011.
2. Vuorinen, Asko. 2009. Energiankäyttäjän käsikirja. Ekoenergo Oy. ISBN 978-952-67057-2-9.
3. Hellgren, Heikkinen ja Suomalainen. 1996. Energia ja ympäristö. Hakapaino Oy. 2. painos. ISBN 951-719-674-1.
4. Kankaanpää, Timo. Ohjelmiston määrittely. Internet-sivu: http://www.cc.puv.fi/~tk/kurssit/Ohjelmiston_maarittely/luennot.htm [viitattu 3.9.2011]
5. McConnell, Steve. 2002. Ohjelmistotuotannon hallinta. Edita Prima Oy. ISBN: 951-826-194-6
6. Nielsen, Jakob. 2000. WWW-suunnittelu. Gummerus kirjapaino, Jyväskylä. ISBN 951-826-203-9.
7. Lehto, Tero. Tietokone 3/2005. Ohjelmistojen testaus. Saatavissa: http://www.tietokone.fi/lehti/tietokone_3_2005/ohjelmistojen_testaus_2534 [viitattu 3.9.2011]
8. Howard ja LeBlanc. 2004. Ohjelmoijan tietoturvaopas. Edita Prima Oy. ISBN: 051-826-663-8.
9. Microsoft Visual Studio. Internet-sivusto: <http://www.microsoft.com/visualstudio/en-us> [viitattu 2.8.2011]
10. Halvorson, Michael. 2008. Microsoft Visual Basic 2008. Gummerus Kirjapaino Oy. 1. painos. ISBN 978-952-220-021-1.
11. Microsoft SQL Server. Internet-sivusto: <http://www.microsoft.com/sqlserver/en/us/default.aspx> [viitattu 2.8.2011]
12. SAP Crystal Solutions. Internet-sivu: <http://www.sap.com/solutions/sap-crystal-solutions/query-reporting-analysis/index.epx> [viitattu 2.8.2011]
13. Peck, George. 2008. Crystal Reports 2008: The Complete Reference. ISBN 9780071590983.
14. A history of Internet Explorer. Internet-sivu: <http://windows.microsoft.com/en-US/internet-explorer/products/history> [viitattu 4.8.2011]
15. History of the Mozilla Project. Internet-sivu: <http://www.mozilla.org/about/history.html> [viitattu 4.8.2011]
16. Mozilla Firefox. Internet-sivusto: <http://www.mozilla.com/en-US/firefox/fx/> [viitattu 4.8.2011]

17. Inkinen, Ville. 2003. ASP.NET. Dark Oy. 1. painos. ISBN: 951-846-145-7.
18. Akarte, Chetankumar. 2008. .NET Framework Overview. Saatavissa:
<http://www.tipsntracks.com/6/dotnet-framework-overview.html> [viitattu 8.8.2011]
19. Franklin, Keith. 2002. VB.NET. Dark Oy. 1. painos. ISBN: 951-846-156-2.
20. W3Schools.com. Ajax Tutorial. Internet-sivusto:
<http://www.w3schools.com/Ajax/default.asp> [viitattu 8.8.2011]
21. Asleson, Ryan. Schutta, Nathaniel T. 2007. Ajax – Tehokas hallinta. Gummerus Kirjapaino Oy. ISBN 13:978-952-5655-08-7.
22. Retzlaff, Donald Alan, 2008. CSCE 3410 Introduction to PHP Programming. Internet-sivusto: http://www.cse.unt.edu/~donr/courses/3410/PHP/PHPnotes3.php#aPage_17 [viitattu 9.8.2011]
23. Kettunen, Eero. Päivitetty 5.2.2001. Johdatus T-SQL-ohjelmointiin. Lahden ammattikorkeakoulu. Liiketalouden laitos. Internet-sivu:
<http://www.lpt.fi/it/opetus/tietokannat/tsql.html> [viitattu 9.8.2011]