

KYMENLAAKSON AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma / Ohjelmistotekniikka

Jussi Tuominen

SOVELLUSKEHITYS MICROSOFT SHAREPOINT 2010 -ALUSTALLE

Opinnäytetyö 2013

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

TUOMINEN, JUSSI	Sovelluskehitys Microsoft SharePoint 2010 -alustalle
Opinnäytetyö	83 sivua + 33 liitesivua
Työn ohjaaja	Laboratorioinsinööri Marko Oras
Toimeksiantaja	Danisco Sweeteners Oy / Tomi Javanainen, SHE Manager
Marraskuu 2013	
Avainsanat	Microsoft, SharePoint, sovelluskehitys, ASP.NET

Microsoft SharePointin rooli on tulevaisuudessa yhä merkittävämpi, ja Microsoft onkin panostanut sen kehitykseen viime vuosina yhä enemmän. Se toimii yhä useamman erikokoisen organisaation keskitettynä tietovarastona siellä, missä sen on todettu olevan omimmillaan: intranet-ympäristössä. SharePointin käyttöönotto on tuonutkin lukuisille yrityksille lisäarvoa liiketoimintaan monilla eri osa-alueilla. SharePoint-ohjelmistoprojektit ovat usein aikaa vieviä, haastavia ja monimutkaisia, eikä ammattilaiseksi kehitytä hetkessä. Tämä johtuu kehitysympäristön laajuudesta.

Opinnäytetyön tarkoituksena oli tutustua Microsoft SharePoint 2010 -järjestelmän ominaisuuksiin ja sovelluskehitykseen. Työssä käsitellyt yleisiä asioita ovat SharePointiin olennaisesti liittyvän .NET-sovelluskehityksen kuvaaminen, itse Microsoft SharePoint 2010 -järjestelmän perusominaisuuksien selostaminen ja työn käytännön toteutuksena rakennetun sovelluksen kuvaus käyttöohjeineen. Sovelluksen kuvauksessa on käytetty myös teknisiä koodiesimerkkejä tuomaan vastapainoa sanallisesti selostetulle aineistolle. Työssä selostetaan projektin lähtökohdat, riskit ja sovelluksen elinkaaren rakentuminen.

Opinnäytetyössä suunniteltiin ja toteutettiin Kotkassa sijaitsevalle Danisco Sweeteners Oy:n tehtaalte Microsoft SharePoint 2010 -alustalla toimiva perehdytyssovellus, Danisco Orientation, jonka toimintaympäristö on yrityksen intranet. Perehdytykseen osallistuvat kaikki yrityksen yhteistyökumppanien työntekijät. Web-pohjainen Danisco Orientation -sovellus sisältää tietokannan perehdytyksen suorittaneista henkilöistä ja heidän työturvallisuuskorttitietonsa. Ylläpitäjät pystyvät hallitsemaan sovelluksen sisältämiä tietoja: käyttäjäkohtaista ja perehdytykseen liittyvää dataa. Sovellus toteutettiin pääosin ASP.NETin tarjoamilla tekniikoilla, mutta myös SharePointin tarjoamia ominaisuuksia hyödynnettiin. Sovellus saatiin valmiiksi sovitussa aikataulussa.

Sovelluksen rakentamisessa pyrittiin mahdollisimman loogiseen, toimivaan, tietoturvalliseen ja yrityksen tarpeet täyttävään ohjelmaan, jonka avulla yritys voi siirtyä aiemmin kirjallisesti toteutetusta perehdytyksestä sähköisesti toteutettavaan kokonaisuuteen. Myös SharePoint-sovelluksen asennusprosessin läpikäyminen kuului työhön. Ohjelmistokehittäjällä ei ollut aikaisempaa kokemusta SharePoint-kehitystyöstä, ja oman osaamisen edistäminen oli myös yksi olennainen tavoite tässä opinnäytetyössä.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Information Technology

TUOMINEN, JUSSI

Application Development for Microsoft SharePoint 2010

Bachelor's Thesis

83 pages + 33 pages of appendices

Supervisor

Marko Oras, Laboratory Engineer

Commissioned by

Danisco Sweeteners Oy / Tomi Javanainen, SHE Manager

November 2013

Keywords

Microsoft, SharePoint, application development, ASP.NET

Microsoft SharePoint's role as an information management solution will be even more important in the future. In the recent years Microsoft has invested significantly more than before in its development. During its history, SharePoint has added value to the business of companies of all sizes. Application development for SharePoint is generally challenging, time-consuming and complex because of its extent.

The objective of this study was to explore Microsoft SharePoint 2010's functionality and application development and to develop a practical application for it. In addition to SharePoint itself, this study focused on .NET software framework, which is one of the essential elements of the application development for SharePoint. The latter part of the study concentrated on the application, which was developed as a part of this thesis. It was mainly created with ASP.NET techniques, but some of SharePoint-provided techniques were also used. The web-based application, called Danisco Orientation was developed for Danisco Sweeteners Ltd to digitize the employee orientation process at the factory. The technical side with code examples was explained. The study discussed also the project's starting points, risks and the construction of software life cycle.

The purpose was to build a logical, user-friendly and secure application, which met the requirements of the commissioner. Going through the installation process was also a part of the project. The software developer did not have prior experience in developing applications for SharePoint. The improvement of personal programming skills was also an important objective of this thesis.

As a result of the project, a functional application containing the required features was completed. Operating in the company's intranet environment, Danisco Orientation application replaced the old orientation methods and overall simplified employees' orientation process in the company.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

LYHENTEET JA TERMIT

1	JOHDANTO	10
2	TOIMEKSIANTAJA	11
	2.1 Yritys	11
	2.2 Vaatimukset sovellukselle	12
3	TOTEUTUKSEN LÄHTÖKOHDAT	14
	3.1 Aikataulu	14
	3.2 Resurssit	15
	3.3 Riskit	15
4	.NET-SOVELLUSKEHYS	17
	4.1 Arkkitehtuuri	18
	4.2 Ohjelmointikielet	19
	4.2.1 C#	19
	4.2.2 Visual Basic .NET	20
	4.2.3 C++/CLI	20
	4.3 ASP.NET	20
	4.4 ADO.NET	22
5	MICROSOFT SHAREPOINT 2010	23
	5.1 Hierarkia	24
	5.2 Versiot	26
	5.2.1 SharePoint Foundation 2010	27
	5.2.2 SharePoint Server 2010 Standard	28
	5.2.3 SharePoint Server 2010 Enterprise	28
	5.3 Sovellustyypit	28
	5.4 Hallinta	29
	5.5 Sivut	31
	5.5.1 Perustyyllisivut	31
	5.5.2 Application Page -sivut	32
	5.5.3 Site Page -sivut	32

5.6	Web-osat	32
5.7	Piirteet	33
5.8	Tietolähteet	33
5.8.1	Listat ja SQL-taulut	34
5.8.2	Business Connectivity Services	35
5.9	Objektimallit	36
5.10	Tietoturva	38
5.10.1	Todennusmenetelmät	38
5.10.2	Käyttöoikeudet	39
6	DANISCO ORIENTATION -SOVELLUS	41
6.1	Käyttöliittymä	41
6.2	Rakenne	43
6.3	Ominaisuudet	45
6.3.1	Tietokanta	45
6.3.2	Tietojenhallintatyökalut	48
6.3.3	Piirteet	50
6.3.4	Monikielisyys	50
6.3.5	Tietoturva	54
6.3.6	Istuntojen käsittely	59
6.3.7	Käyttöoikeudet	60
6.3.8	Mukautetut komponentit	62
6.3.9	Erikoisominaisuudet	64
6.4	Testaus	66
6.5	Siirtokuntoon saattaminen	67
6.6	Käyttöönotto	69
7	SOVELLUKSEN TOTEUTTAMINEN	71
7.1	Kehitysympäristö	71
7.2	Kehitystyökalut	71
7.2.1	Microsoft Visual Studio 2010	72
7.2.2	Microsoft SQL Server 2008 R2	72
7.2.3	Microsoft SharePoint Designer 2010	73
7.3	Kehityksen elinkaari	74
8	YHTEENVETO JA POHDINTA	75
	LÄHTEET	78

LIITTEET

Liite 1. Danisco Orientation -käyttöohje (v1.0)

LYHENTEET JA TERMIT

Active Directory	Windows-ympäristön keskitetty hallinta tapahtuu Active Directoryn (AD) avulla. Se on tietokanta kaikista saman verkon tietokoneista ja käyttäjistä. Active Directory sisältää tietoa työasemien ohjelmistoista ja asetuksista sekä käyttäjäryhmistä ja käyttöoikeuksista.
AJAX	<i>Asynchronous JavaScript And XML</i> . Tekniikka, joka mahdollistaa sivun osittaisen päivittymisen asiakaspään ja palvelimen välillä jonkin tiedon muuttuessa.
API	<i>Application Programming Interface</i> . Ohjelmointirajapinta, jonka kautta eri ohjelmat voivat tehdä pyyntöjä ja olla vuorovaikutuksessa keskenään.
CAS	<i>Code Access Security</i> . Microsoftin .NET-kehikseen kehitetty tietoturvamekanismi. Sen avulla voidaan kontrolloida koodin ajamista sovelluksissa riippuen sen mukaan, miten luotettavaa se on.
CSS	<i>Cascading Style Sheets</i> . Tyylikieli www-dokumenttien ulkonäön määrittelyyn.
ECMAScript	JavaScriptin virallisesti standardoitu versio, joka kehitettiin aikoinaan estämään JavaScript-kielen hajaantuminen.
EULA	<i>End-User License Agreement</i> . Sopimus ohjelmiston valmistajan, julkaisijan ja/tai jakelijan sekä lisensoidun käyttäjän välillä. EULAssa selostetaan loppukäyttäjän oikeudet käyttää ohjelmistoa.
GAC	<i>Global Assembly Cache</i> . Keskitetty komponenttivarasto, josta jaetaan koodikirjastoja työasemassa sijaitseville ohjelmille ilman ristiriitojen syntymistä. GAC:ssa sijaitsevat komponentit ovat välimuistissa, joten ne ovat nopeasti sovellusten käytettävissä.
GUID	<i>Globally Unique Identifier</i> . Objektin yksilöllinen, 128-bittinen tunniste.

JSON	<i>JavaScript Object Notation</i> . Kevyt, JavaScript-luonteinen tiedonsiirtotapa.
jQuery	Nykyään ylivoimaisesti eniten web-sivuilla käytetty JavaScript-kirjasto. Se on ilmainen, helppokäyttöinen ja sillä on web-yhteisöissä laaja tuki. jQuery toimii yleisimmissä selaimissa.
HTML	<i>HyperText Markup Language</i> . Kuvauskieli, jota käytetään yleisesti verkkosivujen muotoiluun.
IDE	<i>Integrated Development Environment</i> . Sovelluskehitysympäristö, joka sisältää kaikki sovelluksen kehityksessä alusta loppuun saakka tarvittavat elementit, kuten lähdekoodimuokkaimen, kääntäjän ja virheenjäljittimen (debuggerin) yhden ohjelmiston sisällä.
IIS	<i>Internet Information Services</i> . Microsoftin kehittämä ja Windows-pohjaisissa palvelimissa ajettava palvelinohjelmistokokonaisuus.
LDAP	<i>Lightweight Directory Access Protocol</i> . Yleiskäyttöinen hakemistotietojen kyselyprotokolla.
NTLM	<i>NT LAN Manager</i> . Microsoftin kehittämä protokollasarja autentikointia ja istuntojen suojausta varten. Sitä käytetään Windows-ympäristössä työasemien ja palvelimien välillä.
RAD	<i>Rapid Application Development</i> . Nopean kehityksen malli. Ohjelmiston kehitysmenetelmä, jossa pyritään kehittämään ohjelmistoja hyvin pienessä ajassa tavallisesti joitakin sovelluksen ominaisuuksia karsien. Siinä hyödynnetään ennalta määritettyjä prototyyppitekniikoita ja työkaluja ohjelmistojen tuottamiseksi.
SAML	<i>Security Assertion Markup Language</i> . Laajennettava XML-standardi käyttäjien tunnistamis- ja valtuuttamistietojen jakamiseksi tietojärjestelmissä.

Silverlight	Microsoftin kehittämä teknologia niin kutsuttujen rikkaiden internet-sovelluksien kehittämiseen.
Staattinen metodi	Luokkametodi, johon viitataan luokan nimen perusteella. Siihen ei voida näin ollen viitata luokan instanssin eli esiintymän yhteydessä.
SQL	<i>Structured Query Language</i> . Relaatiotietokantojen standardoitu kyselykieli.
SQL-injektio	Hakkerointitekniikka, jossa hyökkääjä pystyy muuttamaan SQL-kyselyn rakennetta antamalla sopivan syötteen.
QueryString	Merkkijono, jonka avulla siirretään tietoa kahden verkkosivun välillä. Ne kulkevat parametreina verkko-osoitteessa, johon ne liitetään kysymysmerkillä. Mikäli QueryString-merkkijonoja on useampi, ne erotetaan toisistaan ”&”-merkillä.
XHTML	<i>EXtensible HyperText Markup Language</i> . XML-pohjainen, HTML:n korvaajaksi kehitetty merkintäkieli.
XML	<i>EXtensible Markup Language</i> . Rakenteisten dokumenttien merkintäkieli.
XSLT	<i>EXtensible Stylesheet Language Transformations</i> . Selaimessa tai palvelimella suoritettavan XML-muotoisen aineiston muuntamiseen käytetty tekniikka.
XSS	<i>Cross-Site Scripting</i> . Haavoittuvuuden muoto, jonka avulla web-sivustolla voidaan ajaa ulkoista selainpuolen koodia, käytännössä JavaScriptiä, joka voi aiheuttaa vakaviakin tietoturvauhkia.
WCF	<i>Windows Communication Foundation</i> . .NET-ympäristön teknologia palvelukeskeisten sovellusten kehittämiseen. Sen avulla voidaan yhdistää eri tekniikoita käyttävät palvelusovellukset toimimaan yhden tekniikan alaisuudessa.

1 JOHDANTO

Perehdyttäminen on investointi, jolla työntekijöiden osaaminen ja mielekkyys kasvavat. Lisäksi työn laatu voi parantua. Perehdytyksen kautta voidaan ehkäistä tehokkaasti työtapaturmia ja poissaoloja. Se on jatkuva prosessi, ja sitä kehitetään jatkuvasti yritysten omien tarpeiden mukaisesti. Järjestelmällisesti suunniteltuun perehdytykseen kuuluvat kaikki henkilöstöryhmät, myös ulkopuolisen työnantajan työntekijät. Perehdytyksestä säädetään työturvallisuuslaissa. (Fennia 2012.) Sen merkityksen kasvaessa myös erilaisten nykyaikaisten perehdytysmenetelmien tarve on kasvanut. Nykyihmiset ovat jo laajalti tottuneet suorittamaan erilaiset toimenpiteet sähköisesti. Tietotekniikan hyödyntäminen myös työntekijöiden perehdyttämisessä on näin ollen aiheellista.

Tässä opinnäytetyössä toteutettiin sähköinen perehdytysohjelma, Danisco Orientation, Kotkassa sijaitsevalle Danisco Sweeteners Oy:lle. Aihe valittiin, koska se oli mielenkiintoinen, nykyisellä sovelluskehityskentällä merkityksellinen ja tulevaisuutta ajatellen hyödyllinen kokonaisuus. Perehdytysohjelmaa käytetään tehtaan yhteistyökumppanien koulutukseen. Se sisältää käyttäjien perustiedot ja tiedot käyttäjien yrityksistä, yhteyshenkilöstä, perehdytysasuorituksista sekä työturvallisuuskorteista. Hyväksytystä perehdytyksen suorittamisesta käyttäjä saa tulostettavan todisteen. Suoritus on voimassa vuoden ajan. Ylläpitäjä voi hallita kokonaisvaltaisesti perehdytyssovelluksen sisältämää dataa ja tarkastella käyttäjätietojen kehittymistä erillisellä hallintasivulla. Danisco Orientation -sovelluksen toimintaympäristö on Danisco Sweeteners Oy:n intranet. Yritykselle laadittiin sovelluksen käytöstä erillinen käyttöohje.

Danisco Sweeteners Oy:n Kotkan tehtaan työntekijät noudattavat kaikille yhteistä työturvallisuusohjeistusta, jonka sisältöä perehdytyksessä läpikäydään. Perehdytyksen tarkoituksena on turvata työ- ja tuoteturvallisuus sekä sujuva työnteko. Yleiskoulutus järjestetään kaikille yhteistyökumppaneiden työntekijöille ja yksityiskohtaisempi perehdytys suoritetaan työkohtaisesti. (Danisco Sweeteners Oy, Kotkan tehdas 2012, 9.) Perehdytyskoulutus on ollut tehtaan käytäntönä aiempinakin vuosina. Se on suoritettu aiemmin työntekijöille näytetyn PowerPoint-esityksen pohjalta kirjallisesti ja tulokset on kirjattu manuaalisesti ylläpidettäviin Excel-taulukoihin.

Opinnäytetyön tavoitteena oli tutustua Microsoft SharePoint 2010 -ympäristöön ja rakentaa toimiva sovellus tälle kyseiselle alustalle. Työ kattoi kaiken alusta alkaen: se

alkoi kehitysympäristön rakentamisesta ja ohjelmistoihin perehtymisestä ja jatkui sovelluskehityksen merkeissä siihen asti, kunnes lopputulos oli valmis toimeksiantajan omiin järjestelmiin asennettavaksi. Toteutuksessa pyrittiin omia resursseja myötäillen ottamaan huomioon toimeksiantajan vaatimukset ja toiveet.

Opinnäytetyö toteutettiin sen laajuuden vuoksi parityöskentelynä pääasiallisesti Kymenlaakson ammattikorkeakoulun tiloissa. Sovelluksen kehitystyöhön kanssani osallistui opiskelija Sami Saranpää, joka oli jo aiemmin kehittänyt sovellusta omana projektityönään. Sitä lähdettiin alun alkaen rakentamaan syksyllä 2012 PHP- ja MySQL-pohjaiseksi, mutta vuoden 2013 alussa esiin tulleet muutokset siirsivät kehityssuunnan aivan toisaalle. Projekti kaipasi lisäapua ja ilmoittauduin mukaan, sillä halusin oppia SharePointin peruskäytön ja vahvistaa omia ohjelmointitaitojani Microsoftin .NET-tekniologioita käyttäen. Koska uusi alusta oli molemmille ennalta tuntematon, työn asettamat haasteet kasvoivat merkittävästi. Vaikka itse sovelluksen perustoiminnallisuus säilyi ennallaan, projektin voitiin katsoa alkaneen kokonaan alusta.

2 TOIMEKSIANTAJA

Perehdytyssovelluksen toimeksiantajayritys oli Danisco Sweeteners Oy:n Kotkan tehdas. Yrityksen puolelta projektin vastuuhenkilönä toimi SHE Manager Tomi Javanainen.

2.1 Yritys

Danisco Sweeteners Oy:n Kotkan tehdas on osa yhdysvaltalaisista DuPont-konsernia. DuPont perustettiin vuonna 1802. Se on kansainvälisesti arvostettu yritys, jonka liiketoiminta-alueisiin sisältyy useita laajoja kokonaisuuksia. DuPontin yritystoiminta keskittyy ympäristön tutkimus- ja kehitystyöhön, ja sen pääasialliset tavoitteet ovat kestävän kehityksen ylläpitäminen, ympäristön suojeleminen sekä terveellisen ja turvallisen elämän edesauttaminen ihmisille tieteen keinoin. (DuPont 2013.)

DuPontin ydinarvoja ovat turvallisuus, terveys, ympäristöstä huolehtiminen, hyvä eettinen toiminta ja ihmisten kunnioittaminen. (DuPont 2011, 3.) Yritys on kehittänyt yli 200-vuotisen taipaleensa aikana useita innovaatioita maailmanmarkkinoille. Esimerk-

kejä DuPontin kehittämistä uraauurtavista tuotteista ovat nylon eli synteettinen kuitu ja neopreeni eli synteettinen kumi. Kuva 1 näyttää yrityksen logon.



Kuva 1. DuPontin logo

DuPont osti toukokuussa 2011 tanskalaisen elintarvikeyhtiö Daniscon, joka on maailman suurin elintarvikkeiden ainesosien, kuten makeutusaineiden, entsyymien ja biopohjaisten aineiden valmistaja. Se on toistaiseksi DuPontin suurin kauppa 2000-luvulla. (Bloomberg 2011.) Kansainvälisesti Danisco työllistää lähes 7 000 henkilöä yhteensä 23 maassa (Seppälä 2011). Kuva 2 näyttää Daniscon logon.



Kuva 2. Daniscon logo

Danisco Sweeteners Oy:n Kotkan tehtaalla, joka toimii osana DuPont Nutrition & Health -liiketoimintayksikköä, valmistetaan ksylitolia ja luonnonsokeria eli fruktoosia elintarvike- ja lääketeollisuuskäyttöön. Yli 95 % tuotteista menee vientiin, joka kohdistuu yhteensä noin 50 maahan. Danisco Sweeteners Oy:n Kotkan tehtaan liikevaihto oli 86 miljoonaa euroa vuosina 2009–2010. Tehdas työllistää 115 ihmistä. (Danisco Sweeteners Oy, Kotkan tehdas 2012, 10.)

2.2 Vaatimukset sovellukselle

Danisco Sweeteners Oy:n puolelta esitettiin sovellukselle lukuisia vaatimuksia, joiden mukaisesti sitä lähdettiin kehittämään. Ensimmäisessä varsinaisessa palaverissa oli opinnäytetyön tekijöiden ja Tomi Javanaisen lisäksi läsnä DuPont IT:n edustaja Seppo Kohonen, jonka kanssa käytiin läpi sovelluksen ominaisuuksia. Seuraavassa on listattuna oleelliset tekniset vaatimukset, jotka sovelluksen täytyy täyttää:

- Sovelluksen käyttöympäristönä on Windows Server 2008 R2 (x64).
- Sovelluksen täytyy toimia työasemassa User-oikeuksilla.
- Sovelluksen toteutuksessa käytettävänä ohjelmointikielenä olisi joko Visual Basic tai C# (C Sharp). Näistä kahdesta valittiin C#, koska ohjelmistokehittäjillä oli siitä enemmän kokemusta.
- Sovellusta käytettäisiin Internet-selaimella (yksinomaan Internet Explorer 8- ja 9-versioilla sekä tulevaisuudessa 10-versiolla).
- Tietokantaympäristönä on käytettävä Microsoft SQL Server Express 2008 R2:ta tai Microsoft Access Runtime 2010:a. Näistä kahdesta valittiin Microsoft SQL Server Express 2008 R2, koska ohjelmistokehittäjillä oli siitä enemmän kokemusta.
- Sovelluksesta tulee tuottaa käyttöohje ja sen lähdekoodi on dokumentoitava niin tarkasti, että ohjelmaa voidaan mahdollisuuksien mukaan kehittää jatkossa yrityksen omilla resursseilla. Käyttöohje on tämän dokumentin liitteenä (Liite 1).

Seuraavassa on listattu itse sovelluksen toimintavaatimukset, jotka tosin tarkentuivat vielä ohjelman kehityksen aikana:

- Sovelluksessa on rekisteri perehdytyksen suorittaneista henkilöistä.
- Perehdytysmoduuleja on kaksi kappaletta (urakoitsijat/vierailijat).
- Sovellukseen on jälkeempään mahdollista lisätä enemmänkin perehdytysmoduuleja.
- Tietokantaan on mahdollisuus syöttää perehdytysaineistoja, joita tulee pystyä myös muokkaamaan jälkeempään.
- Kullakin sovellukseen lisätyllä aineistolla on omat kysymyksensä, joita tulee pystyä sovelluksesta hallitsemaan.
- Ohjelman rekisterissä on seurantamahdollisuus työntekijöiden työturvallisuuskorttien voimassaoloista.
- Ohjelman käyttöliittymän kielinä ovat sekä suomi että englanti.
- Ohjelmassa on erillinen ylläpito-osio, mistä löytyy työkalut tietojen tarkasteluun ja hallintaan.
- Ohjelmasta tulee pystyä tulostamaan raportteja ja erikseen määritettyjen ulkoasuvaatimusten mukaisia todistuksia perehdytyksen suorittaneille työntekijöil-

le. Ne ovat käyntikorttikokoisia dokumentteja, joita työntekijät itsellään säilyttävät ja tarvittaessa esittävät, mikäli näin pyydetään.

3 TOTEUTUKSEN LÄHTÖKOHDAT

Koska opinnäytetyön aihepiiri on laaja-alainen, oli tärkeää kartoittaa tarkasti lähtökohdat, joista sovellusta lähdettiin kehittämään. Kaikista suotuisimmaksi toteutustavaksi valikoitui suuren kokonaisuuden jakaminen pienempiin osakokonaisuuksiin. Niistä kummallekin ohjelmistokehittäjälle valittiin sopivat osa-alueet, joihin kumpikin omilla tahoillaan keskittyivät. Kaiken kaikkiaan jo varsinaisen ohjelmoinnin aloitus antoi odottaa itseään, sillä oli oltava varma resurssien saatavuudesta ja yleensä oikealla tiellä olemisesta.

3.1 Aikataulu

Opinnäytetyön voidaan katsoa virallisesti alkaneen huhtikuusta 2013 eteenpäin, vaikka suunnittelupalavereja oli pidetty jo alkuvuodesta 2013. Ensimmäiset kolme kuukautta sisälsivät tavallisen koulutyön ohella pääasiassa aiheeseen perehtymistä ja kehitysympäristön pystyttämistä siihen kuntoon, että varsinainen kehitystyö voitiin aloittaa ilman lisätoimenpiteitä. Työparini Sami Saranpää aloitteli projektin työstämistä täysipäiväisesti toukokuusta 2013 lähtien. Minä aloitin sovelluksen kokopäiväisen työstämisen kesäkuun puolivälin jälkeen saatvani harjoittelujakson päätökseen. Kesällä 2013 oli tarkoitus työstää sovellusta niin pitkälle kuin mahdollista.

Sovelluksen valmistumiselle asetettiin takarajaksi 31.12.2013. Koska aikaa oli käytössä melko runsaasti, sovellusta pystyttiin työstämään rauhassa myös pienempiin yksityiskohtiin paneutuen. Tämä oli positiivinen asia ottaen huomioon aiheen laajuuden ja ohjelmistokehittäjien kokemattomuuden. Ajan runsas määrä mahdollisti oman tietotaidon aktiivisen kehittämisen ja suuren työmäärän jakamisen tasaisesti käytetylle aikavälille. Varsinaisia välietappeja ei matkan varrelle ollut määritelty, mutta toimeksiantajan kanssa aika ajoin pidetyissä tapaamisissa keskityttiin loppua kohti yhä enemmän sovelluksen käytännön toimintaan.

3.2 Resurssit

Käytettävissä oleva tietokonelaitteisto oli Kymenlaakson ammattikorkeakoulun tarjoamaa. Sovellusta kehitettiin ammattikorkeakoululta opiskelukäyttöön saaduilla kannettavilla tietokoneilla. Näin ollen laitteiden käytöstä ei koitunut ohjelmistoprojektille kustannuksia. Myös projektissa käytetyt ohjelmistot olivat ilmaisia, lukuun ottamatta Visual Studio 2010 -ohjelmistoa. Kyseisestä ohjelmistosta käytettiin näet Professional-versiota. Ohjelmistokehittäjät saivat sen käyttöönsä kuitenkin ilman kustannuksia Microsoftin oman DreamSpark-palvelun välityksellä. Tämä sähköinen palvelu tarjoaa korkeakouluopiskelijoille huomattavan määrän työkaluja ohjelmistojen suunnitteluun ja kehittämiseen.

Sovelluksen kehityksen tukena oli kahden projektiin osallistuneen ohjelmistokehittäjän osaaminen ja tietotaito. Työssä tarvittava informaatio hankittiin itse ja se sovellettiin kehittyvään tuotteeseen ohjelmistokehittäjien yhdessä sopimalla tavalla. Sovelluksen kehityksessä ei siis ollut mukana ulkopuolisia asiantuntijaresursseja, joten tästäkään projektille ei aiheutunut lisäkustannuksia.

3.3 Riskit

Ohjelmistoprojekti piti sisällään heti alusta lähtien erilaisia riskitekijöitä, jotka oli hyvä tiedostaa saman tien. Siinä kiinnitettiin huomiota ennen kaikkea riskeihin, jotka esittää taulukko 1.

Taulukko 1. Ohjelmistoprojektin riskitekijät ja ennalta ehkäisevät toimet

Riski	Ennalta ehkäisevät toimet
<p>Sovelluksen toteutusryhmä on pienikokoinen: mahdolliset työkyvyttömyyttä aiheuttavat tilanteet voivat vähentää työpanosta ja tehokkuutta.</p>	<ul style="list-style-type: none"> Vastuualueiden tarkka rajaaminen, jotta kumpikin on tietoinen toisen työstämisestä osa-alueista. Näin ollen toinen kykenee paremmin ja nopeammin paikkailemaan toisen työstämää koodisisältöä.
<p>Ohjelmistokehittäjien välinen yhteydenpito takkuilee ja tuotantotiimin yhtenäisyys rakoilee.</p>	<ul style="list-style-type: none"> Selkeiden yhteisten sääntöjen määrittäminen, yhteinen hyväksyminen ja noudattaminen. Epäselvien tilanteiden selvittäminen ensi tilassa - ilman omin päin päätettyjen ratkaisujen toteutusta tai toisaalta epärointiin kulutettavaa aikaa.
<p>Projektin vaatimusten epäselvyys: asiakkaan vaatimukset esitetty epäselvästi tai harhaanjohtavasti. Asiakas ei tiedä riittävän tarkasti, mitä haluaa.</p>	<ul style="list-style-type: none"> Intensiivinen kommunikointi toimeksiantajan suuntaan ja kaikkien mahdollisten epäselvien asioiden esille nostaminen heti ensi tilassa. Toimeksiantajan tietoisuuden parantaminen kehitteillä olevan tuotteen toiminnasta ja ominaisuuksista. Esimerkkejä, kuvankaappauksia... mitä vain tuotteen liittyvää materiaalia.
<p>Aikataulun pettäminen: projektille asetettu aika-arvio on arvioitu harhaan ja voi aiheuttaa aikataulujen pettämistä ja projektin venymistä.</p>	<ul style="list-style-type: none"> Aikarajojen asettaminen: ylitsepääsemättömältä tuntuvien, luvattoman paljon aikaa vievien toimintojen vaihtoehtoinen toteutustapa. Riittävän pienien välitavoitteiden asettaminen, esimerkiksi viikon aikana jokin ominaisuus toteutettu ja testattu. Omien resurssien tiedostaminen ja lopputuotteen ominaisuuksista karsiminen, mikäli välttämätöntä.

<p>Odottamattomat ympäristöongelmat: laite- tai verkko-ongelmat.</p>	<ul style="list-style-type: none"> • Varmuuskopiointi projektin säilymisen turvaamiseksi ja mahdollisimman nopean palauttamisen mahdollistamiseksi. • Vaihtoehtoisen, kehitettävän tuotteen kanssa yhteensopivan kehitysympäristön pystyttäminen.
<p>Asiakas ei ole tyytyväinen lopputuotteen.</p>	<ul style="list-style-type: none"> • Asiakkaan asettamien vaatimusten huolellinen läpikäyminen ja vahvistaminen asiakkaalta vielä tarvittaessa useaan kertaan projektin edetessä.
<p>Testaus ei löydä kaikkia sovelluksesta löytyviä bugeja tai epäloogisuuksia.</p>	<ul style="list-style-type: none"> • Sovelluksen kokonaisvaltainen testaaminen ja testauttaminen erilaisilla koehenkilöillä ennen varsinaiseen tuotantoon siirtämistä.
<p>Sovelluksen siirtäminen tuotantoympäristöön epäonnistuu tai se ei toimi tuotantoympäristössä.</p>	<ul style="list-style-type: none"> • Perusteellinen ennakkoselvittely sovelluksen siirtämisen vaatimuksista ja prosessin kulusta. • Sovelluksen kehittäminen tuotantoympäristöä silmällä pitäen ja samalla yhteensopivuusongelmien minimoiminen.

4 .NET-SOVELLUSKEHYS

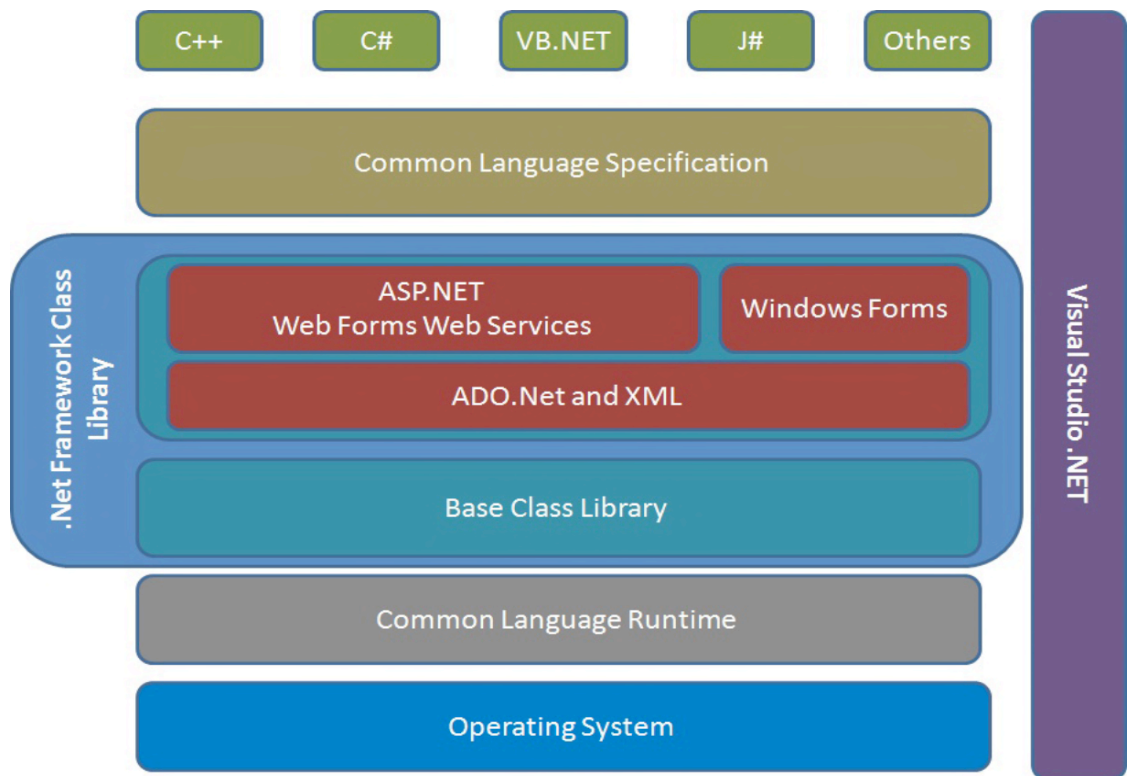
.NET-sovelluskehys (.NET Framework) on Microsoftin kehittämä ohjelmistokehitysympäristö, joka on kehitetty etupäässä Windows-käyttöjärjestelmille. .NET-kehys tarjoaa työkaluja ja laajan komponenttikirjaston erityyppisten sovellusten, kuten työpöytä-, verkko-, mobiili- ja web-palvelusovellusten, kehittämiseen ja suorittamiseen. (Microsoft MSDN 2013b.) Sen ensimmäinen versio 1.0 julkaistiin vuonna 2002.

.NET-kehys on ladattavissa ilmaiseksi Microsoftin verkkosivuilta, mutta uudempiin käyttöjärjestelmiin se tulee valmiiksi asennettuna. Se tarvitaan, jotta työasemassa pysyttäisiin ajamaan .NET-pohjaista koodia.

SharePoint on rakennettu .NET-sovelluskehityksen päälle. Näin ollen sen sisältämät teknologiat ovat myös SharePoint-sovelluskehityksen käytettävissä. Koska .NET-sovelluskehityksen ja ennen kaikkea ASP.NETin tarjoamat työkalut ovat oleellinen osa SharePoint-sovelluskehitystä, on perusteltua avata hieman tarkemmin tämän ympäristön rakennetta ja sisältöä. SharePoint 2010 -version ollessa kyseessä vaatimuksena on .NET-sovelluskehityksen versio 3.5.

4.1 Arkkitehtuuri

.NET-sovelluskehityksen tavoitteena on tarjota johdonmukainen, yhtenäinen, tehokas ja oliolähtöinen ympäristö erityyppisten sovellusten kehitykseen joko paikallisesti tai etänä. Myös ohjelmointikielten yhteentoimivuus on olennainen osa-alue. .NET-sovelluskehityksessä ohjelmointikielillä on muun muassa yhteinen ohjelmointimalli (*Common Programming Model, CPM*) ja luokkakirjasto. Tämä tekee .NET-kehityksestä niin kutsutun kielineutraalin ympäristön, missä kaikki järjestelmän tukevat ohjelmointikieliset ovat tasavertaisia. (Alikoski 2006.) Kuva 3 esittää .NET-kehityksen arkkitehtuurin.



Kuva 3. .NET-kehityksen arkkitehtuuri (ASP.NET Tips for Projects 2013.)

Kuten kuva 3 esittää, .NET-kehiksen arkkitehtuuri on hierarkkinen ja se voidaan esittää tasoina. Alimmaisena tasona on käyttöjärjestelmä (*Operating System*). .NET-kehys sijaitsee näet käyttöjärjestelmän ja sovellusten välissä. Toisessa tasossa sijaitsee Common Language Runtime (CLR). Kolmas taso on Base Class Library (BCL), joka sisältää kaikki .NET-ympäristössä käytettävät perusluokat ja -teknologiat, jotka ovat sovelluskehityksen hyödynnettävissä. Näitä teknologioita ovat esimerkiksi Windows Presentation Foundation (WPF), Windows Forms, ASP.NET ja Windows Communication Foundation (WCF). (Microsoft 2010.) Tämän jälkeen, ennen varsinaisia, aiemmista tasoista riippuvaisia .NET-sovelluksia, tulee Common Language Specification (CLS), joka määrittelee koodin yhteentoimivuuden. Tämä on välttämätöntä, jotta se pystyisi saumattomasti toimimaan muiden ympäristön kielten kanssa.

Common Language Runtime (CLR) on olennaisin osa .NET-kehystä. Se tarjoaa virtuaalikoneeseen verrattavan ajonaikaisen ympäristön, joka huolehtii koodin suorittamisesta ja tarjoaa palveluita kehityksen, käyttöönnoton ja testauksen parantamiseksi. Sen tärkeimpiin tehtäviin kuuluvat suoritettavan koodin muistinhallinta, säikeistys, virheiden käsittely ja turvallisuuden hallinta esimerkiksi CAS-mallin (*Code Access Security*) avulla. Koodi, joka on osoitettu Common Language Runtimeille ja jota ajetaan sen alaisuudessa, on niin sanottua hallittua koodia (*Managed Code*). (Sapossnek 2005.)

4.2 Ohjelmointikielät

.NET-kehys tukee yli 40:tä eri ohjelmointikieltä (Rajaram 2006, 103). Joukkoon mahtuu sekä Microsoftin tarjoamia että kolmannen osapuolen ohjelmointikieliä. Seuraavassa on lyhyesti käyty läpi niistä tunnetuimpia eli Microsoftin kehittämiä ohjelmointikieliä.

4.2.1 C#

C# eli C Sharp on Microsoftin kehittämä, yksinkertainen, moderni ja oliopohjainen ohjelmointikieli .NET-ympäristöön. Se on kehitetty ohjelmakoodin laadukkuutta ja ohjelmistokehittäjän tuottavuutta parantavaksi ohjelmointikieleksi. C# perustuu C++-kieleen ja on syntaksiltaan Javan kaltaista. Se on käyttökelpoinen sekä laajoissa että sulautetuissa järjestelmissä. (Whitehead 2008.) Vuonna 2000 julkaistu C# on kasvatanut merkittävästi suosiotaan Windows-ympäristön sovelluskehityksessä.

4.2.2 Visual Basic .NET

Visual Basic .NET eli VB.NET on oliopohjainen ohjelmointikieli, joka on kehitetty vanhan Visual Basic -kielen pohjalta osaksi .NET-kehystä. Suurimmat erot VB.NETin ja C#:n välillä ovat syntaksissa ja Visual Studio IDE:ssä sovelluksia kehitettäessä. VB.NET tukee nopean kehityksen mallia (*Rapid Application Development, RAD*) niissä sovelluksissa, joissa on graafinen käyttöliittymä. (Nasr 2004.)

4.2.3 C++/CLI

C++/CLI eli *Managed C++* sitoo C++-kielen Microsoftin .NET-kehityksen piiriin Common Language Runtimeella ajettavaksi. CLI- (Common Language Infrastructure) arkkitehtuuri tukee dynaamista ja kieliriippumatonta ohjelmointimenetelmää, joka perustuu virtuaaliseen suoritusympäristöön (*Virtual Execution System, VES*). .NET-ympäristössä tämä virtuaalikoneeseen verrattava järjestelmä on tässä raportissa aiemmin mainittu Common Language Runtime (CLR). C++/CLI on kehitetty olemaan alimman tason ohjelmointikieli .NET-ympäristössä. .NET-ympäristö tukee kuitenkin myös natiivin C++-kielen käyttöä .NET-pohjaisissa sovelluksissa, joten kokonaan C++/CLI:n ei tarvitse C++:aa korvata. (Sivakumar 2007, 4 - 5.)

4.3 ASP.NET

ASP.NET eli Active Server Pages on Microsoftin kehittämä kehys dynaamisten web-sivustojen ja web-palveluiden rakentamiseksi .NET-kehityksen tukemien ohjelmointikielten keinoin. ASP.NET on palvelinpuolen teknologia ja sisältää joukon luokkia, jotka on suunnattu erityisesti web-sivustojen rakentamiseen. ASP.NET-kehystä ajetaan web-palvelimella, tavallisimmin IIS:llä, johon on asennettu .NET-sovelluskehys. ASP.NET-sivu muunnetaan palvelimella selaimen ymmärtämään muotoon. ASP.NET-teknologiaa käyttävät www-sivut tunnistaa tiedostopäätteestä .aspx. (Wright 2010, 3, 5.)

ASP.NETistä on olemassa kolme ohjelmointimallia: ASP.NET Web Forms, ASP.NET Web Pages ja ASP.NET MVC, jotka kaikki perustuvat ASP.NET-teknologiaan. ASP.NET Web Forms -malli on vanhin ja perinteisin malli. Tässä mallissa luodut www-sivut voivat sisältää perinteisen HTML-koodin seassa palvelinpuo-

len kontrolleja, joiden toimintaa ohjataan tapahtumankäsittelijöiden avulla. ASP.NET Web Forms -malli sisältää suuren joukon valmiita kontrolleja. SharePoint-sovelluskehityksessä käytetään tätä mallia ja tämän myötä myös tämän opinnäytetyön ohjelmistoprojektissa. ASP.NET MVC on uudempi menetelmä, joka perustuu niin kutsuttuun Model-View-Controller-mekanismiin. Siinä sovelluksen toimintalogiikka erotetaan käyttöliittymästä. Tämä kehys soveltuu parhaiten testivetoiseen ohjelmistokehitykseen. ASP.NET Web Pages -malli on suunnattu yksittäisille sivuille, eikä siinä sivuilla ole niin kutsuttua elinjaksoa kuten ASP.NET Web Forms -mallissa. Ohjelmakoodi kirjoitetaan ASP.NET-sivulle avoimesti C#:n tai VB.NETin lisäksi niin kutsutulla Razor-syntaksilla kuten ASP.NET MVC -mallissakin. (Bandarupalli 2012.) Useiden mallien käyttö samassa web-sovelluksessa on mahdollista. (Wright 2010, 5.)

ASP.NET-www-sivut voidaan toteuttaa kahdella tavalla. Joko niin, että sekä toiminnallisuus että kontrollit (eli lomake-elementit ja niin edelleen) sijoitetaan kaikki samaan ASPX-tiedostoon tai siten, että kirjoitetaan ohjelmakoodi (esimerkiksi C# tai VB.NET) erilliseen osioon (*Code-Behind*) sovelluksen toiminnallisuuden toteuttamiseksi. Näin on esimerkiksi menetelty tässä opinnäytetyössä, koska SharePoint-ympäristössä ASP.NET-sivuille avoimesti kirjoitettu ohjelmakoodi ei ole sallittua. ASP.NET-sivut noudattavat omanlaista syntaksia, joka sisältää direktiivejä, palvelinkontrollien määrittämiä, ohjelmakoodilohkoja, datan sitomislausekkeita ja niin edelleen. Myös tavallista HTML-kieltä on mahdollista käyttää, ja palvelimella suoritettaviksi kontrollit voidaan asettaa lisäämällä niihin määrittäminen `runat="server"`. Taulukko 2 esittää esimerkkejä koodista, joita ASP.NET-sivuilla voi esiintyä.

Taulukko 2. ASP.NET-sivun koodiesimerkkejä

Merkitys	Syntaksi
Direktiivi	<code><%@ Page Language="C#" %></code>
Palvelinkontrolli	<code><asp:Button runat="server" /></code>
Koodilohko	<code><script runat="server">...</script></code>
Datan sitomislauseke	<code><%# %></code>
Palvelinpuolen kommentti	<code><%-- --%></code>
Ohjelmakoodilohko	<code><%= %></code> ja <code><% %></code>

ASP.NETissä *www*-sivuille on määritelty tietty elinjakso (*Page Life Cycle*). Tiettyä sivun tapahtumaa kuvastaa oma metodinsa: esimerkiksi **Page_Load**-metodiin sisällytetään tapahtumat, joita tapahtuu sivun latautuessa. Sivulla eri kontrollien tapahtumille on olemassa omat tapahtumankäsittelijänsä. Esimerkiksi valintalistan valinnan vaihtuessa tai tiettyä painiketta painettaessa voidaan määritellä tiettyyn metodiin tapahtuvaksi jotakin. Kontrollien määrittelyyn tulee viittaus tapahtumankäsittelijämetodiin, joka on esimerkiksi painikkeilla **OnClick** ja valintalistalla **OnSelectedIndexChanged**. Yksi tärkeä ominaisuus ASP.NET-kehityksessä, joka oli käytössä myös tässä työssä, on nimeltään *PostBack*. Se tarkoittaa tapahtumaa, jolloin ASP.NET-sivu lähetetään palvelimelle prosessoitavaksi.

ASP.NET-web-sovelluksen määrittelyä varten sille luodaan XML-muotoinen *web.config*-tiedosto. Se sisältää muun muassa sovelluksen tietoturvaan ja tilan- ja muistinhallintaan liittyviä määrittelyjä. Erityisessä roolissa on myös ASP.NET-web-sovelluksen hakemistorakenne, missä kukin kansio varastoi ennalta määrättyjä tiedostoja. Esimerkiksi *bin*-hakemisto sisältää sovelluksessa käytetyt komponentit (.dll) ja *App_GlobalResources*-hakemisto resurssitiedostot sovelluksen lokalisoimista varten. Sekä määrittelytiedosto että ASP.NET-web-sovelluksen kansiorakenne olivat olennaisia asioita tämän opinnäytetyön sovelluskehityksessä.

4.4 ADO.NET

ADO.NET on osaksi .NET-kehystä rakennettu teknologia, joka tarjoaa työkalut erilaisiin datalähteisiin kytkeytymiseksi. Eri datalähteissä sijaitsevaa data-aineistoa voidaan hallita ADO.NETin avulla monin tavoin. Sen sisältämien ”tiedonvälittäjien” (*Data Providers*) tarjoamien metodien kautta nämä toimenpiteet tapahtuvat sekä yhteydellisissä että yhteydettömissä ympäristöissä. Yhteys muodostetaan erityisen merkkijonon avulla, joka on nimeltään **ConnectionString**. Se on datalähdekohtainen ja sisältää tyypillisesti ainakin autentikointitiedot ja tietokantapalvelimen sijaintitiedot. (MacDonald, Hamilton 2003, 3, 16.)

ADO.NETin tiedonvälittäjät muodostavat yhteyden datalähteisiin, kuten Microsoft SQL Serveriin, Oracleen tai OLE DB -palveluun, ja tarjoavat luokkia erilaisten kysyjen suorittamiseksi niihin. Eri tiedonvälittäjille tarkoitettut luokat periytyvät yhteisestä

luokkakirjastosta ja toteuttavat yhdenmukaisen joukon rajapintoja, jolla toiminnallisuus toteutetaan datalähteestä riippumatta. (MacDonald, Hamilton 2003, 3 - 4.)

ADO.NET-tekniikan yhteydellinen tila tarjoaa yhdensuuntaisen ja vain luku-muotoisen yhteyden datalähteeseen komentojen suorittamiseksi siihen. Yhteydellisen ympäristön luokkia ovat **Connection**, **Command**, **DataReader**, **Transaction**, **ParameterCollection** ja **Parameter**. Yhteydetön ympäristö mahdollistaa datan tuomisen lähteestä, sen muokkaamisen ja palauttamisen takaisin. Sen sisältämät luokat ovat datalähteestä riippumattomia. Ne ovat seuraavat: **DataSet**, **DataTable**, **DataColumn**, **DataRow**, **Constraint**, **DataRelation** ja **DataGridView**. **DataAdapter**-luokka eroaa muista siinä, että se siltaa datalähteen ja yhteydettömän ympäristön luokat yhteydellisen ympäristön luokkien tavoin. (MacDonald, Hamilton 2003, 4 - 5.)

5 MICROSOFT SHAREPOINT 2010

Microsoft SharePoint on kokonaisvaltainen, turvallinen ja tarpeisiin mukautuva web-pohjainen tiedonhallintaratkaisu kaikenkokoisille yrityksille ja yhteisöille. Se muodostaa lukuisine ominaisuuksineen yhteistyöympäristön, joka mahdollistaa henkilöiden nopean pääsyn järjestelmän sisältöön, kuten dokumentteihin ja muuhun tietoaan, yhdestä keskitetystä sijainnista. Tämän myötä se voi nostaa työskentelyn tehokkuutta ja tuottavuutta sekä parantaa henkilöstön viestintää työyhteisössä merkittävästi. SharePoint tukee toiminnassaan Microsoftin kehittämiä teknologioita, esimerkiksi Microsoft Office -tuoteperheen sovelluksia. Sen voi toimia alustana niin intranet- ja extranet-ratkaisuille kuin julkisille www-sivustoillekin. (Moonsoft 2013.)

Tässä opinnäytetyössä tarkastellaan yksinomaan Microsoft SharePoint 2010 -järjestelmän ominaisuuksia eikä oteta kantaa sen seuraajan, Microsoft SharePoint 2013:n, sisältämiin uusiin ominaisuuksiin. Tässä luvussa kerrotaan SharePoint-järjestelmän perusrakenteesta, sen eri versioista sekä perusominaisuuksista, joista jokaisen SharePoint-sovelluskehittäjän on hyvä olla tietoinen. Kaikkia järjestelmän ominaisuuksia on mahdoton käydä tässä raportissa läpi, ja siksi seuraaviin lukuihin onkin valikoitu ainoastaan tämän projektin kannalta olennaisimmat kokonaisuudet.

Huomioitavana seikkana mainittakoon, että tässä opinnäytetyössä ei perehdytä lainkaan SharePoint 2010 -järjestelmän asennusprosessiin ja kehitysympäristön pystytykseen. Siihen on sitä vastoin paneuduttu työparinani tässä projektissa olleen Sami Saranpään opinnäytetyössä ”*Microsoft SharePoint 2010 -kehitysympäristön pystyttämisen*”.

5.1 Hierarkia

SharePointin hierarkian rakentuminen kannattaa käydä läpi mahdollisimman aikaisessa vaiheessa, kun aloittaa työskentelyn SharePointin kanssa. Hierarkia on monitasoinen ja se muotoutuu hyvin pitkälle käyttäjien tarpeiden mukaisesti. Myös ohjelmoinnin kannalta hierarkian eri osien tiedostaminen on tärkeää, sillä SharePointin sisältämä palvelinobjektimalli, josta kerrotaan tarkemmin luvussa 5.9, perustuu hierarkian rakenteeseen. Tässä luvussa esitellään lyhyesti hierarkian tärkeimmät elementit.

Farmi tai SharePoint-farmi (*Farm*) on yhden tai useamman fyysisen palvelimen joukko, joka muodostaa yhteisesti alustan SharePointille. Farmi sisältää tavallisimmin ainakin www- ja tietokantapalvelimen. Www-palvelimen ohjelmistona on tavallisimmin Internet Information Services (IIS) ja tietokantapalvelimen Microsoft SQL Server, joiden versiot riippuvat SharePointin versiosta. Lisäksi sovelluspalvelin (*Application Server*), joka pyörittää SharePointin sisältämiä palveluita, voi tulla kysymykseen. Farmin muodostama verkkotopologia eli verkon rakenne riippuu organisaation koosta ja tarpeista sekä web-sovellusten määrästä. Pienemmissä ympäristöissä koko farmi voi sijaita yhdessä ja samassa palvelimessa.

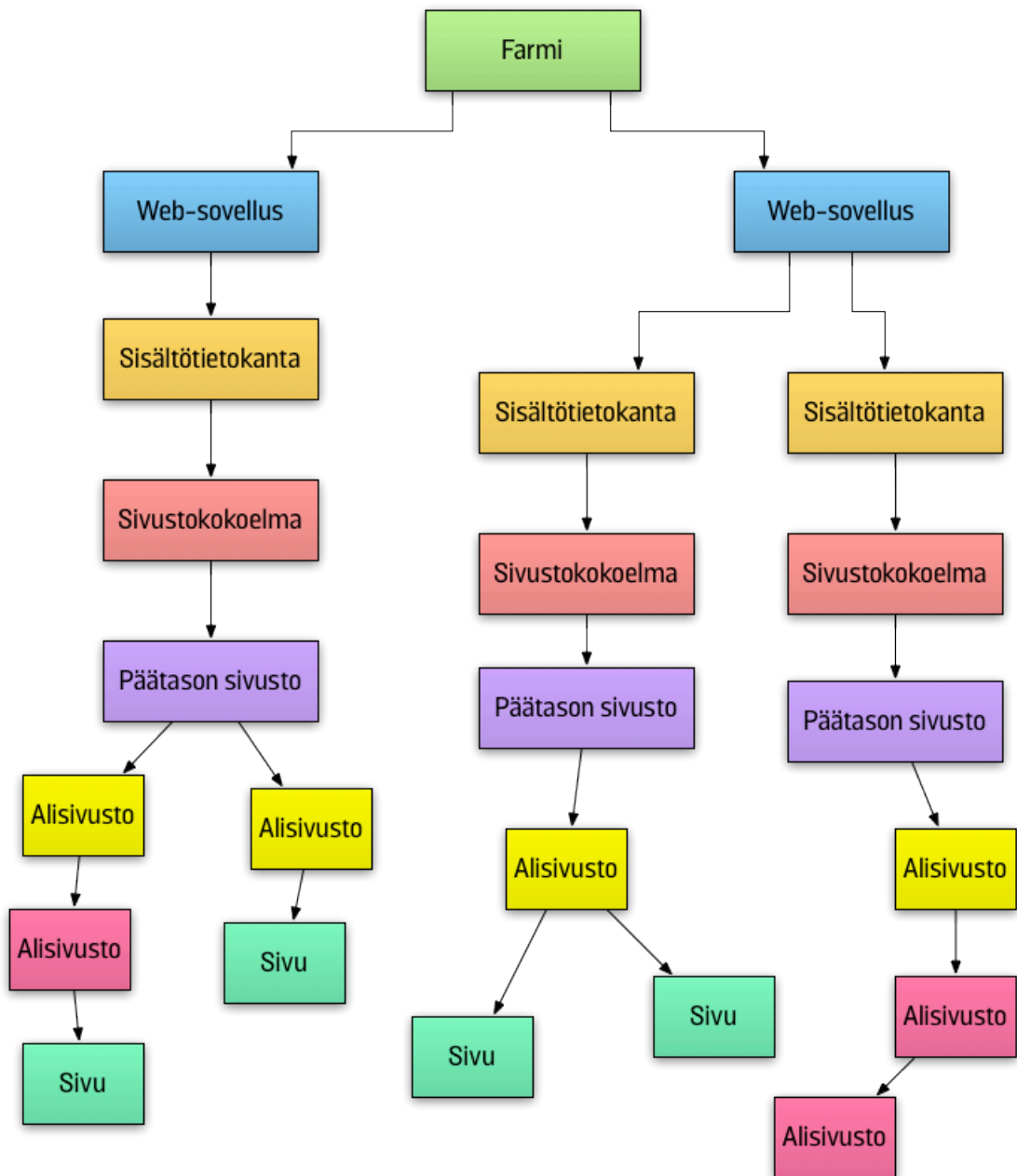
Web-sovellus (*Web Application*) on Internet Information Services (IIS) -www-sivusto, joka on määritetty käytettäväksi SharePointissa. Web-sovellus isännöi yhtä tai useampaa sivustokokoelmaa. Uutta web-sovellusta luotaessa sille joko luodaan uusi tai valitaan käytettäväksi jo olemassa oleva sovellussarja (*Application Pool*). Sen tehtävänä on prosessoida HTTP-pyyntöjä IIS Worker -prosessin (w3wp.exe) kautta. Se eristää web-sovellukset toisistaan niin, että ne käyttävät omia palvelinresurssejaan. Sovellussarja mahdollistaa sen, etteivät epävakaa sovellukset vaikuta muiden sovellusten toimintaan palvelimella, mikäli ne sijaitsevat eri sovellussarjoissa. (Microsoft MSDN 2011.)

Web-sovellus vaatii toimiakseen yhden tai useamman **sivustokokoelman** (*Site Collection*). Se on hierarkkinen www-sivustojen joukko, jota voidaan hallita yhteisesti. Yhden sivustokokoelman sivustoilla on yleensä samoja ominaisuuksia kuten oikeuksia, sivustoelementtejä ja layouteja sekä yleensä sama navigaatio. Ne sijaitsevat kaikki samassa sisältötietokannassa. Jokaisella sivustokokoelmalla on ylimmän tason sivusto (*Top-Level Site*), joka voi sisältää yhden tai useamman alisivuston. Alisivustoilla voi olla alisivustoja ja alisivustojen alisivustoilla alisivustoja ja niin edelleen. Suositeltu sivustojen maksimimäärä yhden sivustokokoelman sisällä on 250 000 kappaletta. Sivustokokoelman hallitsijalla (*Site Collection Administrator*) on oikeus hallita ”omistamaansa” sivustokokoelmaa, mutta ei muita sivustokokoelmia. (Naji 2012.) Sivustokokoelma voidaan tunnistaa joko niin sanotusta GUID-tunnuksesta tai ylimmän tason sivuston verkko-osoitteesta (Microsoft MSDN 2013a).

Sivusto (*Site*) on www-sivujen kokoelma, joka varastoi tietoa. Se toimii tietovarastona esimerkiksi dokumenteille, blogeille, keskusteluille, tapahtumille, tehtäville tai muulle aineistolle. Sivusto tarjoaa mahdollisuuden rajoittaa käyttäjien pääsyä tiettyyn sisältöön. Yksi sivustokokoelma voi pitää sisällään useita sivustoja, joista yksi on aina päätason sivusto, ja niillä voi taas olla monia alisivustoja. (Joy 2010.)

Alisivusto (*Subsite*) on kokonainen www-sivusto, joka on sijoitettu nimettyyn alihakemistoon toisen sivuston sisällä. Alisivuston ylemmän tason sivusto voi olla sivustokokoelman ylimmän tason sivusto tai toinen alisivusto. (Microsoft MSDN 2013a.) Alisivustolle voi määrittää sille ominaiset käyttäjäryhmät ja käyttöoikeudet. Oletuksena se perii ne ylemmän tason sivustolta.

Kuva 4 esittää SharePoint-järjestelmän perustavaa laatua olevan hierarkian.

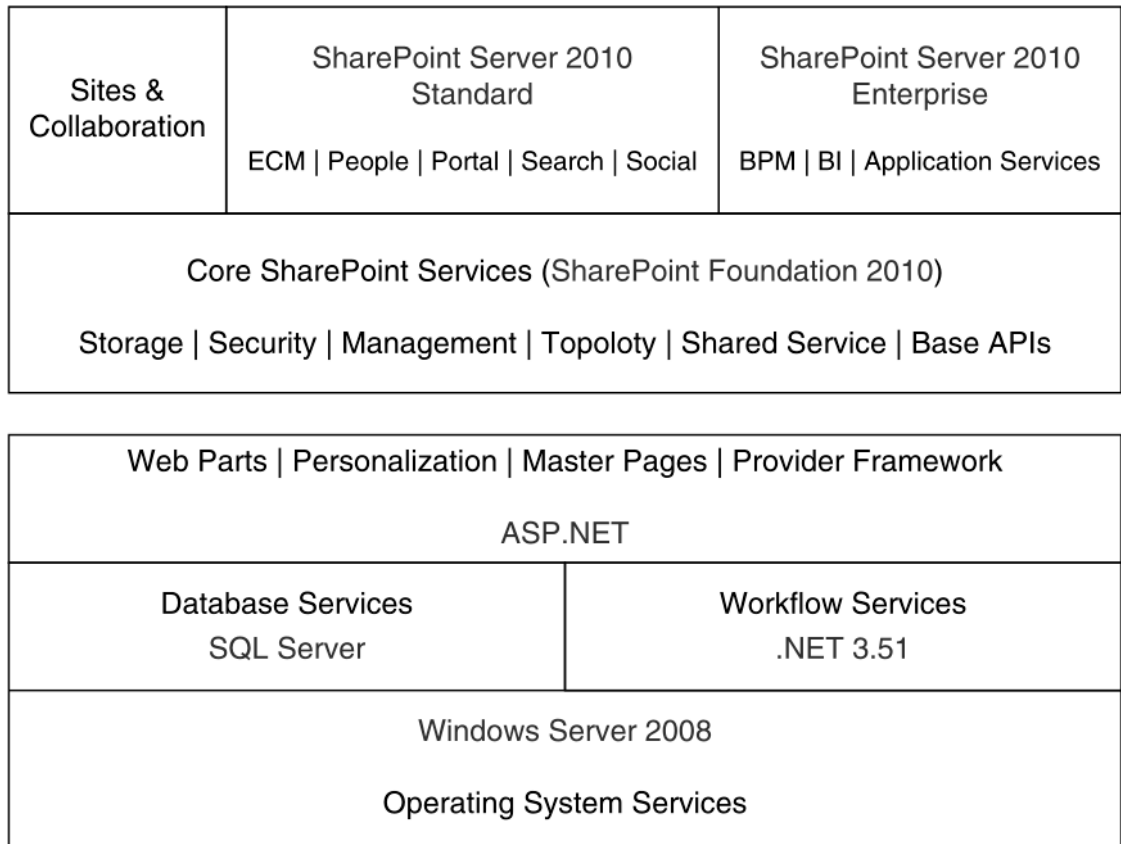


Kuva 4. SharePointin hierarkia

5.2 Versiot

SharePoint 2010 -järjestelmästä on julkaistu kolme eri versiota, jotka eroavat toisistaan lähinnä ominaisuuksiltaan. Niitä ovat ilmainen SharePointin perusversio SharePoint Foundation 2010 ja sen päälle rakennettu kaupallinen SharePoint Server 2010, josta on olemassa kaksi erilaista versiota: Standard ja Enterprise. SharePointin toiminta perustuu .NET-kehityksen versioon 3.5. SharePointin virheetön toiminta edellyttää myös Microsoft SQL Serverin olemassaoloa, koska erilaiset konfiguraatiot ja tietosisällöt tallennetaan sinne. Huomioitavan arvoinen asia on myös se, että SharePoint

2010 toimii vain 64-bittisessä ympäristössä, eikä siitä ole lainkaan olemassa 32-bittistä versiota. Kuva 5 esittää SharePointin teknologiat yhdessä kuvassa.



Kuva 5. SharePoint 2010 -teknologiat (Jamison ym. 2010, 28.)

Versioiden nimeämiskäytännöt eroavat vanhemmista: SharePoint Foundation 2010 tunnettiin sitä edeltävässä versiossa nimellä Windows SharePoint Services 3.0 ja SharePoint Server 2010 nimellä Microsoft Office SharePoint Server 2007. SharePoint 2010 -versiot sisältävät edeltäjiinsä nähden paljon uusia ominaisuuksia.

5.2.1 SharePoint Foundation 2010

SharePoint Foundation 2010 on lisenssitön versio SharePointista. Tämä versio muodostaa SharePoint 2010 -tuotteiden perustan, ja niinpä kaupalliset versiot perustuvat siihen. Koska SharePoint Foundation 2010 sisältää kaikki SharePointin perusominaisuudet ja -toiminnot, kaikki SharePoint-sovellukset sisältävät ainakin osan SharePoint Foundationin sisältämistä ominaisuuksista. Vaikka SharePoint Foundation 2010 on itsessään ilmainen, se vaatii toimiakseen toimintaympäristön, joka on maksullinen (Windows Server 2008 (R2) tuotanto- tai kehitysympäristönä ja / tai Windows 7 tai Windows Vista kehitysympäristönä).

SharePoint Foundation 2010 voidaan periaatteessa nähdä web-pohjaisena ASP.NET-sovelluksena, joka on laajennettu ASP.NET-sivujen HTML-pyyntöjä, .NET-kehiksen API-kutsuja sekä XML-pohjaisia protokollakutsuja eri lähteistä prosessoivasta IIS-verkkosivusta. Sovelluslogiikka (*Business Logic*) toteutetaan sovelluksessa .NET- ja SharePoint-komponenttien avulla. Tieto varastoidaan SQL-tietokantaan. Prosessin tuloksena SharePoint näyttää verkkosivun selaimen ymmärtämässä HTML-muodossa. SharePoint Foundation 2010 varastoi tietokantaan sekä konfiguraation että järjestelmäsivujen sisällön kuten sivujen sisällön ja käyttäjätiedot. Konfiguraatietietokanta luodaan SharePointin asennuksen yhteydessä. Lisäksi keskitetyn hallinnan sisältö varastoidaan Microsoft SQL-palvelimen tietokantaan. (Bamboo Solutions Corporation 2010.)

5.2.2 SharePoint Server 2010 Standard

SharePoint Server 2010 on kaupallinen ja lisensoitu versio SharePoint Foundation 2010:stä. Se sisältää kaikki SharePoint Foundationin ominaisuudet ja arkkitehtuurin. SharePoint Server 2010:tä asennettaessa asennetaan aluksi SharePoint Foundation 2010 ja sen päälle SharePoint Server -version sisältämät mallipohjat, komponentit ja palvelusovellukset. Lisäominaisuuksina SharePoint Foundation 2010 -versioon verrattuna ovat muun muassa kehittyneemmät käyttäjäprofiilien hallintatyökalut, organisaatioympäristöön kehitetyt ominaisuudet, kuten hakutoiminnot, yrityswikisivut ja työyhteisön sosiaaliset verkostot.

5.2.3 SharePoint Server 2010 Enterprise

SharePoint Server 2010 Enterprise on laajennettu versio SharePoint Server 2010 Standard -versiosta. Lisäominaisuuksia on lisätty tietoturvaan, Business Intelligenceen, hallittuihin metatietoihin ja Microsoft Office -sovelluksiin yhdistämiseen. (Pattison ym. 2010, 12 - 13.) Näin ollen SharePoint Server 2010 Enterprise pitää sisällään kaiken mahdollisen, mitä SharePoint-järjestelmä vain voi sisältää.

5.3 Sovellustyypit

SharePoint-ympäristössä toimii kahdenlaisia sovelluksia: farmisovelluksia ja hiekkalaatikkosovelluksia, jotka tulivat uutena ominaisuutena SharePoint 2010 -versioihin. Aiemmissa SharePointin versioissa kaikki sovellukset olivat farmityyppisiä. Merkittä-

vin ero farmi- ja hiekkalaatikkosovelluksissa on niiden käyttöönotossa: farmisovellukset asennetaan ja otetaan käyttöön, kun taas hiekkalaatikkosovellukset kopioidaan palvelimelle ja aktivoidaan (Parmstig 2012). Uutta SharePoint-sovellusta luotaessa Visual Studio 2010:lla käyttäjää pyydetään valitsemaan sovellustyyppi. Valinnan pystyy vaihtamaan vielä myöhemmin projektin asetuksista.

Farmisovelluksia isännöi IIS Worker -prosessi (w3wp.exe). Näin ollen niiden toiminta voi vaikuttaa koko muuhun SharePoint-farmiin. Farmisovellukset voi ottaa käyttöön ainoastaan farmin ylläpitäjä. Kun farmisovelluksessa otetaan käyttöön piirrettä, se vaatii sovellussarjan (*Application Pool*) uudelleenkierrätyksen. (Parmstig 2012.) Sovelluksen sisältämät komponentit sijoitetaan Global Assembly Cacheen tai sovelluksen bin-hakemistoon. Koska farmisovelluksien ulottuvuus asetetaan koko SharePoint-farmin alueelle, niillä on pääsy kaikkiin resursseihin. Farmisovelluksille ei ole rajoituksia SharePointin omien luokkien tai objektien käytössä.

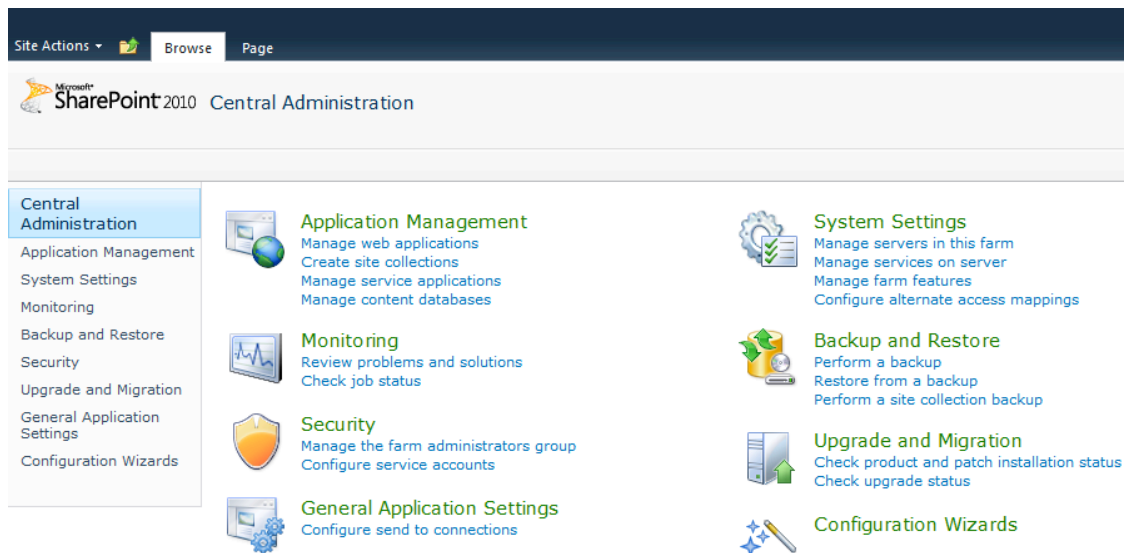
Hiekkalaatikkosovelluksia isännöi web-palvelinprosessista erillinen SPUCWorker-Process.exe- eli User Code Solution -prosessi. Ajettu koodi vaikuttaa ainoastaan sivustokokoelmaan, missä se ajetaan. Lisäksi niiden ajaminen ei vaadi sovellussarjan uudelleenkierrätystä. (Parmstig 2012.) Hiekkalaatikkosovelluksien toiminnassa on lukuisia rajoituksia, jotka on hyvä tiedostaa sovellustyyppiä valittaessa. Ne eivät voi esimerkiksi tehdä muutoksia tiedostojärjestelmään, ne eivät tue Global Assembly Cacheen sijoittamista ja ne ovat vain osittain luotettuja. Nämä sovellukset voi ottaa käyttöön sivustokokoelman hallitsija.

5.4 Hallinta

Keskitetty hallinta (*Central Administration*) on sivu, josta käsin hoituvat kaikki SharePoint-palvelimen hallinnolliset toimet. Tämä sivu luodaan SharePointin asennuksen yhteydessä automaattisesti tiettyyn porttiin. SharePointin Käynnistä-valikon asennuskansista löytyy pikakuvake keskitettyyn hallintaan pääsemiseksi. Keskitetyn hallinnan etusivulla, jonka kuva 6 esittää, on näkyvissä kahdeksan pääkategoriaa:

- **Sovellusten hallinta** (*Application Management*): SharePoint-palvelimella sijaitsevien sovellusten hallinta, sivustokokoelmien ja tietokantojen hallinta, palvelusovellusten hallinta.

- **Valvonta** (*Monitoring*): Ongelmien vianmääritys, ajastettujen tehtävien tilan tarkastelu, raporttien tarkastelu ja hallinta.
- **Turvallisuus** (*Security*): SharePoint-farmin ylläpitäjien, käyttäjäryhmien ja tilien hallinta, salasanan vaihtamisilmoitukset, virusturva-asetukset, ei-sallittujen tiedostopäätteiden hallinta.
- **Yleiset sovellusasetukset** (*General Application Settings*): Ulkoisten palveluiden yhteysasetukset, SharePoint Designer -asetukset, etsimistoiminto, raportointipalvelut, sisällön käyttöönottoasetukset.
- **Järjestelmäasetukset** (*System Settings*): SharePoint-farmin palvelimien ja palveluiden tarkasteleminen ja hallinta, sähköposti- ja tekstiviestiasetukset, farmin ylläpitotoimet.
- **Varmuuskopiointi ja palautus** (*Backup and Restore*): SharePoint-farmin tai sivustokokoelman varmuuskopiointi, varmuuskopiosta palauttaminen, varmuuskopioasetukset.
- **Päivitykset ja siirtäminen** (*Upgrade and Migration*): SharePoint-farmin version tarkistaminen, päivitysten tarkistaminen, tietokannan tilan tarkistaminen, olemassa olevien SharePoint-sovellusten piirteiden tarkasteleminen.
- **Ohjatut määritykset** (*Configuration Wizards*): Eri toimenpiteiden suorittaminen ohjatuin menetelmin.



Kuva 6. SharePoint 2010:n keskitetty hallintasivu

5.5 Sivut

SharePoint-sovellus on aina web-pohjainen, ja tämän tyyppinen ohjelma koostuu aina vaihtelevasta määrästä www-sivuja.

5.5.1 Perustyyllisivut

Perustyyli- eli master-sivut ovat ASP.NET 2.0 -kehyksessä esitelty konsepti, jonka avulla voidaan luoda johdonmukainen, koko web-sovelluksen kattava käyttöliittymä. Käytännössä kaikki yhden web-sovelluksen sivut voivat mukailla perustyyllisivun noudattamaa kaavaa muun muassa asetteluissa ja sivustoelementeissä. Perustyyllisivu- ja voi olla useampia. Niissä määritellään esimerkiksi, mitkä elementit web-sovelluksen sivuilla ovat yhteisiä ja millainen sivun perusrakenne on. Myös koko web-sovelluksen yhteiset CSS- ja JavaScript-määrytykset kannattaa sijoittaa perustyyllisivulle. Yksittäisillä sivuilla määritellään, minkä perustyyllisivun rakennetta ne noudattavat. Niille voidaan lisätä omaa sisältöä perustyyllisivulla määriteltyihin placeholder- eli paikanpitäjäelementteihin. (Sayed ym. 2012, 27.) Perustyyllisivut ovat laajasti tuettuja SharePointissa ja suositeltava käytäntö hyödynnettäväksi sovelluskehityksessä.

5.5.2 Application Page -sivut

Application Page on sivutyyppi, joka sijoitetaan SharePoint-farmin tiedostojärjestelmään; tarkemmin sanottuna SharePointin juurihakemistossa sijaintiin `$SharePointRoot$\14\TEMPLATE\LAYOUTS\`. Tämä kyseinen sijainti on jaettu SharePoint-sovelluksessa Internet Information Services:n virtuaalisen `_layouts-`kansion kanssa. Application Page -tyyppiset sivut ovat tämän vuoksi kaikkien saman farmin SharePoint-sovellusten tavoitettavissa. Niitä käytetään yleensä sivuina, jotka on tarkoitettu erilaisten konfiguraatioiden tekemiseen. Application Page -tyyppiset sivut eivät tue käyttäjien muutoksia, eikä niitä voi näin ollen muokata käyttöliittymästä saati SharePoint Designer 2010 -ohjelmasta käsin. Ne muistuttavat ASP.NET-kehiksen perinteisiä ASP.NET-sivuja. Application Page -sivut sijoitetaan yleensä SharePoint-tiedostopakettin (*Windows Solution Package, WSP*) yhteyteen. Näille sivuille voi kirjoittaa avoimesti ohjelmakoodia.

5.5.3 Site Page -sivut

Site Page on sivutyyppi, joka sijoitetaan SharePoint-sovelluksen sisältötietokantaan. Se voitaisiin määrittellä myös tavalliseksi sisältösivuksi. Site Page -tyyppinen sivu ei sijaitse fyysisesti SharePoint-farmin tiedostojärjestelmässä, ja näin ollen sivut ovat tavoitettavissa vain yhden sivustokokoelman sisällä. Site Page -tyyppiset sivut tukevat käyttäjän muokkauksia käyttöliittymän kautta tai SharePoint Designer 2010 -ohjelmalla. Site Page -sivuja on olemassa erityyppisiä, esimerkiksi Web Part- ja Wiki-sivut. Omia Site Page -sivutyyppejä voi myös luoda itse muokkaamalla olemassa olevia tyyppisiä. Sivuille ei voi lisätä avoimesti ohjelmakoodia SharePointin turvallisuusasetuksien vuoksi, eikä se ole suositeltavaakaan. Se on kuitenkin mahdollista määrittelemällä SharePoint-sovelluksen määritystiedostoon `<PageParserPaths />`-lohkon ja sen sisälle sivut, joilla koodia halutaan suorittaa. Normaalisti Site Page -sivuilla toiminnallisuus toteutetaan web-osien avulla, jotka sijoitetaan sivuille ennalta määriteltäviin paikkoihin (*Web Part Zones*).

5.6 Web-osat

Web-osat ovat keskeinen SharePointin ominaisuus. Ne voidaan määrittellä itsenäisiksi pienoishjelmiksi, jotka liitetään osaksi www-sivua ennalta varatuille paikoille. Web-

osat ovat pohjimmiltaan ASP.NET 2.0 -kehyksestä lähtien järjestelmässä mukana olleita web-kontrolleja, jotka periytyvät `System.Web.UI.WebControls`-luokasta, mutta sisältävät tavallisiin web-kontrolleihin verrattuna erityisominaisuuksia. Koska SharePoint 2010 rakentuu .NET-kehiksen 3.5-version päälle, mitä tahansa sen kirjastosta löytyvää komponenttia voi käyttää sovelluskehityksessä. Tämä pätee myös web-osiin. Käyttäjät voivat valita, mitä web-osia haluavat sivulla näytettävän ja miten web-osat sivulla näytetään. Web-osat voivat myös keskustella keskenään. (Wilén 2011, 3 - 6.) SharePoint 2010:ssä on paljon valmiita web-osia ja lisää voi ladata ilmaiseksi tai maksua vastaan internetistä. Web-osat jaotellaan SharePoint 2010:ssä eri kategorioihin tai gallerioihin. Ne sisältävät `Elements.xml`-tiedoston, jossa niille määritellään, mihin ryhmään ne halutaan sijoittaa. Web-osat sisältävät lisäksi `.webpart`-päätteisen tiedoston, joka määrittelee muun muassa web-osan nimen ja kuvauksen.

5.7 Piirteet

Piirteiden avulla SharePoint-sivulle voidaan lisätä erilaisia toiminnallisuuksia. Piirteet voivat sisältää esimerkiksi listamäärittelyjä, tapahtumankäsittelijöitä, työkulkujia, web-osia ja omia muokkauksia. Piirteiden sisältämiin komponentteihin ja niiden sijainteihin viitataan `Feature.xml`-tiedostossa, jossa piirteen ominaisuudet määritellään. Ne sisällytetään sovelluksessa SharePoint-tiedostopakettiin. Jokainen piirre sijoitetaan SharePoint-farmilla käyttöön otettaessa omaan kansioonsa hakemistossa `$SharePointRoot$\14\TEMPLATE\FEATURES\`. (Krause ym. 2010, 481.)

Piirteet voivat toimia tietyissä ulottuvuuksissa (*Scope*). Vaihtoehtoja piirteen ulottuvuudeksi ovat Web, Site, WebApplication ja Farm. Farm- ja WebApplication-ulottuvuudet eivät ole sallittuja hiekkalaatikkosovelluksissa. Ulottuvuuden määrittäminen tehdään Visual Studion kautta. Piirteille voi lisätä myös tapahtuman vastaanottimia (*Event Receivers*). Ne ovat luokkia, jotka periytyvät `SPFeatureReceiver`-luokasta ja käsittelevät piirteisiin liittyviä tapahtumia. (Sayed ym. 2012, 56 - 59.)

5.8 Tietolähteet

SharePoint-sovelluksien sisältämä tieto voi sijaita useissa eri paikoissa. Järjestelmä tarjoaa tehokkaat työkalut halutun tiedon löytämiseksi.

5.8.1 Listat ja SQL-taulut

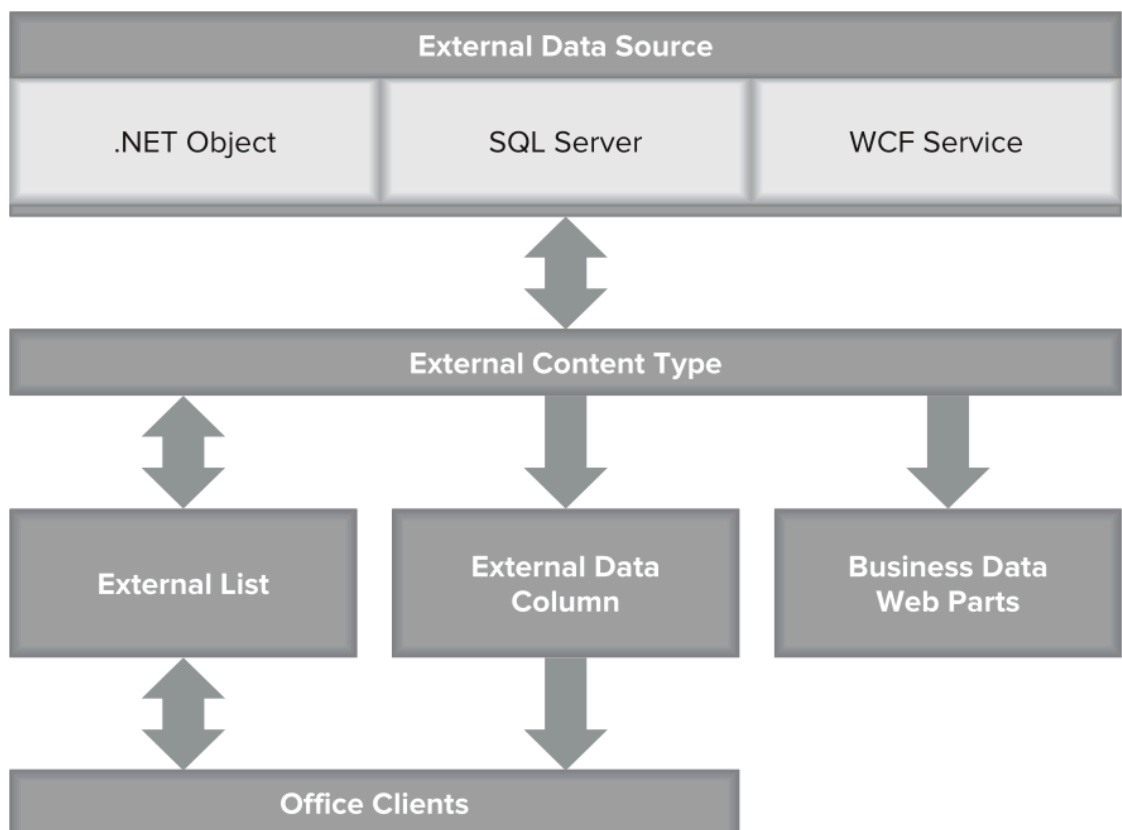
SharePoint-sovelluskehittäjät voivat valita tiedon järjestelmälliseen varastointiin joko SharePointin tarjoamat listat tai perinteiset SQL-taulut. Listat toimivat pohjimmiltaan kuten SQL-taulutkin: ne varastoivat dataa riveihin ja sarakkeisiin, joilla on omat tietotyypinsä. Listoja hallitaan graafisesti SharePointin käyttöliittymän kautta, kun taas SQL-tietokannassa sijaitsevan datan hallitseminen vaatii hallintatyökalujen rakentamisen ohjelmoinnin keinoin. Molemmilla tietovarastoilla on omat hyvät ja huonot puolensa, jotka esittää taulukko 3.

Taulukko 3. SharePointin listojen ja SQL-tietokantojen etuja ja haittoja (Microsoft MSDN 2013c.)

SharePointin listat	
+	Integroitu SharePointiin: tietoturva ja käyttöoikeudet periytyvät listoille
+	Tuki versionhallinnalle
+	Luominen ja datan hallinta tapahtuvat SharePointin omassa käyttöliittymässä
+	Microsoft Office -integraatio (Microsoft Excel ja Microsoft Access)
+	Tuki työnkuluille (Workflows) ja SharePointin hakutoiminnoille
-	Monimutkaisten datayhteyksien käsittely hankalahkoa (Many-To-Many)
-	Datan varastoimiselle rajoituksia
SQL-tietokannan taulut	
+	Suuren datamäärän ja monimutkaisten tietorakenteiden varastointi
+	SQL-transaktiot eli peräkkäiset tapahtumat
+	Joustavuus ja tehokkuus → toimivaksi todettu, ”tuttu ja turvallinen” ratkaisu
-	Datan hallintatyökalut rakennettava itse ohjelmoinnin keinoin
-	Sovelluksen tietoturvaa parannettava itse, SharePoint ei tarjoa tähän apuja

5.8.2 Business Connectivity Services

Business Data Connectivity (BDC), joka tunnettiin aiemmin nimellä Business Data Catalog, on palvelu, joka mahdollistaa ulkoisten tietojärjestelmien lukemisen ja kirjoittamisen sekä SharePoint-järjestelmässä että Microsoft Officen ohjelmistoissa. Ulkoisia tietojärjestelmiä voivat olla esimerkiksi web-palvelut, tietokannat ja .NET-kehityksen sovelluskomponentit. Niiden sisältämää dataa voidaan käsitellä suoraan sekä yhteydellisessä että yhteydettömässä tilassa. (Microsoft MSDN 2010.) Business Data Connectivity on keskeinen osa Business Connectivity Services -kokonaisuudessa. Sen vastuulla on yhteyden muodostaminen ulkoisiin datalähteisiin. Kuva 7 näyttää Business Connectivity Servicesin pelkistetyn periaatekuvan.



Kuva 7. Business Connectivity Services -ominaisuuden pintapuolinen rakennekuva (Perran ym. 2011, 560.)

SharePoint Foundation 2010 -versiossa Business Connectivity Services -mekanismille on asetettu rajoituksia verrattuna SharePoint Server 2010 -versioon. SharePoint Foundation 2010 -versiossa ei ole muun muassa seuraavia ominaisuuksia:

- Secure Store -palvelu esimerkiksi tiedon hakemiseksi ulkoisesta tietokannasta SQL-autentikoinnin kautta
- Business Data -web-osat
- Profiilisivut
- Rich Client Integration. (Bordner 2011.)

5.9 Objektimallit

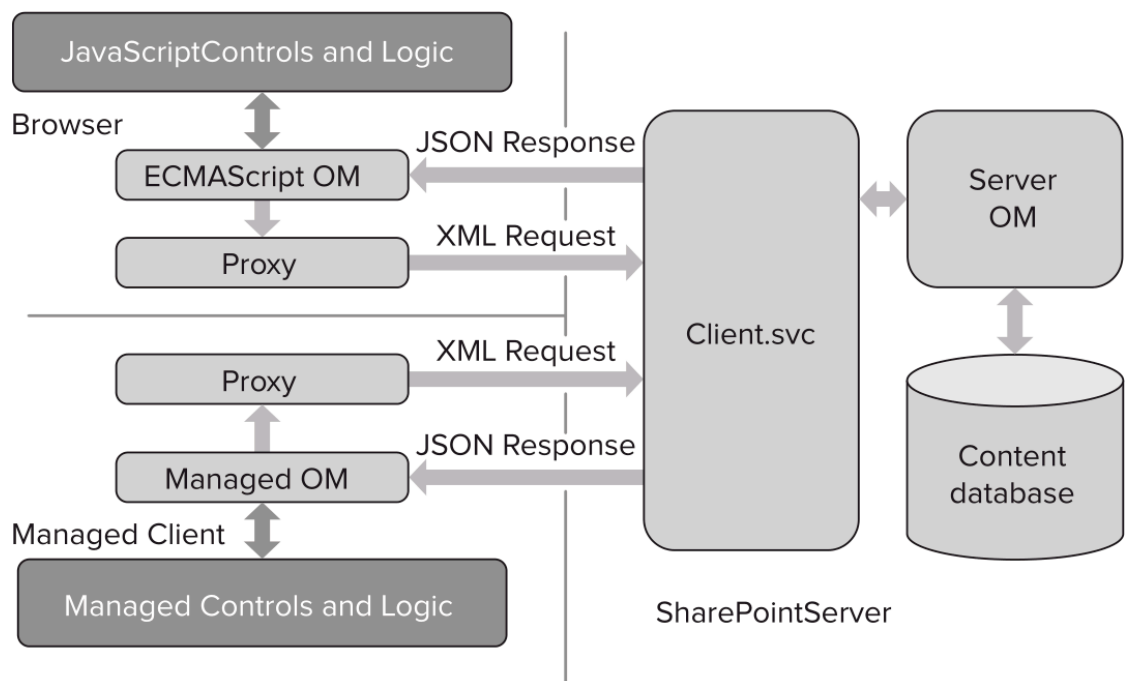
SharePoint-ympäristössä on yhteensä neljä objektimallia: palvelinympäristössä ajettava Server Object Model ja asiakaspään objektimallit .NET- ja Silverlight-sovelluksille sekä JavaScript-objektimalli. Palvelinympäristön objektimalli on kaiken perusta, ja se on määritelty kokonaisuudessaan `Microsoft.SharePoint.dll`-komponentissa. Palvelinobjektimalli sisältää joukon luokkia, jotka muodostavat yhdessä API:n lähes kaiken tiedon hallintaan SharePoint-ympäristössä. Tärkeimpinä luokkina mainittakoon ne, jotka taulukko 4 esittää.

Taulukko 4. SharePointin palvelinobjektimallin tärkeimmät luokat ja niitä vastaavat komponentit (Paul 2010.)

Luokka	SharePoint-komponentti
SPFarm	SharePoint-farmi
SPServer	SharePoint-palvelin
SPWebApplication	SharePoint-web-sovellus
SPSite	SharePoint-sivustokokoelma
SPWeb	SharePoint-sivu
SPControl	SharePoint-kontrolli
SPList	SharePoint-lista
SPDocumentLibrary	Sharepoint-dokumenttikirjasto
SPContentType	SharePoint-sisältötyyppi
SPUser	SharePoint-käyttäjä
SPException	SharePoint-virhe

Asiakaspään eli selaimella suoritettavan koodin yhteydessä käytettävät objektimallit tulivat uutena ominaisuutena SharePoint 2010 -versioihin. Niiden avulla voidaan tuoda dataa SharePoint-palvelimelta asiakaspäässä. Asiakaspään objektimallien sisältö perustuu palvelinobjektimalliin, sisältäen vain asiakaspäälle olennaiset osuudet. Tämän myötä objektimallista muodostuu kevyempi kokonaisuus ja näin ollen kirjastojen latausajat pienenevät (Krause ym. 2010, 711).

Jotta asiakaspään objektimallia voidaan käyttää, täytyy koodissa viitata tiettyihin komponentteihin. JavaScript- eli yleisimmin ECMAScript-objektimalli tarvitsee viittauksen JavaScript-kirjastoihin ja .NET- sekä Silverlight-sovellukset järjestelmän komponentteihin eli DLL-tiedostoihin. Asiakaspään objektimallien ja SharePoint-palvelimen sisältötietokannan välissä on WCF-palvelu nimeltään `Client.svc`, joka huolehtii yhteydenpidosta. Data lähetetään tälle palvelulle XML-pyyntönä ja palauteaan SharePoint-palvelimelta JSON-muodossa, joka muunnetaan sopivaan muotoon riippuen käytettävästä objektimallista. Kuva 8 esittää periaatekuvan asiakaspään objektimallin toiminnasta.



Kuva 8. SharePoint 2010:n asiakaspään objektimallin toimintakaavio (Rizzo ym. 2010, 142.)

5.10 Tietoturva

SharePointin turvallisuusmalli on vahvasti kytköksissä Windowsiin, IIS:ään ja ASP.NETiin. Seuraavissa kappaleissa selostetaan hieman tarkemmin SharePointin tietoturvaan liittyviä kokonaisuuksia.

5.10.1 Todennusmenetelmät

Todennus- eli autentikointiprosessi käsitellään käytännössä SharePointin sijasta Internet Information Services (IIS):n puolella. Onnistuneen tunnistautumisen jälkeen SharePoint ohjaa käyttäjän sivulle, jonka tarkastelemiseen hänen oikeutensa riittävät. Mikäli käyttäjää ei tunnistettu, hänen pääsynsä sivulle estetään. Seuraavassa käydään läpi hieman tarkemmin SharePoint-ympäristön yleisimpiä todennustyyppejä.

- **Windows Authentication** on yleisin autentikointitapa intranet-ympäristössä. Se mukailee Active Directoryn käyttäjätietoja käyttäjien tunnistamiseen. Windows-autentikoinnissa käytetään IIS:ssä määritettyä protokollaa. Näitä protokollia ovat NTLM (*NT LAN Manager*), Kerberos, sertifikaatit sekä Basic- ja Digest-protokollat. Käyttäjätileihin liittyvät suojauskäytännöt, esimerkiksi salasanan vahvuusvaatimus ja käyttäjätilien voimassaolo, määritellään Active Directoryssa. Niitä ei määritellä erikseen SharePointissa. Käyttäjien oikeudet haetaan Active Directorysta ja otetaan käyttöön SharePoint-sivustolla käyttäjän tunnistautuessa. (Krause ym. 2010, 134.)
- **Claims-Based Authentication** perustuu Windows Identity Foundation -kehykseen, joka tunnettiin aiemmin nimellä Security Token Service (STS). Se tuli uutena ominaisuutena SharePoint 2010 -versioon. Menetelmään perustuva tunnistautuminen tapahtuu niin kutsutun turvatunnisteen (*Security Token*) avulla, joka sisältyy käyttäjän tunnistetietoihin. *Claim* on eräänlainen identiteetin ominaisuus, jonka tunnistustietojen välittäjä (*Identity Provider*) tarjoaa käyttäjälle. Claims-Based-menetelmää voidaan käyttää käyttäjien kirjautumisessa alustasta riippumatta: käyttäjät voivat tulla joko Windows-pohjaisista tai ei-Windows-pohjaisista järjestelmistä. Käyttäjän tunnistautuessa sitä mukailevasta järjestelmästä SharePoint ohjaa hänet sopivalle tunnistustietojen välittäjälle, joka suorittaa autentikaation kyseisessä järjestelmässä ja toimittaa turva-

tunnisteen käyttäjälle. Tämä turvatunniste välitetään SharePointin Security Token Servicelle, joka palauttaa käyttäjälle SharePointia varten luodun SAML-pohjaisen turvatunnisteen. Tämä tunniste on käyttäjäkohtainen, ja sillä käyttäjä voi todistaa identiteettinsä saadakseen luvan kirjautua järjestelmään. Tunnistustietojen myöntäjiä on mahdollisuus myös luoda itse. (Maxwell 2010.)

- **Forms-Based Authentication** on autentikointimenetelmä, jota käytetään yleensä niissä tilanteissa, kun tavallinen Windows-autentikointi ei jostain syystä tule kysymykseen. Sitä käytetään tyypillisesti extranet-järjestelmissä, koska se ei vaadi käyttäjätilien olemassaoloa Active Directoryssa. Forms-Based-autentikointi säilyttää käyttäjien tunnistetietoja esimerkiksi itse määritellyssä LDAP-hakemistossa tai SQL-palvelimella. (Krause ym. 2010, 136.) Se toimii ainoastaan Claims-Based-autentikoinnin kanssa SharePoint 2010 -versiosta lähtien. Tämän menetelmän käsittely tapahtuu IIS:n sijasta ASP.NET-kehyksessä.
- **Anonymous Access** lasketaan usein Windows-autentikoinnin muodoksi, sillä se yhdistää tuntemattoman käyttäjän tuntemattomaan käyttäjätiliin. Käytännössä tämä tarkoittaa sitä, että käyttäjän oikeudet eivät rajoita hänen pääsyään SharePoint-sovellukseen. Anonymous Access -menetelmää käytetään yleisimmin internet-sivuilla. SharePoint 2010:ssä se ei ole oletuksena päällä. Menetelmä on oletusmäärittäyksiltään sellainen, että SharePoint-palvelimen oletetaan sijaitsevan intranetissä. Anonymous Access -menetelmän piirissä olevilla käyttäjillä on joitakin rajoituksia todennettuihin käyttäjiin verrattuna. Oletusarvoisesti heillä on lupa vain lukea sisältöä. (Krause ym. 2010, 135 - 136.)

5.10.2 Käyttöoikeudet

SharePoint-ympäristössä oikeustaso (*Permission Level*) kuvaa joukkoa oikeuksia, jotka on myönnetty käyttäjälle tai ryhmälle tiettyyn SharePoint-objektiin. Tällainen objekti voi olla esimerkiksi sivu, lista tai dokumentti. SharePointissa on oletuksena joitakin oikeustasoja, mutta niitä voi myös tehdä uusia. Tämä on suositeltava tapa, sillä järjestelmän oletustasojen muuttaminen voi tehdä järjestelmän oikeuksista sekavan kokonaisuuden. SharePointin oletusoikeustasot esittää taulukko 5.

Taulukko 5. SharePoint 2010:n oletusoikeustasot (Perran ym. 2010, 343.)

Oikeustaso	Oikeudet
Restricted Read	Käyttäjä voi tarkastella sivuja ja dokumentteja, mutta ei voi tarkastella niiden historiatietoja tai käyttäjäoikeuksia.
Read	Käyttäjä voi tarkastella sivuja ja listaelementtejä sekä ladata dokumentteja.
Contribute	Käyttäjä voi tarkastella, lisätä, päivittää ja poistaa listaelementtejä ja dokumentteja.
Design	Käyttäjä voi tarkastella, lisätä, päivittää, poistaa, hyväksyä ja muokata listaelementtejä ja sivuja.
Approve	Käyttäjä voi muokata ja hyväksyä sivuja, listaelementtejä ja dokumentteja.
Manage Hierarchy	Käyttäjä voi luoda sivuja ja muokata sivuja, listaelementtejä ja dokumentteja.
Full Control	Käyttäjällä on täydet oikeudet.
Limited Access	Käyttäjä voi oikeuksien mukaan tarkastella tiettyjä listoja, kirjastoja, listaelementtejä, kansioita tai dokumentteja.

Käyttäjryhmien (*Site Group*) jäsenyys määrittelee henkilöiden roolin organisaatiossa. Ne voivat sisältää Windowsin (Active Directory) käyttäjryhmiä, ASP.NET Forms Authentication -ryhmiä (käyttäen jäsenyyden välittäjälle (*Role Provider*) määriteltyjä rooleja) tai yksittäisiä käyttäjiä, joilla on käyttäjätili paikallisella palvelimella tai Windows-domainissa. (Bamboo Solutions Corporation 2011, 18.) Oletuksena SharePoint 2010 sisältää kolme käyttäjryhmää. **Site Members** -ryhmällä on ”Contribute”-oikeus, **Site Owners** -ryhmällä ”Full Access”- ja **Site Visitors** -ryhmällä ”Read”-oikeus. Omien käyttäjryhmien tekeminen ja olemassa olevien käyttäjryhmien muokkaaminen on mahdollista.

SharePoint 2010:ssä käyttäjäryhmät muodostetaan oletuksena sivustokokoelmatasolle, ja oikeudet periytyvät alisivustoille ylimmän tason sivustolta. Joissain tapauksissa alisivustoille on tarvetta luoda vain niille ominaisia oikeuksia. Tällöin oikeudet voidaan katkaista. Tämä onnistuu SharePointin käyttöliittymän sivuasetuksista tai SharePoint Designer 2010 -ohjelman avulla. Oikeuksia voidaan määrittää myös tarkemmin listojen ja jopa yksittäisten kohteiden tasolla. Oletuksena yksittäinen kohde perii listan tai kirjaston oikeudet, missä se sijaitsee.

6 DANISCO ORIENTATION -SOVELLUS

Opinnäytetyön käytännön toteutuksena rakennettiin Microsoft SharePoint 2010 -ympäristöön sähköinen perehdytyssovellus, jonka loppukäyttäjänä oli Kotkassa sijaitseva Danisco Sweeteners Oy:n tehdas. Perehdytyssovellus, nimeltään Danisco Orientation, koostuu kahdesta eri sivustokokonaisuudesta: varsinaisesta perehdytyssovelluksesta ja ylläpitäjien sivustosta.

Perehdytysohjelma toteutettiin kohdeyrityksen tarpeiden pohjalta ja tavoitteena oli helppokäyttöisyys ja toimivuus. Sovelluksen perusominaisuudet rakentuivat hyvin pitkälti ASP.NET-kehiksen päälle, mutta siinä oli hyödynnetty myös joitakin SharePointin tarjoamia ominaisuuksia.

6.1 Käyttöliittymä

Danisco Orientation -sovelluksen käyttöliittymästä oli tavoitteena luoda yksinkertainen ja käyttäjäläheinen, sillä sovelluksen tulevat käyttäjät käyttävät sitä erilaisista lähtökohdista. Toisille tietotekniikka on tutumpaa kuin toisille. Tämä tiesi sitä, että omien tietojen käsittely ja varsinainen perehdytysosuus tuli rakentaa niin, että niiden läpivieminen oli hoidettavissa mahdollisimman sujuvasti.

Koska sovellusta oli määrä käyttää ainoastaan Internet Explorer -selaimilla (vähimmäisvaatimus Internet Explorer 8 -versio), se toi käyttöliittymän suunnitteluun joitakin rajoituksia. CSS-tyylien rakentamisessa ei voitu käyttää uusimman CSS3-version ominaisuuksia, ja sivustolla käytettyjen, pääasiassa jQuery-kirjastoa käyttävien JavaScript-lisäpalikoiden toimivuus oli varmistettava erikseen. Tieto siitä, että muut se-

laimet jäävät pois laskuista, myös helpotti kehitystä. Sovelluksen testaaminen toisissa selainympäristöissä jäi näin ollen kokonaan pois.

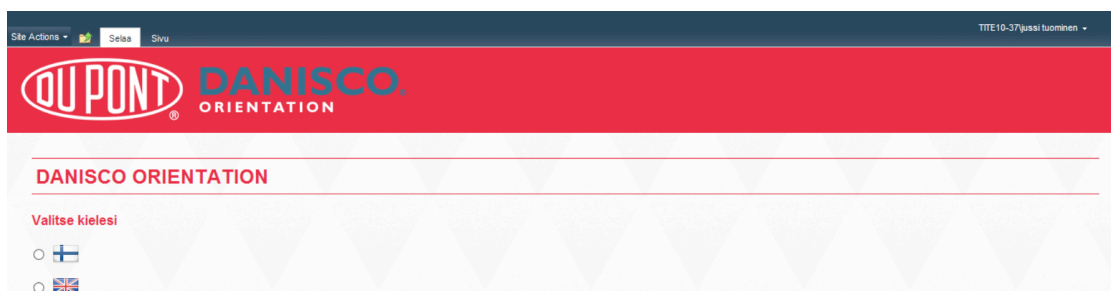
Sekä perehdytys- että ylläpitosivusto noudattivat käyttöliittymässään samaa kaavaa. Sivustojen pääelementit sijoittuvat näytöllä samoihin paikkoihin. Kuva 9 näyttää sivun pääelementtien asettelumallin. Merkittävin ero sivustojen ulkoasujen välillä on värimaailma, joka on puettu perehdytysosiossa yrityksen väreihin, kun taas ylläpito-puolen pääväreiksi valittiin sininen ja vihreä. Perehdytysosion käyttöliittymä toteutettiin mukailemaan yrityksen värimaailmaa, koska yrityksen omat säännökset määräävät näin. Värimaailman muodostuksessa pyrittiin pitämään eri värien määrä maksimissaan kolmessa.



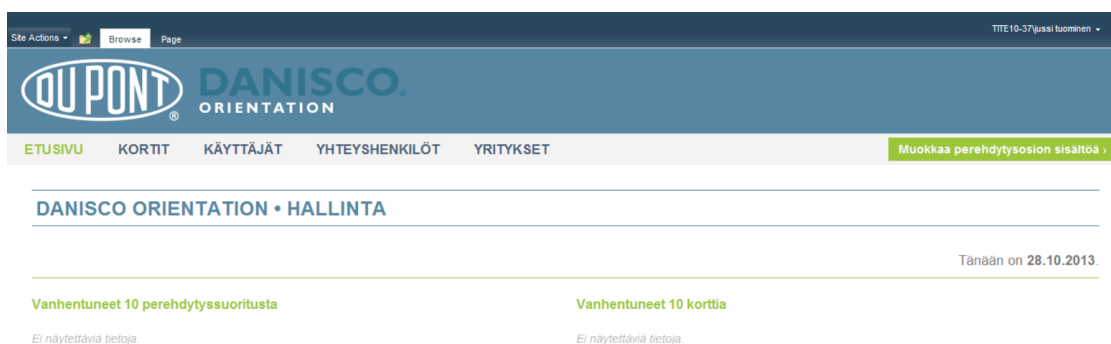
Kuva 9. Perehdytyssovelluksen pääelementit

Molempien sivustokokonaisuuksien sivuasettelu eli layout toteutettiin SharePointin perustyylisivujen avulla ja kummallekin sivustolle omilla CSS-tyylimäärittelyillä. Paras työkalu layoutin kehittämiseksi oli SharePoint Designer 2010. CSS- ja JavaScript-tiedostoille SharePoint-sivuston kansiohierarkiassa on oma dokumenttikirjastonsa nimeltään *Style Library*. Kuvatiedostoille kirjastona on *Site Assets*. Koska näitä tiedostoja ei ollut tarpeen käyttää muualla kuin tämän sivustokokoelman sisällä, oli aiheellista sijoittaa ne sivustokokoelman omiin dokumenttikirjastoihin, joihin voitiin lisäksi viitata koodissa suhteellisin polkumäärittelyksin. Kuva 10 näyttää perehdytysosion ulko-

asun ja kuva 11 ylläpitosivuston ulkoasu. Käyttöliittymän yksityiskohtaisempi tarkastelu ei tässä raportissa ole relevanttia, sillä tarkempia esimerkkejä tästä aiheesta on saatavissa raportin liitteenä olevasta käyttöohjeesta.



Kuva 10. Perehdytys sivuston ulkoasu



Kuva 11. Perehdytyssovelluksen ylläpitosivuston ulkoasu

6.2 Rakenne

Sovelluksen sivustorakenteessa, sekä perehdytysosiossa että hallintaosiossa, on havaittavista kaksi päähaaraa: käyttäjät ja perehdytys. Osa sivuista keskittyy käyttäjiin liittyvän datan käsittelyyn, kun taas toiset sivut perehdytykseen liittyvään dataan. Osassa sivuissa tietoja luonnollisesti yhdistellään, esimerkiksi silloin kun tarkastellaan tietyn käyttäjän perehdytyksen suorituspäivää tai kun perehdytystä suoritetaan, jolloin käyttäjän yksilöity ID-numero seuraa koko prosessin ajan mukana. Kuva 12 esittää kaavion, jossa hahmotetaan käyttäjien etenemismahdollisuudet perehdytys sivustolla.



Kuva 12. Danisco Orientation -sovelluksen perehdytysosion etenemiskaavio

Kuten kuvasta 12 voidaan nähdä, palaavan käyttäjän kohdalle oli rakennettava jonkinlainen tunnistusmenetelmä. Oli itsestään selvää, että ohjelman tuli jotenkin rajoittaa pääsy käyttäjien henkilökohtaisiin tietoihin. Mikäli tunnistusta ei olisi olemassa, kuka tahansa pystyisi tahallisesti tai tahattomasti manipuloimaan toisten käyttäjien tietoja

tai suorittamaan perehdytystä toisen käyttäjän nimissä. Tämän vuoksi jokaiselle työntekijälle muodostetaan sovelluksessa uuden käyttäjän luonnin yhteydessä viisinumeroinen tunnistusnumero, jonka avulla palaava käyttäjä pääsee tarkastelemaan sovellukseen tallennettuja tietojaan. Jotta numero säilyisi käyttäjien muistissa, se tulostetaan perehdytyksen suorittamisen jälkeen muodostettavan todisteen taakse.

Ylläpitosivustolla datan hallintatyökalut ryhmiteltiin niin, että perehdytysosion informaation (moduulien, aineistojen, kysymysten ja vastausten) hallinta eriytettiin käyttäjiin liittyvän informaation hallinnasta. Tämä jaottelu selkeytti työkalujen sijoittelua hallintasivustolla ja toi siihen ripauksen käyttäjäläheisyyttä, koska tällöin ylläpitäjä voi tehdä toimenpiteet osa-alue kerrallaan. Tarkemmin työkalujen sijaintia ja toimintaa on selostettu tämän raportin liitteenä olevassa käyttöohjeessa.

Sivutyyppeinä perehdytyssovelluksessa käytettiin yksinomaan Site Page -tyyppisiä sivuja. Tämä oli looginen menettelytapa, koska tietojen hallintaan suunniteltuja web-osa ei voi käyttää Application Page -tyyppisillä sivuilla. Lisäksi Site Page -sivut toimivat sovelluksen sisältötietokannasta käsin, eivätkä sijaitse fyysisesti web-palvelimen tiedostojärjestelmässä. Ajatus siitä, että sivut olisivat tavoitettavissa jokaisesta samalla SharePoint-farmilla sijaitsevasta web-sovelluksesta, ei kuulostanut hyvältä. Sivut luotiin SharePoint Designer 2010 -ohjelmalla niille tarkoitettuun *Site Pages* -kirjastoon Web Part -sivu -tyyppisinä.

6.3 Ominaisuudet

Danisco Orientation -sovellusta voisi kuvailla ”ASP.NET-vivahteiseksi SharePoint-sovellukseksi”. Sen rakentamisessa pyrittiin ominaisuuksien puolesta kokonaisvaltaiseen ratkaisuun ja tämän myötä kaikki mahdolliset sudenkuopat sovelluksen toiminnassa pyrittiin poistamaan.

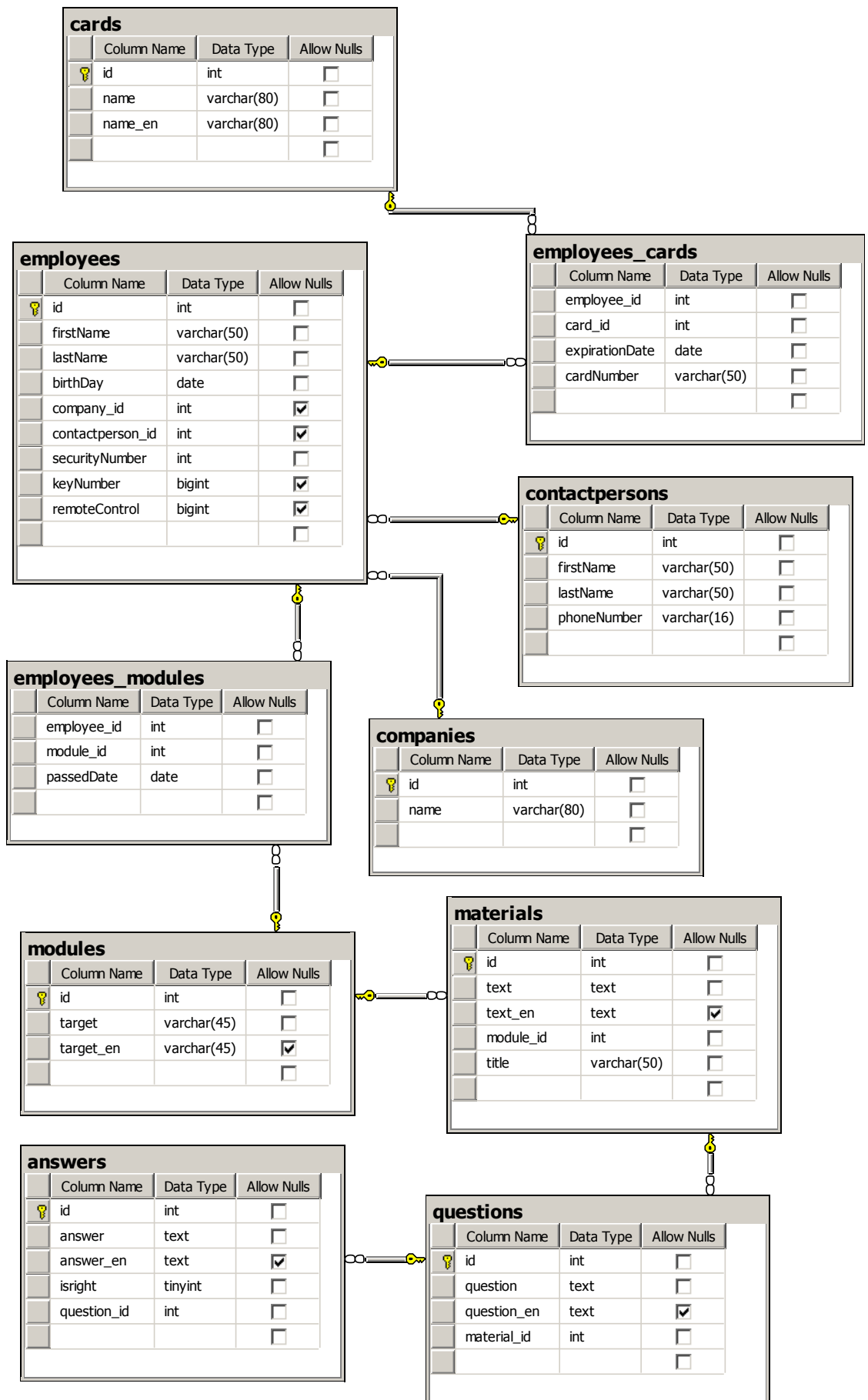
6.3.1 Tietokanta

Perehdytyssovelluksen tietojen lähteeksi valikoitiin sille asetettujen vaatimusten pohjalta Microsoft SQL Serverillä sijaitseva SQL-tietokanta. Se valittiin datalähteeksi ennen kaikkea SQL-taulujen relaatioiden vuoksi, joita SharePointin omilla listoilla olisi ollut haastavampi toteuttaa. Todettiin, että sovellukseen suunnitellut datarakenteet oli-

vat toteutettavissa suotuisammin SQL-tietokannoilla ja lisäksi asetetut vaatimukset myötäilivät tietokannan käyttöä. Tämän myötä sovelluksessa käsiteltävän datan rakenteesta oltiin täysin tietoisia ja sitä voitiin yhdistellä eri tavoin. Yksi esimerkki tästä on käyttäjien tietojen tulostamiseksi tehty näkymä, johon on haettu tietoa useammasta eri taulusta ja lisäksi yhdistelty sitä järkevästi tiivistettyyn raporttimuotoon. SQL-tietokannan käyttö tiesi sitä, että työkalut datan hallitsemiseksi oli itse ohjelmoitava. Sovelluskehittäjien aiemmasta kokemuksesta saadut tiedot ja taidot helpottivat työkalujen rakentamista ja tämä ohjelmistoprojekti vielä vahvisti entisestään jo hankittua osaamista. Sovellus käyttää SQL-autentikointia kirjautuessaan tietokantapalvelimelle.

Tietokannan sisältämä data sidottiin lomake-elementteihin ADO.NET-tekniikan keinoin. Sen sisältämien luokkien avulla muodostettiin yhteys tietokantaan ja suoritettiin SQL-kyselyitä, joiden palauttamista tietokannan riveistä muodostettiin `DataSet`-objekteja. Ne ovat muistiin sidottuja, yhdenmukaisia ja datalähteestä riippumattomia dataesityksiä. Niitä voidaan käyttää useissa erilaisissa datalähteissä samanaikaisesti tietoaineiston hallitsemiseen. `DataSet`-objekti voi sisältää yhden tai useamman `DataTable`-objektin, joka on yksi taulukollinen dataa `DataSet`-objektin sisällä. Se on kokonaisvaltainen tietosäiliö, joka sisältää myös relaatiotaulut, rajoitteet ja relaatiot taulukoiden välillä. (Microsoft MSDN 2012.) `DataSet`-objektien sitominen esimerkiksi alavetovalikoihin, valintalistoihin tai taulukkomuotoisiin `GridView`-kontrolleihin on helppo ja vähän koodirivejä vaativa toimenpide.

SQL-lauseet kirjoitettiin suoraan koodiosaan (*Code-Behind*). Siinä ei hyödynnetty SharePointin Business Connectivity Services -omaisuutta, koska SharePoint Foundation 2010 ei sisällä sen tarjoamaa *Secure Store* -palvelua, joka mahdollistaa kirjautumisen SQL-palvelimelle Windows-autentikoinnin sijasta SQL-autentikoinnin kautta. Suorien SQL-lauseiden käytön myötä ainoastaan haluttu data näytettiin sivuilla kuhunkin tilanteeseen soveltuvalla tavalla. Tietokantayhteys ja SQL-lauseet suojattiin eri tavoin, joista kerrotaan lisää luvussa 6.3.5. Tietokantatauluja muodostui yhteensä 10 kappaletta. Kuva 13 näyttää sovellukseen suunnitellun tietokantarakenteen.



Kuva 13. Danisco Orientation -perhdytyssovelluksen tietokantarakenne

6.3.2 Tietojenhallintatyökalut

Sovelluksen tietojenhallintatyökalut toteutettiin yksinomaan SharePointin web-osien avulla, joita sovellukseen syntyi yhteensä 29 kappaletta. Näiden web-osien tyyppi oli lähes yksinomaan visuaalinen (*Visual Web Part*). Ne eroavat tavallisista web-osista siten, että ne sisältävät toiminnallisuuden (*Code-Behind*) lisäksi ASCX-tiedoston kontrollien ja muun sisällön luomiseksi. Esimerkkeinä kontrolleista mainittakoon tekstikentät, valintalistat ja valintaruudut. Visuaalisia web-osia käytettäessä niitä ei tarvitse luoda koodillisesti. Toki ASCX-tiedostoon on mahdollista kirjoittaa tavallista HTML-koodiakin.

Sovelluksessa esiintyvää dataa käsitellään lähes poikkeuksetta lomake-elementtien välityksellä. Koska tiedon varastona on SQL-tietokanta, oli elementtien luonnista ja ominaisuuksien määrittelyistä huolehdittava itse. ASP.NETin kontrollivalikoimasta löytyi joitakin työtä helpottavia komponentteja. Niistä kenties tärkein oli **GridView**, jonka avulla tietokannasta tulevaa tietoa voidaan tarkastella taulukkomuodossa ja hallita riveittäin. GridViewin **DataSource**- ja **DataBind**-metodeilla ulkoinen data voidaan sitoa kontrolliin. Tämän myötä tietojen muokkaaminen ja poistaminen onnistuu nopeasti ilman erillisiä lomakkeita.

Sovelluksen käytettävyyden parantamiseksi sivuilla hyödynnettiin ASP.NET-kehiksen **UpdatePanel**-kontrolleja. Ne mahdollistavat sivujen lataamisen osittain ja näin ollen sivua ei tarvitse päivittää kokonaan jokaisen PostBack-tapahtuman yhteydessä. Ne ovat siis eräänlainen AJAX-toteutus. Esimerkiksi jos valintalistalta valitaan jokin kyseiseen kontrolliin sidottu arvo, sivun ei tarvitse kokonaan päivittyä, jotta saadaan tieto listalta valitusta arvosta. Kontrolleille on määritettävä tämän aikaan saamiseksi **AutoPostBack**-arvo päälle. Oletuksena **UpdatePanel**-kontrollin sisällä sijaitsevat kontrollit muodostavat tämän tyyppisiä niin sanottuja asynkronisia PostBack-tapahtumia (*AsynchronousPostBack*), mutta **UpdatePanel**ille on mahdollista määritellä myös sen ulkopuolella sijaitsevia kontrolleja, jotka saavat sen sisällön päivittymään. Ne määritellään `<Triggers />`-lohkon sisällä.

UpdatePanelien liiallinen käyttö ei ole suositeltavaa, koska niiden palvelimelle lähettämän datan määrä on suurempi kuin tilanteissa, joissa niitä ei käytetä. Todellisuus-

nessa niiden mukana kulkeutuu palvelimelle sekäPostBack-tapahtuman sisältämä data että sivun tilatieto (*ViewState*). Joissakin ympäristössä tämä voi aiheuttaa ongelmia järjestelmän tehokkuudessa. `UpdatePanel`ien toimintaperiaate onkin aiheellista selvittää hyvin ja niitä on käytettävä säästeliäästi. Niiden tehokkuuden lisäämiseksi on olemassa erilaisia keinoja, esimerkiksi minimoimalla niiden sisältö ja ottamalla niiden automaattinen päivittäminen pois käytöstä (Web.NET 2009).

Perehdytysaineistojen lähteenä oli kaksi vaihtoehtoa: käyttää joko SharePointin dokumenttikirjastoa tai SQL-tietokantaa. Ensin tutustuttiin dokumenttikirjaston mahdollisuuksiin. Tavoitteena oli selvittää, kuinka integroida dokumenttikirjastossa sijaitsevien dokumenttien sijaintitiedot sovellusta varten luodun tietokannan kanssa. Tiedonhaussa kävi ilmi, että suorat kyselyt SharePointin järjestelmätietokantoihin ovat huono ratkaisu. Tähän on syynsä:

- Se on SharePointin EULAn vastaista toimintaa.
- Päivityspakettien asentaminen saattaa rikkoa hakujen toiminnan, sillä järjestelmätietokantojen rakenne voi muuttua.
- Haut saattavat lukita näiden tietokantojen sisältämän informaation ja aiheuttaa ongelmia järjestelmän toiminnassa.
- Haut saattavat palauttaa väärää informaatiota. (El-Hattab 2009.)

Yleisesti ottaen kävi ilmi, että SharePointin luomiin tietokantoihin koskeminen on huono ratkaisu. Niissä sijaitsevan tiedon oikeaoppiseen hakemiseen olisi voitu käyttää palvelinobjektimallia, mutta saatavissa olevan tietoaineiston suppeuden ja käytettävissä olevan ajan rajallisuuden vuoksi päätettiin tässäkin käyttää tavallisia SQL-tietokannan tauluja tietojen varastointiin.

Koska suurin osa web-osista keskusteli tietokannan kanssa ja data sidottiin kontrolleihin, ne käyttivät muutamia kaikille yhteisiä metodeja. Tämän myötä oli loogisempaa sijoittaa ne johonkin keskitettyyn paikkaan, josta niitä voisi kutsua tarpeen mukaan. Sovellukseen luotiin staattinen luokka nimeltään `ExtensionMethods`, joka sisälsi staattiset metodit `DataSetin` luomiselle ja tyhjän `DataSetin` tarkistamiselle. Luokkatieostoja (.cs) ei suoraan sijoiteta osaksi SharePoint-projektia, vaan ne liitetään projektista luodun DLL-tiedoston yhteyteen, joka sijoitetaan Global Assembly Cacheen sovellusta otettaessa käyttöön. Tämä on määritelty SharePoint-

tiedostopakettin `manifest.xml`-tiedostossa, jonka sisältöä käydään tarkemmin läpi luvussa 6.5.

6.3.3 Piirteet

Danisco Orientation -sovelluksen kaikki web-osat jaoteltiin kahdeksi piirteeksi. Toinen sisälsi perehdytysohjelman web-osat ja toinen ylläpitopuolen web-osat. Piirteiden ulottuvuudeksi (*Scope*) asetettiin sivustokokoelma (*Site*), eli ne toimivat ainoastaan sivustokokoelman sisällä. Tämän myötä niitä voidaan hallita sivustokokoelman juuri-sivun asetuksista.

6.3.4 Monikielisyys

Yhtenä kriteerinä Danisco Orientation -sovellukselle oli tehdä siitä sekä suomenkielinen että englanninkielinen, vaikka suurin osa yrityksen työntekijöistä onkin suomalaisia. Monikielisyys oli toteutettava vielä niin, että käännetyt tekstit haetaan erillisistä tiedostoista, jotta niitä olisi mahdollisuus muokata myöhemmin. Näin ollen ne eivät saaneet olla kiinteästi sovellukseen kirjoitettuja.

SharePoint 2010:ssä monikielisyys toteutetaan resurssitiedostoilla (.resx). Niitä on olemassa kaksi eri tyyppiä: sovellusresurssit (*Application Resources*) ja provisiointi-resurssit (*Provisioning Resources*). Sovellusresursseja käytetään normaalin ohjelman suorittamisen yhteydessä kuten Application Page -sivuja, web-osia ja -kontrolleja ajettaessa. Provisiointiresursseja taas käytetään komponenttien provisioinnin yhteydessä eli silloin, kun ne valmistellaan käyttöön otettaviksi. Näitä komponentteja voivat olla esimerkiksi piirteet (*Features*) sekä sivusto- ja listamäärittelyt (*Site And List Definitions*). (Krause ym. 2010, 437.)

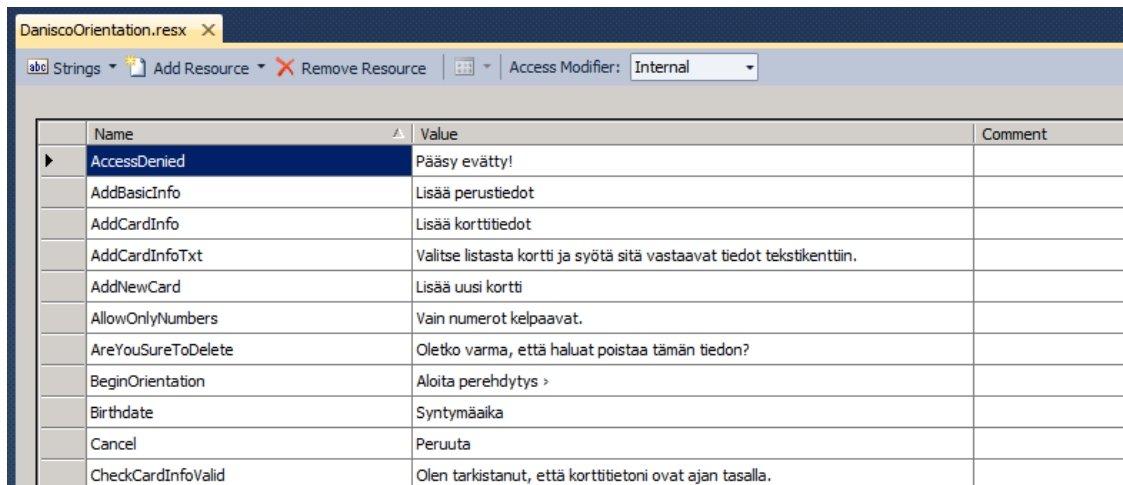
SharePointissa resurssitiedostot sijaitsevat useissa eri hakemistoissa. Niiden sijaintipaikka riippuu resurssin tyypistä. Provisiointiresurssit sijaitsevat joko tiedostopaketin tai piirteiden sisällä seuraavissa hakemistoissa:

- `$_SharePointRoot$\14\TEMPLATE\FEATURES\\Resources\Resources.<lokalisaatio>.resx`

- \$SharePointRoot\$\14\TEMPLATE\FEATURES\- \$SharePointRoot\$\14\Resources\

SharePoint-sovelluksien sovellusresurssit sijaitsevat hakemistossa \$SharePointRoot\$\14\CONFIG\Resources\, josta ne kopioidaan kansioon %SystemDrive%\inetpub\wwwroot\wss\VirtualDirectories\

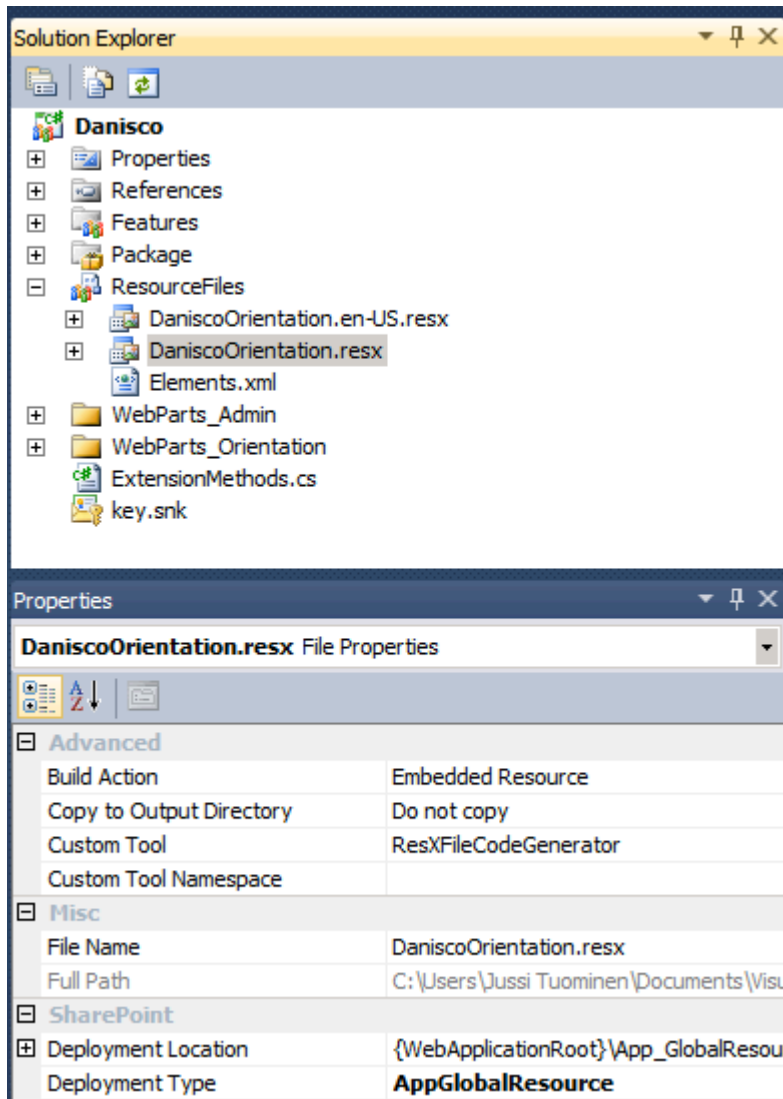
Resurssitiedostot voidaan nimetä joko lokalisoin mukaan tai ilman sitä. Ilman lokalisointia nimetty resurssitiedosto on muotoa `Myresource.resx` kun taas lokalisoin mukaan nimetty resurssitiedosto on muotoa `Myresource.fi-FI.resx`, missä fi-FI on lokalisoin tunnus. Resurssitiedostoja voidaan muokata Visual Studion omalla editorilla tai erillisen XML-editorin avulla. (Krause ym. 2010, 436, 437, 442.) Kuva 14 näyttää Visual Studion resurssieditorin.



Kuva 14. Visual Studio 2010:n muokkaustyökalu resurssitiedostoille

Tässä projektissa resurssitiedostojen nimeämisessä käytettiin lokalisoitua tapaa. Niiden varastoisissa käytettiin hyväksi Visual Studion SharePoint-kehitystyökaluja. Visual Studiassa SharePoint-projektin alle luotiin tyhjä elementti, jonka alle resurssitiedostot sijoitettiin. Tiedostojen ominaisuuksien ”Deployment Type” -kohtaan asetet-

tiin arvoksi ”AppGlobalResource”. Kuva 15 näyttää tämän tilanteen. Tällaisen menettelyn myötä resurssitiedostot kopioidaan aina automaattisesti virtuaalisen hakemiston alla sijaitsevaan App_GlobalResources-kansioon, kun projekti otetaan käyttöön.



Kuva 15. Resurssitiedostojen sijoitus ja ominaisuudet Danisco Orientation-sovelluksessa

ASP.NET-sivulla resurssitiedostoihin voidaan viitata suoraan, kun taas Code-Behind-puolella käytetään `HttpContext`-luokan `GetGlobalResourceObject`-metodia. Koodiesimerkki 1 ja koodiesimerkki 2 esittävät esimerkit resurssitiedostoon määritellyn merkkijonon käytöstä kummassakin tapauksessa:

```
<asp:Button ID="btnIdentifyEmployee" runat="server"
OnClick="identifyEmployee" Text="<% $Resources:DaniscoOrientation,
ConfirmAndContinue %>" />
```

Koodiesimerkki 1. Resurssitiedostossa määritellyn merkkijonon käyttö ASP.NET-sivulla

```
if (DaniscoExtensions.IsEmpty(foundEmployee))
{
    lblError.Text = GetGlobalResourceObject("DaniscoOrientation",
"InvalidConfirmationNumber").ToString();
}
```

Koodiesimerkki 2. Resurssitiedostossa määritellyn merkkijonon käyttö Code-Behindissa

Kuten esimerkistä huomataan, resurssitiedostoon voidaan viitata pelkällä nimellä ilman lokalisaatiotunnusta. Sovelluksen kunkin hetken lokalisaation mukaan järjestelmä tietää, mistä tiedostosta merkkijonon hakee. Koodiesimerkki 3 esittää perehdytysohjelmaan kirjoitetun metodin, jolla sovelluksen kieli asetetaan oletuksena suomeksi, mikäli istuntopuuttujasta ei löydy merkintää kielestä.

```
protected override void FrameworkInitialize()
{
    if (Session["culture"] != null)
    {
        //set new culture
        Thread.CurrentThread.CurrentCulture =
CultureInfo.CreateSpecificCulture(Session["culture"].ToString());
        Thread.CurrentThread.CurrentUICulture = new
CultureInfo(Session["culture"].ToString());
    }
    else
    {
        //default culture is Finnish
        Thread.CurrentThread.CurrentCulture =
CultureInfo.CreateSpecificCulture("fi-FI");
        Thread.CurrentThread.CurrentUICulture = new CultureInfo("fi-
FI");
    }

    //override the base class to change the culture (=language)
    base.FrameworkInitialize();
}
```

Koodiesimerkki 3. Sovelluksen kielen asettamismetodi

6.3.5 Tietoturva

Sovellusta kehitettäessä oli koko ajan pyrkimyksenä tehdä siitä mahdollisimman tietoturvallinen käytettyjä menetelmiä mukaillen. Pääasiallisina kohteina sovelluksen tietoturvan lujittamisessa olivat ”perinteisten” tietoturvauhkien, SQL-injektion ja XSS:n, torjumisen lisäksi ylläpitosivuston suojaaminen.

SQL-injektioiden torjumiseksi tietokantakyselyt suojattiin parametrien avulla. Parametrit ovat muuttujien ”paikanpitäjiä” (*Placeholders*) SQL-lauseissa. Käyttäjien todellisten syötteiden sijasta data kulkeutuu SQL-palvelimelle parametrien välityksellä. Parametrisoidun SQL-kyselyn tarkoituksena on kyky erotella käsiteltävä data suoritettavasta komennosta. Tämän myötä käyttäjän syötteet eivät ole osana suoritettavaa komentoa eikä niiden mahdollisesti haitallinen sisältö tee vahinkoa SQL-palvelimella. (Hurst 2006.) Koodiesimerkki 4 esittää metodin, jossa SQL-lause on suojattu parametrien avulla.

```

private void checkCardCounts()
{
    //define new connection
    conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["sqlConn"].
ConnectionString);

    //use the connection
    using (conn)
    {
        //if available, open it and do stuff
        try
        {
            //sql queries
            string getEmployeesCardsCount = "SELECT COUNT (card_id)
AS Count_EmpCards FROM dbo.employees_cards WHERE employee_id =
(SELECT dbo.employees.id FROM dbo.employees WHERE dbo.employees.id =
@empIdentifier)";

            //sql commands
            Count_EmployeesCards = new
SqlCommand(getEmployeesCardsCount, conn);

            //add parameters to the queries (security reasons)
            Count_EmployeesCards.Parameters.AddWithValue(
"@empIdentifier", Session["employeeIdentifier"]);

            //open connection to database
            conn.Open();

            //build the dataset(s)
            EmployeesCardsCount =
DaniscoExtensions.BuildDataset(EmployeesCardsCount, conn,
Count_EmployeesCards);
        }
        //if there is an error, give an error message
        catch (Exception ex)
        {
            Response.Write(ex.ToString());
        }
    }
}

```

Koodiesimerkki 4. Esimerkkimetodi, jossa suoritettavassa SQL-kyselyssä on käytetty parametria

XSS:n ehkäisemisessä paras tapa on koodata sovelluksessa kulkeutuva mahdollisesti haitallinen data tekstimuotoon, jotta se ei tekisi sovelluksessa tuhojaan. Haitallista koodia voi esiintyä lomake-elementtien lisäksi esimerkiksi sivujen välillä kulkevissa QueryString-merkkijonoissa, istuntonmuuttujissa tai ulkoisesta datalähteestä, kuten SQL-tietokannasta, haetussa tietoaaineistossa. Tämän tyyppinen data voi olla esimerkiksi asiakaspään HTML- tai JavaScript-koodia. ASP.NET tarjoaa valmiin menetel-

män ehkäistä mahdollisesti haitallista dataa nimeltään Request Validation. Se tarkistaa kaikki mahdollisesti haitalliset syötteet ja palauttaa virheilmoituksen, mikäli sellaisen havaitsee. Ominaisuus aktivoidaan web.config-tiedostosta `<system.web />`-lohkoon. Tämän näyttää koodiesimerkki 5.

```
<system.web>
  <pages validateRequest="true" />
</system.web>
```

Koodiesimerkki 5. Request Validation -ominaisuuden kytkeminen päälle sovelluksen määrittelytiedostossa

Oletuksena Request Validation -ominaisuus on päällä, mutta tässä sovelluksessa se kytkettiin pois päältä vaadittavan toiminnallisuuden aikaansaamiseksi. Lisäksi se on kytkettävissä ainoastaan ”kaikki tai ei mitään” -periaatteella, eli sitä ei voi poistaa käytöstä yksittäisissä lomake-elementeissä. Uusimmassa .NET-kehiksen 4.5-versiossa tosin on ominaisuus `ValidateRequestMode`, jolla tämä onnistuu.

XSS:n torjumiseksi on olemassa muitakin mahdollisuuksia. Yksi menetelmä on kontrolloida syötteitä tekstikentissä säännöllisillä lausekkeilla (*Regular Expressions*) joko JavaScriptin tai ASP.NETin `RegularExpressionValidator`-luokan avulla. Niiden avulla estetään haitallisen tekstin kirjoittaminen ja voidaan määritellä vain ne merkit, jotka syötteiksi kelpaavat. Joissain tapauksissa säännöllisten lausekkeiden käyttö on suotavaa, mutta liiallinen käyttäjien syötteiden rajoittaminen saattaa tuoda pitkällä aikavälillä ongelmia. Tässä sovelluksessa otettiin periaatteeksi se, että käyttäjien manuaaliset syötteet jätettäisiin mahdollisimman vähäisiksi käyttämällä tekstikenttien sijasta valintalistoja aina kun mahdollista. Säännöllisiä lausekkeitä käytettiin syöttökentissä, joissa sallittiin vain numerot. Myös syötteiden pituuksia rajoitettiin niissä tekstikentissä, joissa maksimipituus tiedettiin varmuudella. Pakollisten kenttien asettamisessa käytettiin ASP.NETin `RequiredFieldValidator`-luokkaa.

Toinen menetelmä XSS:n torjuntaan, jota hyödynnettiin myös tässä ohjelmistoprojektissa, on käyttää Microsoftin tarjoamaa AntiXSS-kirjastoa. Se on erillinen, Microsoftin verkkosivuilta ilmaiseksi ladattavissa oleva komponentti ASP.NET- ja SharePoint-sovelluksiin. AntiXSS-kirjasto tarjoaa kattavan määrän metodeja ehkäisemään haitallista dataa kulkeutumasta verkkosovelluksen sisällä. AntiXSS-kirjaston metodit eroa-

vat järjestelmän `HttpUtility`-luokasta löytyvistä datan koodausmetodeista siinä, että ne käyttävät kirjoitusmerkeistä niin sanottua sallittujen listan (*Whitelist*) tekniikkaa ei-sallittujen (*Blacklist*) sijasta. Lisäksi AntiXSS-kirjaston `HtmlEncode`-luokka on hieman järjestelmän vastaavaa luokkaa nopeampi. (Basha 2009.) Koska AntiXSS-kirjasto on ulkopuolinen rakennuspalikka, se tuli lisätä osaksi SharePoint-ratkaisua Visual Studion avulla.

Tietokantayhteyden muodostamiseksi tarvittava `ConnectionString`-merkkijono sijoitettiin sovelluksen määrittystiedoston (`web.config`) `<connectionStrings />`-lohkoon, missä se vielä salattiin ASP.NETin salausten menetelmän, RSA-salausalgoritmin (`RSAProtectedConfigurationProvider`), avulla. Tätä varten sovellukselle tuli lisätä avainpalveluntarjoaja. Tämän esittää koodiesimerkki 6.

```
<!-- Luodaan web.config-tiedostoon avainpalveluntarjoaja -->
<configProtectedData>
  <providers>
    <add keyContainerName="DaniscoOrientation" name="Danisco"
type="System.Configuration.RsaProtectedConfigurationProvider,
System.Configuration, Version=2.0.0.0,Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL"
useMachineContainer="true" />
  </providers>
</configProtectedData>
```

Koodiesimerkki 6. Avainpalveluntarjoajan määrittäminen sovelluksen määrittystiedostoon ConnectionString-merkkijonon RSA-salausta varten

Koodiesimerkki 7 esittää komennot, jotka suoritettiin järjestelmänvalvojan oikeuksin Visual Studion komentokehoteessa ASP.NET IIS Registration -työkalun komentojen avulla. Kyseinen työkalu löytyy hakemistosta `%SystemRoot%\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe`.

```
// Luodaan uusi RSA-avainpari nimeltään DaniscoOrientation (pc =
public key, exp = avain on vietävissä)
aspnet_regiis -pc "DaniscoOrientation" -exp

// Tunteettomille (anonymous) käyttäjille (NT AUTHORITY\IUSR)
annetaan lupa käyttää RSA-avainta konetasoisesti
aspnet_regiis -pa "DaniscoOrientation" "NT AUTHORITY\IUSR"

// Salaa <connectionStrings> -osan Danisco-nimisen sovelluksen
Web.config-tiedostosta käyttäen avaimentarjoajaa nimeltä Danisco
aspnet_regiis -pe "connectionStrings" -app "/" -site "Danisco" -prov
"Danisco"

// ConnectionStringistä voidaan poistaa salaus seuraavalla komennolla
aspnet_regiis -pd "connectionStrings" -app "/" -site "Danisco"
```

Koodiesimerkki 7. Komentokehoteessa tehtävät määrytykset ConnectionString-merkkijonon salaamiseksi sovelluksen määrittäytiedostoon (web.config)

Sovellus toteutettiin farmisovellustyyppisenä hiekkalaatikkosovelluksen sijasta. Tähän ratkaisuun päädyttiin ainakin seuraavien rajoitusten pohjalta:

- Visuaalisten web-osien käyttö hiekkalaatikkosovelluksessa ei ole mahdollista muutoin kuin erillisen Visual Studio 2010 SharePoint Power Tools -lisäosan avulla.
- Hiekkalaatikkosovelluksissa ei ole tukea ADO.NET-teknologialle, joten suora tietokannan lukeminen ja sinne kirjoittaminen ei onnistu.
- Hiekkalaatikkosovellukset eivät tue palvelinpuolen uudelleenohjauksia (kuten `Response.Redirect`, `Server.Transfer` ja `SPUtility.Redirect`).
- Resurssitiedostojen (.resx) käyttö on rajoitettu hiekkalaatikkosovelluksissa. (Khan 2013.)

Käyttäjätietojen muokkaus ja itse perehdytys ovat ohjelmassa monivaiheisia prosesseja, joiden aikana siirrytään sivulta toiselle. Tämä herätti kysymyksen siitä, haluaisiko käyttäjä kokeilla onneaan ja siirtyä tietyille sivulle kirjoittamalla suoran osoitteen osoiteriville. Tällainen menettely oli syytä ottaa huomioon, ja sovellukseen lisättiinkin metodi, joka tarkistaa, mitä kautta käyttäjä sivulle on navigoinut. Mikäli tulopaikka ei täsmää metodiin määriteltyjen ehtojen kanssa, käyttäjä siirretään ”Pääsy estetty”-sivulle. Koodiesimerkki 8 näyttää esimerkin metodista, joka päästää sivulle ainoastaan, jos käyttäjä on saapunut sivulle ”Begin.aspx”-sivulta.

```

private void restrictAccess()
{
    //retrieve variables from the referrer
    string employeeId = Request["empId"];
    string firstName = Request["firstName"];
    string lastName = Request["lastName"];
    string clickedLink = Request["lnk"];

    //if referrer page is not null
    if (Request.UrlReferrer != null)
    {
        string[] arr = Request.UrlReferrer.ToString().Split('/');

        //redirect only if we come from the particular page
        (Begin.aspx)
        if (arr[arr.Length - 1].Equals("Begin.aspx"))
            Response.Redirect("ConfirmUser.aspx?empId=" + employeeId
+ "&firstName=" + firstName + "&lastName=" + lastName + "&lnk=" +
clickedLink);
        }
        else
            //if we try direct url access, deny it
            Response.Redirect("AccessDenied.aspx");
    }
}

```

Koodiesimerkki 8. Metodi, joka estää suoran pääsyn tietyille sivulle URL-osoitteen perusteella

6.3.6 Istuntojen käsittely

Sovellus käyttää toiminnassaan istuntomuuttujia (*Session Variables*). Ne ovat palvelimella sijaitsevia objekteja, jotka luodaan `SessionStateItemCollection`-kokoelmaan. Ne ovat saatavilla `HttpContext`-luokan `Session`-ominaisuuden kautta. Istuntomuuttuja luodaan nimen mukaan, ja se voi olla mitä tietotyyppiä tahansa. Istuntomuuttujat ovat keksejä (*Cookies*) luotettavampia, koska niihin tallennettu tieto ei säily istunnon päätyttyä tallessa. Lisäksi käyttäjä ei pysty manipuloimaan niitä, koska ne sijaitsevat palvelimella. Tässä sovelluksessa istuntomuuttujiin tallennetaan käyttäjän valitsema kieli, hänelle luotu yksilöllinen ID-tunnus ja perehdytyksessä valitut moduulit, jotka käyttäjä aikoo suorittaa. ID-tunnus ja valitut moduulit kulkevat koko perehdytyksen suorituksen ajan muistissa. Ne poistetaan perehdytyksen päätyttyä tai selaimen sulkemisen yhteydessä. Perehdytyssovelluksessa asetettiin yhden istunnon pituudeksi oletuksena olevan 20 minuutin sijasta 60 minuuttia, joka riittää yhden käyttäjän perehdyttämiseen.

Jotta istuntomuuttujia voitiin hyödyntää sovelluksessa, ASP.NET Session State -palvelun oli oltava päällä. Oletuksena SharePoint-järjestelmässä näin ei ole. Palvelun saamiseksi käyttöön, sovelluksen määrittiedoston pitäisi sisältää määrittelyt, jotka esittää koodiesimerkki 9.

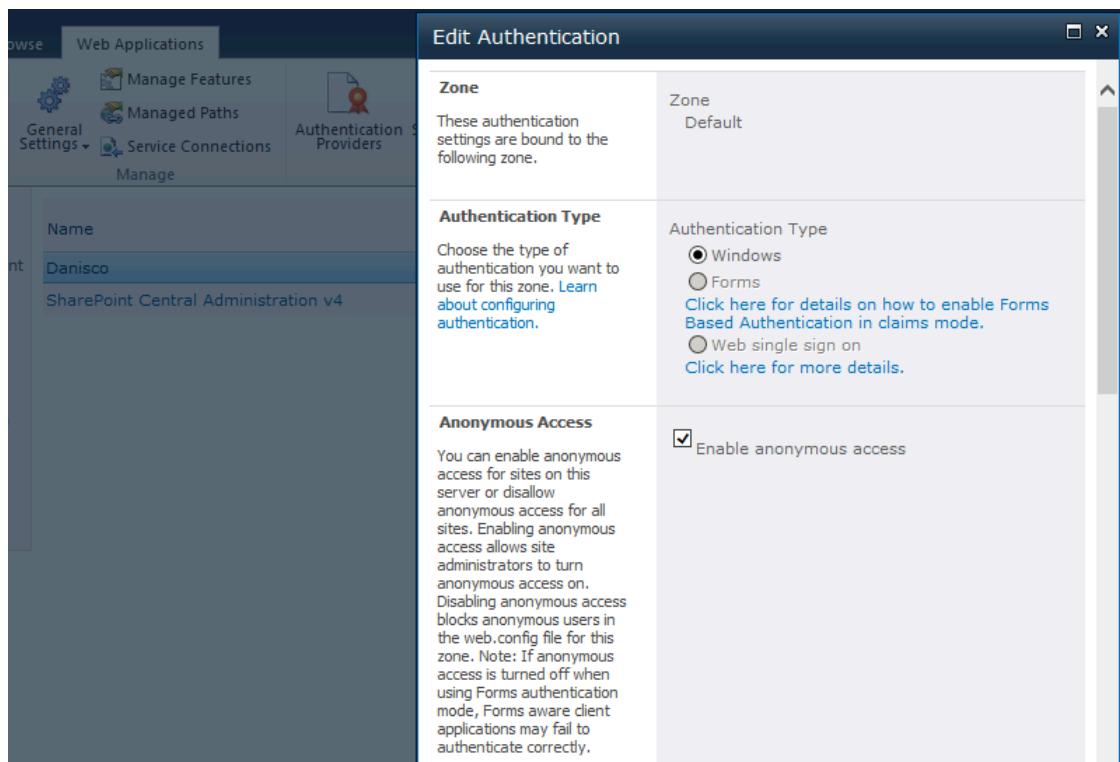
```
<configuration>
  <system.web>
    <sessionState mode="InProc" timeout="60" />
    <pages enableSessionState="true" />
    <httpModules>
      <add name="Session"
type="System.Web.SessionState.SessionStateModule" />
    </httpModules>
  </system.web>
</configuration>
```

Koodiesimerkki 9. Istuntojen kytkeminen päälle sovelluksen määrittiedostosta

Istuntomuuttujat varaavat palvelimelta itselleen riittävän tilan. Tiedon tallentaminen istuntomuuttujaan vie näin ollen aina palvelimen resursseja. Tämän myötä niiden yltiöpäinen käyttäminen ei ole suositeltavaa, vaan on hyvä miettiä hieman tarkemmin, missä tilanteissa niitä on aiheellista käyttää. Pahimmassa tapauksessa ne voivat aiheuttaa palvelinliikenteen häiriöitä ja tehdä koodista vaikeasti hallittavaa, mikäli niiden käyttöä ei osaa kontrolloida.

6.3.7 Käyttöoikeudet

Danisco Orientation -sovellus käyttää autentikointimenetelmänä Windows-autentikointia ja siinä NTLM-protokollaa. Perehdytysivustolla otettiin käyttöön Anonymous Access -menetelmä. Sovelluksen vaatimukseen kuului se, että sen tulee toimia työasemissa User-oikeuksilla ja että se olisi kaikkien tehtaassa työskentelevien henkilöiden käytössä. Anonymous Access -menetelmän myötä sivustolle pääsemiseksi ei tarvitse erikseen kirjautua, vaan se on kaikkien työasemalle kirjautuneiden käyttäjien tavoitettavissa. Näin ollen perehdytyksen voi suorittaa tehtaalla esimerkiksi jonkin yleisen tilan työasemalla. Ominaisuus tulee voimaan koko sivustokokoelman piirissä, ja se on otettava erikseen käyttöön tietyllä sivustolla keskitetyn hallinnan kautta. Kuva 16 näyttää tämän tilanteen.



Kuva 16. Anonymous Access -ominaisuuden kytkeminen keskitetyn hallinnan kautta

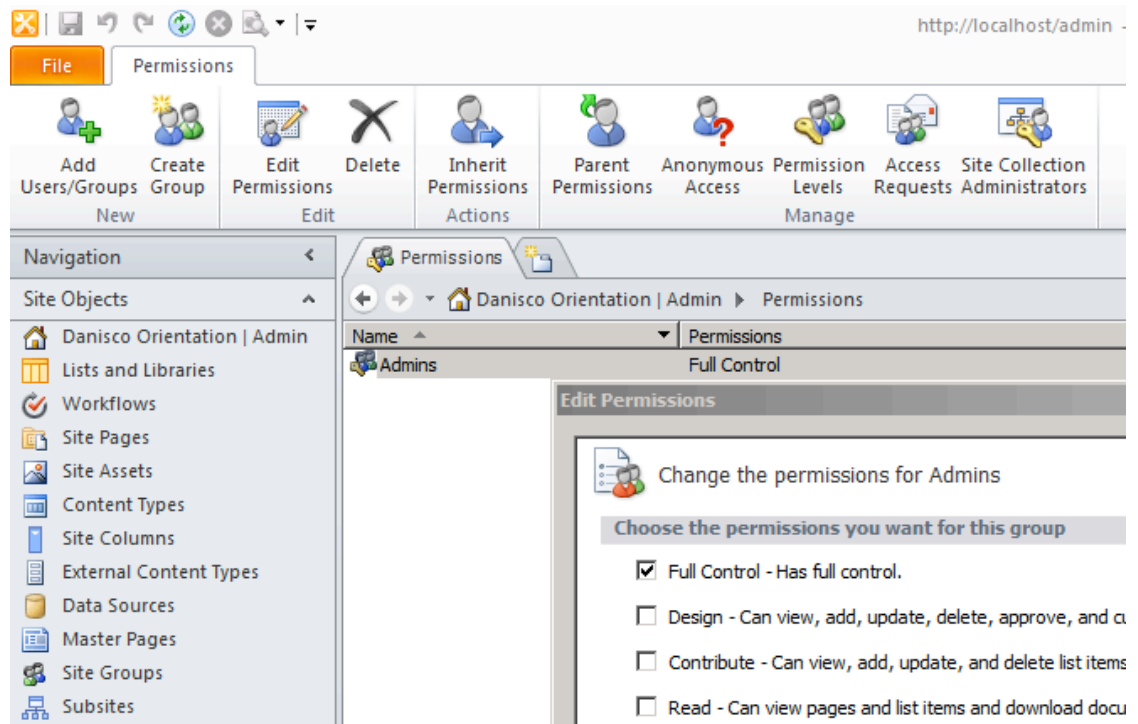
Tuntemattomilta käyttäjiltä poistettiin lisäksi näkyvistä yläreunan Ribbon-palkki. Tämä tapahtui sijoittamalla perustyyllisivussa palkin luomiskoodi `<SharePoint:SPSecurityTrimmedControl />`-lohkon sisälle ja antamalla `PermissionsString`-ominaisuuden arvoksi jokin sellainen oikeus, mitä tuntemattomalla käyttäjällä ei ole. Koodiesimerkki 10 näyttää tämän.

```
<SharePoint:SPSecurityTrimmedControl
PermissionsString="AddAndCustomizePages" runat="server">
<!-- show the ribbon here -->
</SharePoint:SPSecurityTrimmedControl>
```

Koodiesimerkki 10. Ribbon-palkin piilottaminen tuntemattomilta (Anonymous) käyttäjiltä

Hallintasivuston ollessa alisivusto (*Subsite*) sivuston käyttöoikeudet periyetään oletuksena tälle sivustolle. Hallintaan ei kuitenkaan kaikilla käyttäjillä ollut asiaa, ja näin ylimmän tason sivuston oikeuksien periytyminen katkaistiin. Tämän myötä hallintaoisioon voitiin määritellä omia käyttäjien oikeusmäärittelyjä ja luoda erityisiä käyttäjäryhmiä. Hallintaosiota varten luotiin uusi käyttäjäryhmä, ”Admins”, johon ainoastaan sovelluksen ylläpitäjät sijoitettiin. Alisivuston oikeusasetukset ja uusi käyttäjäryhmä

luotiin SharePoint Designer 2010 -ohjelmaa käyttäen. Kuva 17 näyttää kuvankaappauksen tästä tilanteesta.



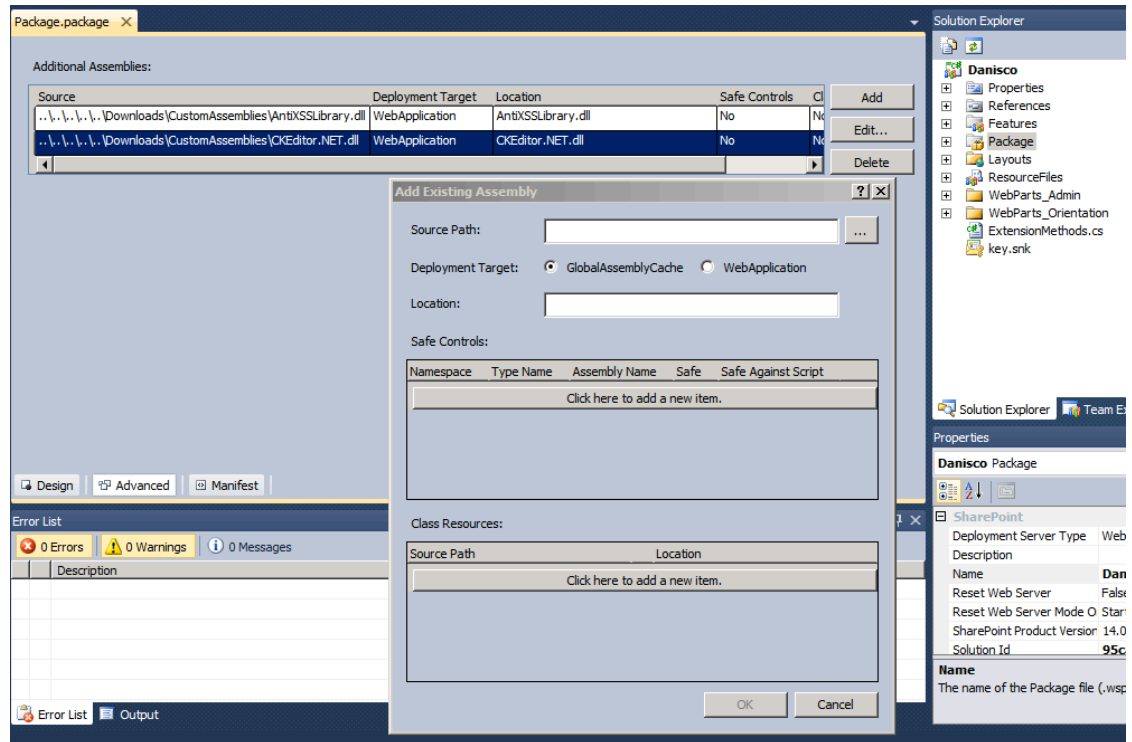
Kuva 17. Käyttäjä- ja käyttäjäryhmätyökalut SharePoint 2010 Designer -ohjelmassa

6.3.8 Mukautetut komponentit

Mukautettuja komponentteja (*Custom Assemblies*) ovat kolmannen osapuolen kokoaamat sovellusosat. Komponenttien tiedostopäätte on .dll, ja niitä voidaan liittää eri tavoin osaksi SharePoint-sovellusta. Visual Studio 2010 tarjoaa kaksi tapaa sijoittaa tämäntyyppinen komponentti, jotta sitä voitaisiin käyttää sovelluksessa. Niitä ovat GAC (*Global Assembly Cache*) ja WebApplication.

Global Assembly Cachessa sijaitsevat komponentit ajetaan täysin oikeuksin. Ne sijoitetaan hakemistoon %SystemRoot%\assembly. GAC:een sijoitetut komponentit ajetaan ilman Code Access Security (CAS) -mallia. Komponentit voidaan sijoittaa ja ajaa myös sovelluksen virtuaalisen hakemiston bin-kansiosta, jolloin niiden oikeudet ovat rajoitetut. Tällöin komponenttien sijoittamismäärityksiin on asetettu WebApplication-valinta. Komponentit otetaan käyttöön Visual Studion avulla ja samalla niille valitaan sijoituspaikka. Tämä onnistuu tiedostopaketin lisäasetusten takaa löytyvistä työkaluista. Kun projekti otetaan käyttöön, komponentit siirtyvät samalla valinnan

mukaisiin paikkoihin. Kuva 18 näyttää tilanteen, jossa projektiin ollaan lisäämässä uutta komponenttia.



Kuva 18. Uuden komponentin lisääminen SharePoint-projektiin Visual Studio 2010:lla

SharePoint Foundation 2010:ssä on oletuksena kolme suojaustasoa (trust level):

`WSS_Minimal`, `WSS_Medium` ja `WSS_UserCode`, jotka perustuvat ASP.NETin sisäänrakennettuihin suojaustasoihin. Suojaustasojen määrittely sijaitsee SharePointin asennushakemiston CONFIG-kansiossa

(`$SharePointRoot$\14\CONFIG`) löytyvistä konfiguraatiodoista.

`WSS_Minimal` on oletuksena käytössä. (Microsoft MSDN 2010.) Yksittäisille kolmannen osapuolen komponenteille voi antaa muokattuja oikeuksia. Tämä tapahtuu ottamalla kopio jostakin edellä mainitusta konfiguraatiodoista ja luomalla siitä uusi tiedosto omien sääntöjen määrittelyä varten. Tämän jälkeen sovelluksen määrittämistiedostossa on määritettävä sovellus käyttämään mukautettua konfiguraatiodoista niin kuin koodiesimerkki 11 näyttää.

```
<trustLevel name="WSS_Custom" policyFile="C:\Program Files\Common
Files\Microsoft Shared\Web Server
Extensions\14\config\wss_custom.config" />
<trust level="WSS_Custom" originUrl="" />
```

Koodiesimerkki 11. Oman suojaustasomäärittelytiedoston määrittelemineen sovelluksen määrittelytiedostossa

6.3.9 Erikoisominaisuudet

Perehdytysaineistojen datalähteeksi valittiin muun informaation tavoin SQL-taulu, kuten aiemmin raportissa on mainittu. Aineistojen sisältöjen toivottiin muistuttavan Word- tai PDF-dokumentteja tekstimuotoiluineen ja kuvineen. Tässä kohtaa päädyttiin käyttämään rikastekstieditoria nimeltään CKEditor, jonka kotisivut löytyvät osoitteesta <http://ckeditor.com>. Siitä on kehitetty myös juuri ASP.NETille tarkoitettu versio ja CKEditorin kotisivuilta löytyikin selkeät ohjeet editorin liittämiseksi osaksi sovellusta. Oli kuitenkin huomattava antaa CKEditor-komponentille riittävät oikeudet, jotka toteutettiin edellisessä kappaleessa mainitulla tavalla. Omaan suojaustasomäärittelytiedostoon lisättiin lohko, jonka esittää koodiesimerkki 12. Se asettaa kaikki tämän SharePoint-sovelluksen virtuaalisen hakemiston bin-kansiosta löytyvät komponentit ajettaviksi täysin oikeuksin. Näin ollen niitä ei tarvitse lisätä Global Assembly Cacheen.

```
<CodeGroup class="UnionCodeGroup" version="1"
PermissionSetName="FullTrust">
  <IMembershipCondition class="UrlMembershipCondition"
version="1" Url="$AppDirUrl$/bin/*" />
</CodeGroup>
```

Koodiesimerkki 12. Luottamuksen asettaminen tietyssä kansiossa (tässä sovelluksen virtuaalisen hakemiston bin-kansio) sijaitseville sovelluskomponenteille omissa suojaustasomäärittelytiedostossa

CKEditor tukee oletuksena ainoastaan kuvan lisäämistä suoran osoitteen perusteella internet-sivulta. Kuvien selaus- ja lataustoiminnot ovat mahdollisia lisäominaisuuksia, jotka tulee lisätä CKEditorin määrittelyyn. Nämä asetukset ovat `filebrowserBrowseUrl` ja `filebrowserUploadUrl`. Tässä sovelluksessa perehdytysaineistojen kuvien hallinnan toteuttaminen yritettiin tehdä mahdollisimman

suoraviivaiseksi. Perehdytysaineistoihin lisättävät kuvat sijoitettiin dokumenttikirjastoon, mistä niitä oli tarkoitus pystyä selaamaan ja poistamaan. Myös uuden kuvan lataus sovellukseen mahdollistettiin. Kuvan latauksen toteuttamista varten sovellukseen tuli lisätä web-käsittelijä, jonka tiedostopääte on muotoa .ashx. Tällaisella tiedostopäätteellä varustetut tiedostot on oletuksena estetty SharePointissa, joten ne tuli sallia tietyssä sovelluksessa keskitetystä hallinnasta. Lisäksi lohko, jonka koodiesimerkki 13 esittää, piti lisätä SharePoint-projektitiedostoon, jotta järjestelmä kykenee tunnistamaan tiedostotyyppin ja etsimään siitä niin kutsutun korvattavan parametrin tai tunnisteen (*Replaceable Parameter*). Oletuksena Visual Studio suorittaa tämän toimenpiteen SharePoint-projekteissa vain seuraavista tiedostopäätteistä: .xml, .ascx, .aspx, .webpart ja .dwp. (Malik 2012.)

```
<PropertyGroup>
<TokenReplacementFileExtensions>ashx</TokenReplacementFileExtensions>
</PropertyGroup>
```

Koodiesimerkki 13. ASHX-tiedostopäätteen lisääminen SharePoint-projektitiedostoon

Web-käsittelijät ovat dynaamisia tiedostoja, jotka toimivat sellaisenaan niitä kutsuttaessa. Jotta kuvan lataus onnistuisi, oli rakennettava web-käsittelijä, joka tunnistaa käyttäjän lataaman tiedoston kuvaksi, lisää sen ennalta määriteltyyn dokumenttikirjastoon ja keskustelelee CKEditorin kanssa siten, että lisätyn kuvan polku lisätään automaattisesti kuvan lisäysikkunaan. Näin ollen sitä pystyy esikatselemaan kyseisessä ikkunassa. Kuvan lisäämiseen on lisäksi tehty sellainen ominaisuus, että mikäli tietyllä nimellä (tässä käytettiin nimeä *Material Images*) olevaa dokumenttikirjastoa ei vielä ole olemassa, se luodaan uuden kuvan lisäämisen yhteydessä automaattisesti. Mikäli samanniminen kuva on jo kirjastossa, se ylikirjoitetaan.

Muut ominaisuudet, jotka CKEditorissa tuli olla mukana, olivat SharePointin dokumenttikirjastossa olemassa olevien kuvien selaus, niiden lisääminen materiaaliin ja kuvien poistaminen. CKEditorin asetusmääryksien muokkaamisen jälkeen kuvan lisäämisikkunaan ilmestyi ”Selaa palvelinta” -painike. Sille annettiin osoitteeksi dokumenttikirjaston sivun osoite, jonka se avasi uudessa ikkunassa. Jotta CKEditor tietäisi, mikä kuva kirjastosta on valittu, oli sivulle lisättävä lisätoiminnallisuutta. Apu löytyi asiakaspään objektimallista ja ECMAScriptistä, joka on aiheena raportin luvussa 5.9. Sen avulla päästiin kiinni dokumenttikirjastosta valittuun kuvaan. Tämän jälkeen sen

polku voitiin palauttaa CKEditorille. Kuvan poistamiseen kirjastosta käytettiin SharePointin listanäkymän omaa Poista-toimintoa, joka voitiin tehdä tässä samassa ikkunassa. Tarkempi kuvaus näiden toimenpiteiden suorittamisesta käytännössä on luetussa liitteessä olevasta käyttöohjeesta.

Perehdyttämisen päättyessä käyttäjälle tulostetaan kaksipuoleinen käyntikorttikokoinen todiste suoritetuista perehdytysmoduuleista. Todisteessa näkyvät myös sen viimeinen voimassaolopäivä ja käyttäjän yksilöllinen tunnistenumero, jota hänen on käytettävä tullessaan seuraavan kerran suorittamaan perehdytystä. Todisteesta muodostetaan PDF-tiedosto avoimen lähdekoodin komponentilla nimeltään iTextSharp. Se on C#-muunnos alun perin Java-ympäristöön kehitetystä iText-PDF-kirjastosta. PDF-tiedoston tekeminen on perusteltu toimenpide, koska tällöin voidaan todisteesta tehdä kooltaan ja ulkoasultaan juuri sellainen, kuin on tarkoituskin. HTML- ja CSS-tekniikoiden käyttö todisteen muodostuksessa ei ollut tämän vuoksi hyvä valinta.

6.4 Testaus

Sovellusta testattiin koko sen kehityksen ajan aktiivisesti. Pienikin muutos koodissa täytyi testata, mikä lisäsi sovelluskehitykseen kuluvaan aikaan merkittävästi. Aktiivisella testauksella oli syynsä: sovelluksesta pyrittiin tekemään mahdollisimman virheetön ja lisäksi kehitysympäristön ollessa kehittäjille aiemmin tuntematon, se oli välttämätöntä. Tosin tietopohjan karttuessa aivan jokaista pientä koodimuutosta ei ollut tarpeen testata.

Oletuksena SharePoint näyttää virheilmoitukset ilman yksityiskohtaisia tietoja. Tämä oli kehityksen kannalta huono asia, sillä tämän myötä virheiden jäljittäminen oli vaikeampaa. Virheilmoitukset yksityiskohtaisine tietoineen voidaan laittaa päälle kirjoittamalla sovelluksen määrittelytiedoston `<system.web />`-lohkoon asetuksen `<customErrors mode="Off" />`.

Sovelluskehittäjien lisäksi myös yrityksen työntekijät testasivat perehdytysohjelmaa. Tämä niin kutsuttu testauttaminen oli oleellinen tapa selvittää sovelluksen virheettömyys lopullisessa ympäristössään. Ulkopuolinen käyttäjä, jolle sovellus on aivan uusi, saattaa löytää ohjelmasta vielä lisää epäkohtia. Nämä epäkohdat olisivat saattaneet jäädä huomaamatta ilman testauttamista, sillä kehittäjän on mahdotonta asettua loppu-

käyttäjän asemaan. Sovellukselle oli määrä järjestää ennen sen varsinaista tuotanto-käyttöä usean viikon testausjakso tuotantoympäristössä, jossa kartoitettiin sovelluksen käytettävyyttä ja toimintaa.

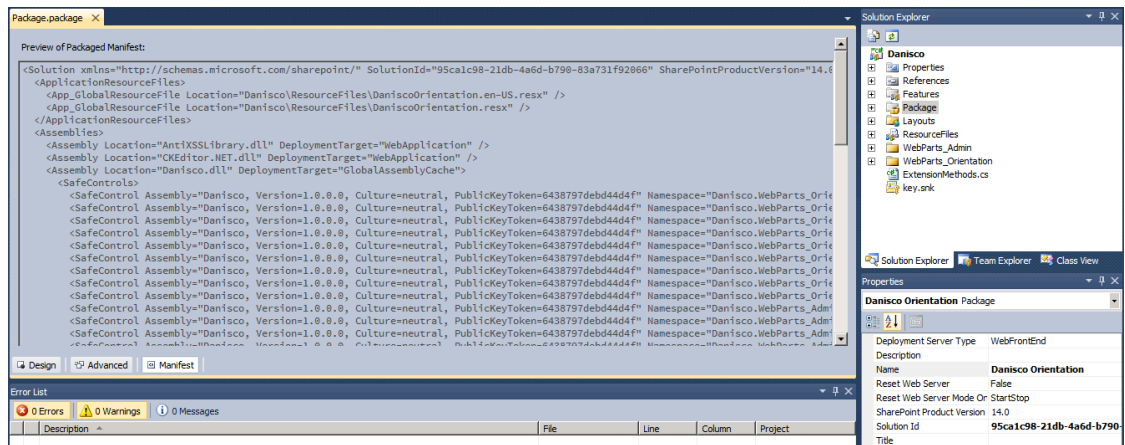
6.5 Siirtokuntoon saattaminen

Kun perehdytyssovellus oli siinä pisteessä, että kaikki tarvittavat ominaisuudet oli toteutettu ja testattu, oli aika alkaa suunnitella sen asentamista tuotantoympäristöön.

Tämä prosessi ei ollut itsestään selvä toimenpide, vaan se piti suunnitella huolellisesti ja tehdä riittävää taustaselvittelyä sen läpiviennin onnistumiseksi. Internet oli myös tässä työn vaiheessa suurin apuväline.

SharePoint-sovellukseen - tai suoraan suomennettuna SharePoint-ratkaisuun (*SharePoint Solution*) - tehtyjen ominaisuuksien asettamiseksi vientikuntoon oli niistä tehtävä tiedostopaketti (*Windows Solution Package, WSP*), jonka tiedostopääte on .wsp. Se on tiivistetty ja käytännössä CAB-muotoinen tiedosto, joka sisältää kaikki SharePoint-projektiin luodut komponentit ja piirteet. Niitä se voi sisältää yhden tai useamman. Nämä piirteet on erikseen aktivoitava kohdeympäristössä, ennen kuin niitä voidaan käyttää. Tiedostopaketti sisältää tietoa muun muassa web-osien turvallisuustasoista. Vain turvallisiksi luokiteltuja komponentteja voidaan käyttää SharePoint-sovelluksessa. Turvalliset komponentit määritellään tiedostopaketin manifest.xml-tiedostossa ja ne rekisteröidään web.config-tiedostoon `<SafeControl />`-lohkoihin. (Wilén. 2011, 172, 178.)

WSP-paketti sisältää XML-muotoisen manifest-tiedoston, joka määrittelee paketin sisällön. Tätä tiedostoa on mahdollista muokata myös manuaalisesti Visual Studion SharePoint-tiedostopaketin lisäasetuksissa. Tiedostossa määritellään myös, mihin sovelluskomponentit on määrä sijoittaa: Global Assembly Cacheen vai web-sovelluksen omaan bin-kansioon. (Wilén. 2011, 172 - 173.) Kuva 19 esittää näkymän manifest.xml-tiedoston esikatseluikkunasta.



Kuva 19. SharePoint-projektin tiedostopaketin manifest.xml-tiedoston esikatse-
lunäkymä

SharePoint-projektista voidaan luoda helppoiten tiedostopaketti Visual Studiolla. Tämä tapahtui klikkaamalla projektin nimeä ja valitsemalla ”Package”. Tämän jälkeen paketti oli saatavilla projektin bin-hakemistosta. Paketin sisältöä voi halutessaan tarkastella muuttamalla sen tiedostopääte .cab:ksi ja purkamalla tiedostot tavallisella purkuohjelmalla.

Testiympäristössä tiedostopaketin käyttöönotto (*Deployment*) on automatisoitu Visual Studioon kautta, mutta tuotantoympäristön palvelimella Visual Studiota ei hyvin todennäköisesti ole käytettävissä. Niinpä käyttöönotto on hoidettava skriptien avulla joko PowerShell-komentotulkilla tai *stsadm*-komentoilla. SharePoint 2010 -versiossa on suositelluinta käyttää PowerShellä. SharePointin asennuksen yhteydessä asennettulla SharePoint 2010 Management Shellillä voidaan toteuttaa järjestelmään liittyvät hallinnolliset toimet. Työkalun voi käynnistää Käynnistä-valikon SharePoint-kansiosta löytyvästä pikakuvakkeesta. Toinen mahdollisuus on käyttää Windowsin omaa PowerShell-työkalua, johon on lisätty tuki SharePoint-komennoille, niin sanoituille *cmdleteille*. (Wilén. 2011, 180 - 181.)

Tietokannan varmuuskopiointi onnistui parhaiten SQL Management Studiolla. Sivustokokoelman siirtämiseksi toiseen ympäristöön oli parasta käyttää SharePointin varmuuskopiointi- ja palautuskomentoja. Käytännössä kehitysfarmille tehdystä sivustokokoelmasta otetaan varmuuskopio ja tämä tiedosto palautetaan kohdeympäristön SharePoint-farmilla. Nämä toimenpiteet tulee suorittaa niin ikään PowerShell-komentotulkilla. Ne on mahdollista toteuttaa myös keskitetyn hallinnan kautta, mutta suurin onnistumisen todennäköisyys on käytettäessä komentoriviä. Varmuuskopion

ottaminen tapahtui komennolla `Backup-SPSite`, jonka näyttää koodiesimerkki 14. Komennolle kirjoitettiin parametreiksi varmuuskopioitavan sivustokokoelman osoite ja hakemisto, minne varmuuskopiotiedosto haluttiin tallennettavan.

```
Backup-SPSite -Identity "http://sivustokokoelma:portti" -Path
C:\Backup\sivusto.bak
```

Koodiesimerkki 14. Sivustokokoelman varmuuskopiointikomento

6.6 Käyttöönotto

Ennen varsinaiseen tuotantoympäristöön asentamista sovellukselle tehtiin testiasennus koulun tarjoamalle testauskäyttöön pystytetylle palvelimelle. Asennuksessa oli huomioitava se, että kohdeympäristön build-numeron tuli olla yhtä suuri kuin kehitysympäristössä. Esimerkiksi tämän projektin SharePoint-kehitysympäristön build-numero oli asennusta aloitettaessa 14.0.7015.1000, joka tarkoittaa SharePoint 2010:n Service Pack 2 -päivitystä. Myös kaikki päivitys- ja kielipaketit, jotka kehitysfarmin SharePointiin oli asennettu, oli asennettava myös kohdefarmin SharePointiin. Testiasennus tapahtui sen jälkeen, kun sovelluksen katsottiin olevan ominaisuuksiltaan valmis testi-käyttöä varten. Perehdytyssovelluksen käyttöön ottaminen sisälsi viisi päävaihetta:

1. Mixed Mode -autentikoinnin (sisältää sekä Windows- että SQL-autentikoinnin) käyttöönotto ja uuden käyttäjän luominen tietokantapalvelimelle riittävin oikeuksin. Toinen vaihtoehto olisi käyttää SQL-palvelimen pääkäyttäjän (*System Administrator*), kirjautumistietoja. Tämän jälkeen sovelluksen käyttämä tietokanta palautetaan kehitysympäristössä tehdystä varmuuskopiosta kohdeympäristön SQL-palvelimelle.
2. Sivustokokoelman palauttaminen kehitysympäristössä tehdystä varmuuskopiosta kohdeympäristön SharePoint-farmille.
3. Tiedostopakettien lisääminen kohdeympäristön SharePoint-farmille ja asentaminen palautetulle sivustokokoelmalle.
4. Tiedostopakettien sisältämien piirteiden aktivoiminen.
5. Mahdolliset lisätoimenpiteet sovelluksen virheettömän toiminnan aikaan saamiseksi.

Lopuksi sivustonkokoelman sisällä sijaitsevalle hallinta-alisivustolle määritettiin käyttäjät ja käyttöoikeudet ja ohjelmaa testattiin kaikilta osa-alueilta perusteellisesti. Mahdolliset epäkohdat kirjattiin ylös ja selvitettiin niiden aiheuttajat. Koska perehdytyssovellus toteutettiin farmisovelluksena, sen asentamiseksi tarvittiin farmin ylläpitäjän oikeudet.

Kohdeympäristön SharePoint-farmissa luotiin uusi web-sovellus ja sivustokokoelma. Varmuuskopio palautettiin tähän sivustokokoelmaan komennolla, jonka esittää koodiesimerkki 15. Parametri **-Force** ylikirjoittaa olemassa olevan sivustokokoelman.

```
Restore-SPSite -Identity "http://sivustokokoelma:portti" -Path
C:\koko\polku\varmuuskopioon.bak -Force
```

Koodiesimerkki 15. Sivustokokoelman palauttamiskomento tuotantoympäristön SharePoint-farmille

SharePoint-projektiin luotu tiedostopaketti otettiin seuraavaksi käyttöön SharePoint-farmilla ja tämän jälkeen asennettiin sinne. Se lisättiin SharePoint-farmille komennolla, jonka koodiesimerkki 16 esittää.

```
Add-SPSolution -LiteralPath "C:\koko\polku\tiedostopakettiin.wsp"
```

Koodiesimerkki 16. SharePoint-tiedostopaketin lisääminen SharePoint-farmille

Paketti asennettiin tietyn sivustokokoelman käyttöön komennolla **Install-SPSolution**. Esimerkin tästä esittää koodiesimerkki 17.

```
Install-SPSolution -Identity tiedostopaketti.wsp -WebApplication
http://sivustokokoelma:portti -GACDeployment
```

Koodiesimerkki 17. Tiedostopaketin asentaminen SharePoint-sivustokokoelmalle

Asennuskomennon parametreina ovat SharePoint-farmille lisätyn tiedostopaketin nimi ja sivustokokoelma, jolle paketti asennetaan. Parametri **-GACDeployment** asettaa paketin Global Assembly Cacheen, eli se saa ”täyden luottamuksen”. Tiedostopaketin poistamiseksi on olemassa luonnollisesti oma komentonsa, **Uninstall-SPSolution**. Käyttöön otetun tiedostopaketin saa myös päivitettyä komennolla

Update-SPSolution. Se on tarpeellinen komento silloin, kun paketin sisältämiin komponentteihin on tehty koodimuutoksia ja vanha versio halutaan ylikirjoittaa uudemmalla.

Tiedostopakettiin sisällytetyt piirteet eivät aktivoidu automaattisesti toiseen ympäristöön asennettaessa, vaan ne tulee ottaa käyttöön manuaalisesti joko keskitetystä hallinnasta tai komentoriviltä **Enable-SPFeature**-komennolla. Piirteen poistamiseksi käytöstä käytettäisiin komentoa **Disable-SPFeature**.

7 SOVELLUKSEN TOTEUTTAMINEN

Tässä projektissa, kuten ohjelmistokehityksessä yleisestikin, ohjelmistokehittäjillä oli vapaus päättää, millaisessa ympäristössä ja millaisin menetelmin he tuotetta vievät eteenpäin.

7.1 Kehitysympäristö

Sovellusta kehitettiin lähes yksinomaan Kymenlaakson ammattikorkeakoulun GameLabin tiloissa. GameLab on peliohjelmoinnin oppimisympäristö Kyamkin tietotekniikan osastossa. Sen tiloissa on saatavilla nykyaikaisia työkaluja pelien ja ohjelmistojen kehittämiseen eri alustoille. GameLab on tarkoitettu opiskelijoille, joiden intohimona on ohjelmointi ja erityisesti pelien tekeminen. Vaikka tilat soveltuvat ehkä kaikkein parhaiten pelien tekemiseen, myös web-sovellusten ohjelmoimiseen ne sopivat todella hyvin.

GameLabin tilat olivat kummallekin projektiin osallistuneelle ohjelmistokehittäjälle entuudestaan hyvin tutut, ja oli luonnollista työskennellä siellä opinnäytetyön ohjaajan valvonnassa. Näin hän voi myös olla koko ajan tietoinen siitä, että työ edistyy. Ohjaajan oli myös mahdollista nähdä työn alla olevaa materiaalia.

7.2 Kehitystyökalut

Opinnäytetyössä toteutetun SharePoint-sovelluksen rakentamiseen käytettiin perinteisiä Microsoftin tarjoamia sovelluskehitysohjelmistoja. SharePoint 2010 -versioista käytössä oli SharePoint Foundation.

7.2.1 Microsoft Visual Studio 2010

Microsoft Visual Studio on Microsoftin kehittämä ohjelmistokehitysympäristö (*Integrated Development Environment, IDE*). Siitä on saatavilla Ultimate-, Premium-, Professional- ja Test Professional -versiot. Ohjelmiston hinta kasvaa luonnollisesti sen sisältämien ominaisuuksien mukaan. Myös ilmaisia Express-versioita on saatavissa yksittäiselle alustalle tai ohjelmointikielille. Visual Studiolla voidaan kehittää muun muassa työpöytä-, mobiili-, konsoli- ja Internet-sovelluksia ja siihen voidaan integroida myös kolmannen osapuolen lisäosia. Sisäänrakennettu ohjelmointikielituki ohjelmassa on Visual Basic .NET:lle, C/C++:lle, C#:lle ja F#:lle. Lisäksi ohjelman tukeisiin kieliin kuuluvat myös XML/XSLT, HTML/XHTML, JavaScript ja CSS. (Atif Nadeem 2012.)

Visual Studio 2010 julkaistiin 12. huhtikuuta 2010. Se sisältää lukuisia ohjelmistokehitystä helpottavia ominaisuuksia, joista esimerkkeinä voidaan mainita kooditäydennysominaisuus IntelliSense, vastaavuuksien automaattinen korostaminen koodissa sekä haku- ja siirtymistoiminnot koodin sisällä (Daily Coding 2010). Se toi myös paremman käyttökokemuksen SharePoint-kehitykseen edeltäjiinsä verrattuna muun muassa uusien SharePoint-projektityyppien ja SharePoint-palvelimen selaustyökalun (*Server Explorer*) muodossa (CodersBarn 2009).

7.2.2 Microsoft SQL Server 2008 R2

Microsoft SQL Server 2008 on Microsoftin kehittämä tietokannanhallintaohjelmisto (*Database Management System, DBMS*), joka sisältää tietokantamoottorin lisäksi paljon muitakin työkaluja. Niitä ovat esimerkiksi SQL-palvelimen hallinta- ja muokkaustyökalut SQL Server Management Studio, SQL Server Configuration Manager ja SQL Server Agent, tietokantojen synkronointityökalut, tietokantojen peilausominaisuudet sekä erilaiset palvelut. SQL Server 2008 (R2) on julkaistu monina erilaisina versioina, joita ovat Standard-, Enterprise-, Workgroup-, Developer-, Web- ja Express Edition. (Rankins ym. 2010, 10 - 28.)

Express Edition, jota käytettiin tässä projektissa, on ilmainen, sulautettavissa ja uudelleen jaettavissa oleva versio SQL Server 2008:sta. Se on suunnattu käyttäjille, joiden kehittämät ohjelmat vaativat toimiakseen paikallisesti asennetun tietokannan. Express-

version tietokantamoottori rakentuu samalle pohjalle kuin muissakin versioissa. Se on ladattavissa Microsoftin verkkosivuilta. (Rankins ym. 2010, 28.)

Perehdytyssovelluksen käyttämää tietokantaa työstettiin SQL Server Management Studio -ohjelmalla. SQL Server Management Studio (SSMS) on graafinen työkalu Microsoft SQL Serverin tietokantojen hallintaan, joka julkaistiin ensimmäistä kertaa vuonna 2005. Työkalu sisältyy Microsoft SQL Serverin asennukseen. Ilmaiseen Express-versioon on olemassa SQL Server Management Studio Express -versio. (Rankins ym. 2010, 28, 63.) Sitä käytettiin tässä projektissa.

SQL Server Management Studiolla hoituvat kaikki hallinnolliset toimet Microsoftin SQL-palvelimella. Sillä voidaan työskennellä usealla palvelimella samanaikaisesti. SQL Server Management Studio sisältää tietokantaskriptimuokkaimia, joilla koko tietokantaa voidaan hallita nopeasti erilaisilla skripteillä. Ohjelma sisältää useita graafisia työkaluja, joilla työstetään tietokantapalvelimen komponentteja. Keskeisimpiä ominaisuuksia ohjelmassa on Object Explorer, jonka avulla hallinnoidaan palvelimella sijaitsevia tietokantaobjekteja. Tietokantoja voi luoda lisää, muokata jo olemassa olevien tietokantojen sisältöjä ja analysoida tietokannan tehokkuutta. Myös graafinen näkymä kyselyiden tekemiseen löytyy ohjelmasta. (Halvorsen 2012, 9.)

7.2.3 Microsoft SharePoint Designer 2010

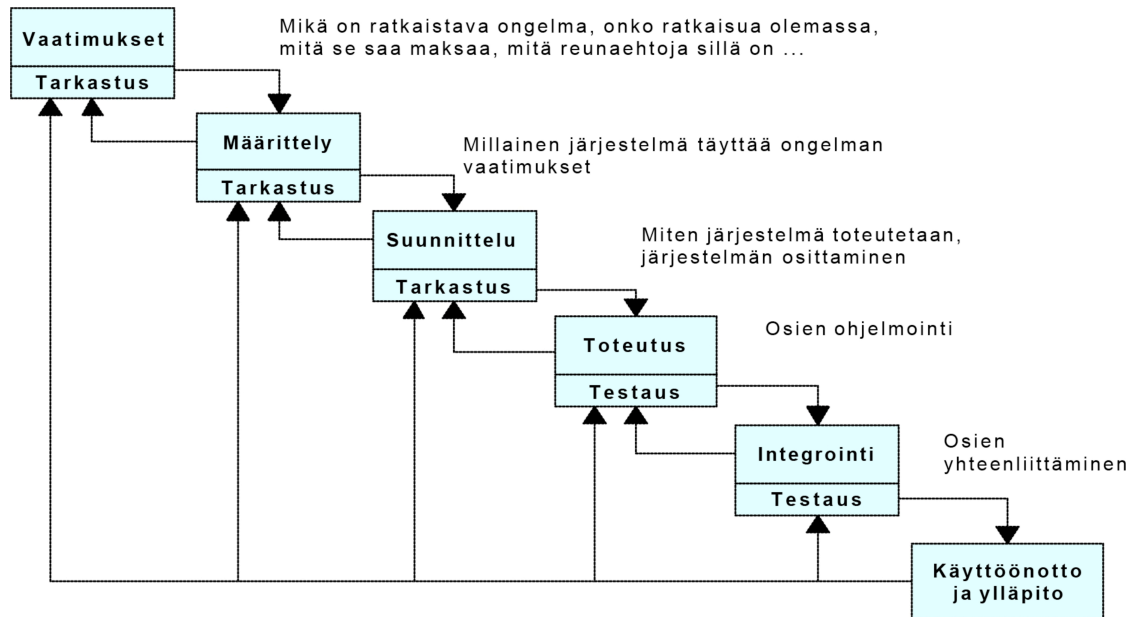
Microsoft SharePoint Designer on ilmainen, visuaalinen työkalu, jonka tarkoitus on helpottaa SharePoint-sivustojen hallintaa. Sen avulla hoituvat kaikki sivustoa koskevat perustoiminnot, mutta myös hieman kehittyneemmätkin toiminnot ilman ohjelmointitaitoja. SharePoint Designer mahdollistaa muun muassa sivujen luonnin, ulkoasun muokkaamisen, käyttäjien ja käyttöoikeuksien konfiguroinnin, listojen, dokumenttikirjastojen ja työnkulkujen hallinnan, sivujen varmuuskopioimisen sekä sellaisten sivujen muokkaustoimenpiteiden suorittamisen, joka ei selaimen kautta ole mahdollista. (Pattison ym. 2010, 19 - 20.) Yksityiskohtaisempien toimintojen toteuttaminen tai muokkaaminen vaatii kuitenkin aina ohjelmointiosaamista ja Visual Studion käyttöä.

7.3 Kehityksen elinkaari

Perehdytysohjelman toteutuksessa edettiin lineaarisesti, ja sen kehityksen elinkaari muistutti eniten vesiputousmallia. Tosin alkuperäistä vesiputousmallia kehitys ei täydellisesti noudattanut. Perehdytyssovelluksen elinkaarimalli voidaankin todeta olevan muunneltu vesiputousmalli (*Modified Waterfall Model*). Tämä elinkaarimalli on kehitetty parantamaan alkuperäisen vesiputousmallin heikkouksia. Suurin ero perinteiseen vesiputousmalliin verrattuna on vaiheiden päällekkäisyyden mahdollisuus: eri tasojen sisältämät kokonaisuudet voivat tapahtua ohjelmistoprojektissa samaan aikaan. Koska sovelluskehitys ei ole kerrallaan sidottu ainoastaan yhteen asiaan kerrallaan, kehitys on joustavampaa. Tällainen kehitys voi nopeuttaa tuotteen valmistumista. Huonona puolena tällä elinkaarimallilla on se, että se voi tehdä eri vaiheista häilyvämpiä ja näin ollen ohjelmistokehityksestä voi olla vaikeampi erottaa selviä kehitysvaiheita. Lisäksi eri vaiheiden väliset riippuvuussuhteet voivat kasvaa suuremmiksi ja aiheuttaa aikaa vievää siirtymistä eri vaiheiden välillä. (Satakar 2010.)

Muunneltu, iteroiva vesiputousmalli kuvasi tämän ohjelmistoprojektin kehitystä parhaiten. Kehitysympäristö oli heti selvillä ja loppukäyttäjä oli alusta alkaen tietoinen, mitä kehitettävän sovelluksen tulee ainakin sisältää. Lisäksi sovelluksen määrittelemisen toteutettiin läheisesti asiakasyrityksen kanssa. Sovellukseen tuli sen elinkaaren aikana pieniä lisävaatimuksia ja -tarkennuksia, jotka tuli ottaa huomioon kehityksessä. Tämä aiheutti sen, että joitakin ohjelmiston elinkaaren vaiheita täytyi toistaa. Toteutuksen ja testausvaiheen välillä siirtyminen eri kehitysvaiheiden välillä oli kaikkein intensiivisintä.

Jokainen ohjelmistokehityksen taso sisälsi käytettävien menetelmien ja työkalujen varmistamisen (*Verification And Validation*), jolla pyrittiin varmistamaan ohjelmiston kehityksen suunnan oikeellisuus ennen seuraavalle tasolle siirtymistä (Nummenmaa 2008, 1). Tuli olla mahdollisimman varma siitä, että käytetyt menetelmät ovat perusteltuja. Asiakasyritykseltä vastaanotettiin palautetta kehityksen alla olevasta tuotteesta jokaisen vaiheen aikana. Tämä edesauttoi sitä, että loppukäyttäjä tulisi saamaan vaatimuksiaan vastaavan sovelluksen elinkaaren lopussa. Kuva 20 esittää perehdytyssovelluksen elinkaaren periaatekuvan.



Kuva 20. Perehdytyssovelluksen kehityksen elinkaarimalli (Aaltonen 2008, 1.)

8 YHTEENVETO JA POHDINTA

Tämän opinnäytetyön tarkoituksena oli tutustua Microsoft SharePoint 2010 -järjestelmään ja sen ominaisuuksiin. Työn käytännön osuudessa rakennettiin SharePointista hankittujen perustietojen pohjalta kyseiselle alustalle käytännön sovellus yrityskäyttöön. Kyseessä oli sähköinen perehdytysohjelma Kotkassa sijaitsevalle Danisco Sweeteners Oy:lle, jonka tuotantoympäristönä oli yrityksen intranet. Sovellus rakennettiin toimeksiantajan toiveiden mukaisesti parityönä omilla laitteistoilla ja oman osaamisen pohjalta. Sovelluskehitys alkoi kehitysympäristön pystytyksestä ja jatkui sen käytännön kehittämisen kautta lopulta asennusprosessin läpikäymiseen.

Opinnäytetyön vaatima tietomäärä oli valtava, ja opin sen aikana merkittävästi uusia asioita sekä SharePoint-järjestelmästä että .NET-sovelluskehityksestä. Työtä aloitettaessa jopa hieman pelotti ajatella, kuinka paljon tietoa oli hankittava suhteellisen lyhyessä ajassa, jotta tilattu sovellus saataisiin kunnialla valmiiksi. Sukellus SharePointin maailmaan oli pikainen, eikä matkan varrellakaan pitkille ”hengähdystauoille” jäänyt aikaa. Tämän myötä opinnäytetyön laajuus vielä korostui: se on toteutuksessa vaaditun tietomäärän suhteen laajin ohjelmistoprojekti, johon olen tähän asti osallistunut.

Perimmäisenä tavoitteena oli saada aikaan toimiva ohjelma. Tämä tavoite toteutui täydellisesti, ja jo siitä on syytä olla tyytyväinen. Projektin edetessä ja samalla ympäristön tullessa tutummaksi työskentely muuttui rutiininomaisemmaksi ja matka kohti

onnistunutta lopputulosta alkoi näyttää selvemmältä. Opinnäytetyö todisti minulle sen, että ihminen kykenee omaksumaan suuren määrän tietoa lyhyessä ajassa, kunhan sitä on mahdollisuus soveltaa käytäntöön. SharePointin ollessa kyseessä yhtä oikeaa menettelytapaa ei aina ole ja näin ollen selvitystyön tulee olla perusteellista. On haettava perusteita sille, miksi juuri tätä kyseistä menetelmää kannattaa tai ei kannata käyttää tietyissä tilanteissa.

SharePoint on mammuttimainen järjestelmä, ja tässä ohjelmistoprojektissa sen sisältämistä mahdollisuuksista opittiin vain murto-osa. Kuitenkin järjestelmän perustoinnallisuuden sisäistäminen ja arvokkaan kokemuspohjan hankkiminen ovat positiivisia asioita ja on perusteltua olla kiitollinen, kun sain osallistua tämän tyyppiseen sovelluskehitykseen. SharePoint-tietämys ei varmasti ole hukkaan heitettyä tietoutta IT-maailmassa. Se on haastava opittava, sillä voi olla vaikeaa päättää, miltä kannalta opiskelu kannattaa aloittaa. Internet antaa kuitenkin tarvittavaa taustatukea, ja sen rooli oli tässä työssä keskeinen. Yleisesti ottaen toisen asian selvittäessä myös siihen sidoksissa oleviin asioihin muodostuu perusta, josta niitä on helpompi lähteä lähestymään.

Kokonaisuutena olen työhön tyytyväinen. Oma osaamiseni kehittyi koko ajan, ja käytössä olleet resurssit olivat suurilta osin täysimittaisessa käytössä. Koko projektin etenemisen ajan oli suhtauduttava kriittisesti käytettyihin menetelmiin ja mietittävä huolellisesti, ollaanko kehityksessä hakoteilla vai oikeilla urilla. Työskentelyssä korostui myös se, että ohjelmistoprojekteissa kehittäjien ja asiakkaan vuorovaikutus on elintärkeää, jotta asiakas saa mieleisensä tuotteen. Kehitystyö sisälsi myös paljon hankaluuksia, mutta se kuului asiaan. Projektin haasteellisuus oli kuitenkin myös eduksi, ja tämä tietyllä tavalla pakottikin viemään työtä eteenpäin. Vaikka haastetta riitti jokaisessa projektin vaiheessa, motivaationi ei horjunut. Kaiken kaikkiaan olen omaan työskentelyyni tyytyväinen ja mielestäni sovellukselle antamani panostus oli riittävä. En olisi paljon enempää tässä ajassa pystynyt projektille antamaan.

Jatkokehitettäviä asioita sovellukseen voisi ajatella ainakin muutamia. Niistä mainittakoon tiiviimmän Microsoft Office -integraation toteuttaminen ja SharePoint-listojen ja -dokumenttikirjastojen parempi hyödyntäminen sovelluksessa. Perehdytysaineistojen sijoituspaikkana dokumenttikirjasto voisi olla vartenotettava ratkaisu tietokantaulujen sijasta. Tällöin niiden suunnitteleminen tutuilla Microsoftin Officen ohjelmistoilla voisi olla käyttäjäystävällisempi ratkaisu kokemattomille käyttäjille. Lisäksi eri-

laisten raporttien kehittyneemmät luomistoiminnot ja raporttien viemistoiminnot esimerkiksi juuri Microsoft Officen ohjelmiin, kuten Exceliin, voisivat tulla kysymykseen. Tässä sovelluksen ensimmäisessä versiossa on vielä tietyillä osa-alueilla hieman perinteisempi ote. Lisäksi ASP.NET-kehityksen laajaan kontrollivalikoimaan voisi puurtua vielä syvällisemmin ja valita sieltä kaikkiin tilanteisiin parhaiten soveltuvat kontrollit ja tehdä sovelluksesta tässä suhteessa optimoidun kokonaisuuden. Lisäksi SharePointin objektimalleja voisi mahdollisesti hyödyntää sovelluksessa vielä enemmän. Mikäli yrityksessä ilmenee tarvetta sovelluksen jatkokehitykselle, näistä palasista muodostuu varmasti kattava opinnäytetyön aihe tulevaisuudessa.

Tärkeintä tässä opinnäytetyössä toteutetussa sovelluskehityksessä oli kuitenkin saada sekä ennalta esitetyt että projektin aikana toivotut ominaisuudet rakennettua ohjelmaan. Tässä tavoitteessa onnistuttiin hyvin. Koska lopputuote sisältää vaaditut ominaisuudet ja se toimii halutulla tavalla, työn voidaan todeta olevan onnistunut ja opinnäytetyölle asetetut tavoitteet voidaan katsoa täytetyiksi. Opintojeni alkuvaiheessa en ajatellut uraa SharePoint-sovelluskehittäjänä, mutta tämän opinnäytetyön myötä järjestelmän lukuisat mahdollisuudet alkoivat kiehtoa minua ja mahdollinen ura SharePointin parissa on syytä pitää harkinnassa.

LÄHTEET

Aaltonen, K. 22.4.2008. Ohjelmointitekniikka lyhyesti: Elinkaarimalleista. s. 1. Saatavissa: <http://users.metropolia.fi/~karita/2010-3ITProg/Introduction/OhjelmistotekniikkaLyhyesti.pdf> [viitattu 3.9.2013].

Alikoski, A. 9.2.2006. Microsoft .NET -arkkitehtuuri. Saatavissa: http://www.soberit.hut.fi/T-76.5650/Spring_2006/lectures/Dotnet%20arkkitehtuuri.ppt [viitattu 16.9.2013].

ASP.NET Tips for Projects. 04/2013. .NET Framework Architecture. Saatavissa: <http://www.dotnettips4u.com/2013/04/net-framework-architecture.html> [viitattu 16.9.2013].

Atif Nadeem, M. 11.9.2012. Visual Studio Overview. Saatavissa: <http://matifnadeem.blogspot.fi/2012/09/visual-studio-overview.html> [viitattu 29.8.2013].

Bamboo Solutions Corporation. 15.11.2010. SharePoint 2010 User Management. s. 3–4, 18. Saatavissa: http://community.bamboosolutions.com/cfs-file.ashx/___key/CommunityServer.Components.PostAttachments/00.00.03.61.32/SharePoint-2010-User-Management.pdf [viitattu 9.9.2013].

Bandarupalli, K. 20.10.2012. Understanding the ASP.NET Programming Models. Saatavissa: <http://www.techbubbles.com/aspnet/understanding-the-asp-net-programming-models/> [viitattu 18.9.2013].

Basha, S.A., 9.7.2009. Differences Between AntiXss.HtmlEncode and HttpUtility.HtmlEncode Methods. Saatavissa: <http://blogs.msdn.com/b/securitytools/archive/2009/07/09/differences-between-antixss-htmleencode-and-httputility-htmleencode-methods.aspx> [viitattu 7.8.2013].

Bloomberg. 10.1.2011. Taloussanomat. Amerikkalainen DuPont ostaa Daniscon. Saatavissa: <http://www.taloussanomat.fi/yriytykset/2011/01/10/amerikkalainen-dupont-ostaa-daniscon/2011315/12> [viitattu 9.7.2013].

Bordner, S. 19.4.2011. Does SharePoint Foundation 2010 have BCS? Saatavissa: <http://mossseo.wordpress.com/2011/04/19/does-sharepoint-foundaton-2010-have-bcs/> [viitattu 22.7.2013].

CodersBarn. 2009. Microsoft Visual Studio 2010 Overview. Saatavissa: http://www.codersbarn.com/file.axd?file=2009%2F5%2FVisualStudio2010_ProductOverview.pdf [viitattu 29.8.2013].

Daily Coding. 14.4.2010. Top 5 new features of Visual Studio 2010 and .NET 4. Saatavissa: http://www.dailycoding.com/Posts/top_5_new_features_of_visual_studio_2010_and_net_4.aspx [viitattu 29.8.2013].

Danisco Sweeteners Oy, Kotkan tehdas. 05/2012. s. 9–10. Yhteistyökumppanien koulutus. Koulutusmateriaali.

DuPont. 01/2011. DuPontin toimintaohje. s. 3. Saatavissa: http://www2.dupont.com/Social_Commitment/en_US/assets/downloads/code_of_conduct/DuPont_CoC_Finnish.pdf [viitattu 9.7.2013].

DuPont. 2013. Yritysesite. Saatavissa: <http://us.vocuspr.com/Newsroom/ViewAttachment.aspx?SiteName=DupontEMEA&Entity=PRAsset&AttachmentType=F&EntityID=125302&AttachmentID=4e091bab-9354-4be2-9506-a3b1f09fe3cf> [viitattu 9.7.2013].

El-Hattab, A.M. 1.10.2009. Five Reasons Not to Directly Query SharePoint Databases. Saatavissa: <http://www.codeproject.com/Articles/42787/Five-Reasons-Not-to-Directly-Query-SharePoint-Data> [viitattu 20.7.2013].

Fennia. 2012. Perehdyttäminen ja työnopastus. Saatavissa:

http://www.fennia.fi/servlet/ContentServer?c=Document_C&pagename=Fennia%2FDocument_C%2FDocumentDetail&cid=1352442357748 [viitattu 8.7.2013].

Halvorsen, H. Structured Query Language. 13.4.2012. s. 9. Saatavissa:

<http://home.hit.no/~hansha/documents/database/documents/Structured%20Query%20Language.pdf> [viitattu 1.9.2013].

Hurst, D. 19.4.2006. SQL Injection: The Hacker's Gold Mine. How to prevent your organization from becoming the next target. Saatavissa: <http://sqlmag.com/t-sql/sql-injection-hacker-s-gold-mine> [viitattu 5.8.2013].

Jamison, S., Hanley, S. & Cardarelli, M. 2010. Essential SharePoint 2010: Overview, Governance, and Planning. s. 28. Boston, USA: Addison-Wesley Professional.

Joy, D. 31.12.2010. Difference between Web Application, Site Collection and Site. Saatavissa: <http://www.c-sharpcorner.com/uploadfile/Roji.Joy/difference-between-web-application-site-collection-and-site/> [viitattu 12.9.2013].

Khan, R. 3.4.2013. 50 Limitations of SharePoint Sandboxed Solutions. Saatavissa: <http://reyazuddinkhan2.blogspot.fi/2013/04/50-limitations-of-sharepoint-sandboxed.html> [viitattu 8.8.2013].

Krause, J., Langhirt, C., Sterff, A., Pehlke, B. & Döring, M. 2010. SharePoint 2010 as a Development Platform, s. 134–136, 436–442, 481, 711. New York, USA: Apress.

MacDonald, M., Hamilton, B. 2003. ADO.NET in a Nutshell. s. 3–5, 16. California, USA: O'Reilly Media.

Malik, S. 14.2.2012. SharePoint and Visual Studio - Replaceable parameters. Saatavissa: http://blah.winsmarts.com/2012-2-SharePoint_and_Visual_Studio_-_Replaceable_parameters.aspx [viitattu 28.8.2013].

Maxwell, R. 27.5.2010. Understanding SharePoint 2010 Claims Authentication. Saatavissa: <http://blogs.msdn.com/b/russmax/archive/2010/05/27/understanding-sharepoint-2010-claims-authentication.aspx> [viitattu 14.8.2013].

Microsoft. 3.5.2010. Introducing the .NET Framework 4.0. s. 2 Saatavissa: http://download.microsoft.com/download/0/0/F/00F1D7AD-4018-48F4-A802-2A67C0C62CDD/01_0672331004_ch01.qxp.PDF [viitattu 17.9.2013].

Microsoft MSDN. 05/2010. Securing Web Parts in SharePoint Foundation. Saatavissa: [http://msdn.microsoft.com/en-us/library/cc768613\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/cc768613(v=office.14).aspx) [viitattu 19.8.2013].

Microsoft MSDN. 05/2010. What's New: Business Connectivity Services (BCS) in SharePoint Server 2010. Saatavissa: [http://msdn.microsoft.com/en-us/library/ee557658\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/ee557658(v=office.14).aspx) [viitattu 22.7.2013].

Microsoft MSDN. 02/2011. SharePoint 2010 Architectures Overview. Saatavissa: [http://msdn.microsoft.com/en-us/library/gg552610\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/gg552610(v=office.14).aspx) [viitattu 11.9.2013].

Microsoft MSDN. 2.8.2012. ADO.NET DataSets. Saatavissa: <http://msdn.microsoft.com/en-us/library/zb0sdh0b.aspx> [viitattu 28.8.2013].

Microsoft MSDN. 2013a. Glossary for SharePoint 2010. Saatavissa: <http://msdn.microsoft.com/en-us/library/ee556558.aspx> [viitattu 12.9.2013].

Microsoft MSDN. 2013b. .NET Framework 4. Saatavissa: [http://msdn.microsoft.com/en-us/library/vstudio/w0x726c2\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/w0x726c2(v=vs.100).aspx) [viitattu 16.9.2013].

Microsoft MSDN. 2013c. Using SharePoint Lists vs. Database Tables. Saatavissa: <http://msdn.microsoft.com/en-us/library/ff647105.aspx> [viitattu 18.7.2013].

Moonsoft. 2013. Microsoft SharePoint Server 2010. Saatavissa: <http://www.moonsoft.fi/products/000717.aspx> [viitattu 9.9.2013].

- Naji, A. 2012. SharePoint 2010 Site Structure Plan. Saatavissa: <http://www.slideshare.net/amdnaji/sharepoint-2010-site-and-site-collection> [viitattu 12.9.2013].
- Nasr, S. 26.10.2004. Comparison of C# and VB.Net. Saatavissa: <http://www.clevelanddotnet.info/presentations/CSVNetComparison.ppt> [viitattu 17.9.2013].
- Nummenmaa, J. 2.9.2008. Software Lifecycle Models. s. 1. Saatavissa: http://www.cs.uta.fi/~jyrki/old_courses/se07/slides-pdf/2-lifecycle-models.pdf [viitattu 3.9.2013].
- Parmstig, T. 9.2.2012. Differences Between Sandboxed and Farm Solutions in SharePoint 2010. Saatavissa: <http://blogg.sigma.se/en/Portals-and-Collaboration/Tony-Parmstig/Dates/2012/2/Differences-Between-Sandboxed-and-Farm-Solutions-in-SharePoint-2010/> [viitattu 5.9.2013].
- Pattison, T., Connell, A. & Hillier, S. 2010. Inside Microsoft SharePoint 2010. s. 12–13, 19–20. California, USA: O'Reilly Media.
- Paul, J. 2.5.2010. SharePoint 2010 - SharePoint Object Model. Saatavissa: <http://www.c-sharpcorner.com/UploadFile/40e97e/sharepoint-2010-sharepoint-object-model/> [viitattu 9.9.2013].
- Perran, A., Perran, S., Mason, J. & Rogers, L. 2011. Beginning SharePoint 2010: Building Business Solutions with SharePoint. s. 343, 560. Indianapolis, USA: Wiley Publishing, Inc.
- Rajaram, J. 2006. Asp. Net Interview Questions And Answers. s. 103. New Delhi, India: Firewall Media.
- Rankins, R., Bertucci, P., Gallelli, C. & Silverstein, A.T. 2010. Microsoft SQL Server R2 Unleashed. s. 10–28. Indianapolis, USA: SAMS Publishing.

Rizzo, T., Alirezaei, R., Fried, J., Swider, P., Hillier, S. & Schaefer, K. 2010. Professional SharePoint 2010 Development. s. 142. Indianapolis, USA: Wiley Publishing, Inc.

Sapossnek, M. 21.11.2005. The Common Language Runtime (CLR). Saatavissa: http://www.dcl.hpi.uni-potsdam.de/teaching/componentVI05/slides/Net_VL1_02_CLR.pdf [viitattu 17.9.2013].

Satalkar, B. 19.10.2010. Modified Waterfall Model. Saatavissa: <http://www.buzzle.com/articles/modified-waterfall-model.html> [viitattu 3.9.2013].

Sayed, S., Singh, M. & Santhakumari, V. 2012. Sams Teach Yourself SharePoint 2010 Development in 24 Hours. s. 27, 56–59. Indianapolis, USA: SAMS Publishing.

Seppälä, J. 10.1.2011. Tekniikka & Talous. Dupont ostaa Daniscon. Saatavissa: <http://www.tekniikkatalous.fi/kemia/dupont+ostaa+daniscon/a558651> [viitattu 9.7.2013].

Sivakumar, N. 2007. C++/CLI in Action. s. 4–5. Connecticut, USA: Manning Publications.

Web.NET. 27.4.2009. How to Improve ASP.NET UpdatePanel Performance. Saatavissa: <http://blog.turlov.com/2009/02/how-to-improve-aspnet-updatepanel.html> [viitattu 27.9.2013].

Whitehead, J. 22.1.2008. Introduction to C#. Saatavissa: <http://classes.soe.ucsc.edu/cmsp020/Winter08/lectures/intro-csharp.pdf> [viitattu 17.9.2013].

Wilén, W. 2011. SharePoint 2010 Web Parts in Action. s. 173–181. Connecticut, USA: Manning Publications.

Wright, T.B. 2010. ASP.NET 4 24-Hour Trainer. s. 3–5. Indianapolis, USA: Wiley Publishing, Inc.



Danisco Sweeteners Oy - Kotkan tehdas

DANISCO ORIENTATION

Perehdytysohjelma Microsoft SharePoint 2010 -järjestelmään

Käyttöohje (v1.0)

29.10.2013

SISÄLLYS

1	YLEISTÄ	3
2	SOVELLUKSEN OSAT	3
2.1	Perehdytys sivusto	3
2.1.1	Uusi käyttäjä	5
2.1.2	Palaava käyttäjä	9
2.1.3	Perehdytys	16
2.2	Hallintasivusto	19
2.2.1	Käyttäjiin liittyvän datan hallinta	20
2.2.1.1	Tietojen tarkastelu	20
2.2.1.2	Tietojen muokkaaminen ja poistaminen	22
2.2.2	Perehdytykseen liittyvän datan hallinta	24
2.2.2.1	Tietojen tarkastelu	24
2.2.2.2	Tietojen muokkaaminen ja poistaminen	25
3	SIVUJEN MUOKKAAMINEN	31



1 YLEISTÄ

Danisco Orientation -ohjelma on kehitetty Kotkassa sijaitsevalle Danisco Sweeteners Oy:lle opinnäytetyönä vuonna 2013. Kyseessä on sähköinen perehdytysohjelma, joka rakennettiin perehdytyksen nykyaikaistamiseksi yrityksen tarpeiden pohjalta. Se on kehitetty Microsoft SharePoint 2010 -alustalle ja se käyttää tietojensa varastointiin Microsoft SQL Server 2008 -ympäristössä sijaitsevaa SQL-tietokantaa.

Ohjelman sisältämän koulutuksen käyvät läpi yrityksen yhteistyökumppanien työntekijät. Ohjelmassa on oletuksena kaksi moduulia: urakoitsijat ja vierailijat, mutta uusien moduulien lisääminen on mahdollista. Kukin moduuli suoritetaan samalla tavoin: perehdytysaineisto luetaan ja tämän jälkeen vastataan sitä koskeviin kysymyksiin.

Perehdytysohjelman avulla työntekijät voivat myös tarkastella ja muokata omia henkilökohtaisia tietojaan, jotka sovelluksen rekisteriin on tallennettu. Niitä ovat nimen ja syntymäajan lisäksi yritys- ja yhteyshenkilötiedot sekä työturvallisuuskorttitiedot. Perehdytyksen hyväksytyt suoritus on voimassa yhden (1) vuoden ajan, jonka jälkeen se on uusittava. Hyväksytysti suoritetusta perehdytyksestä tallennetaan tieto ohjelmaan ja sen vanhentumista seurataan ohjelman sisällä.

2 SOVELLUKSEN OSAT

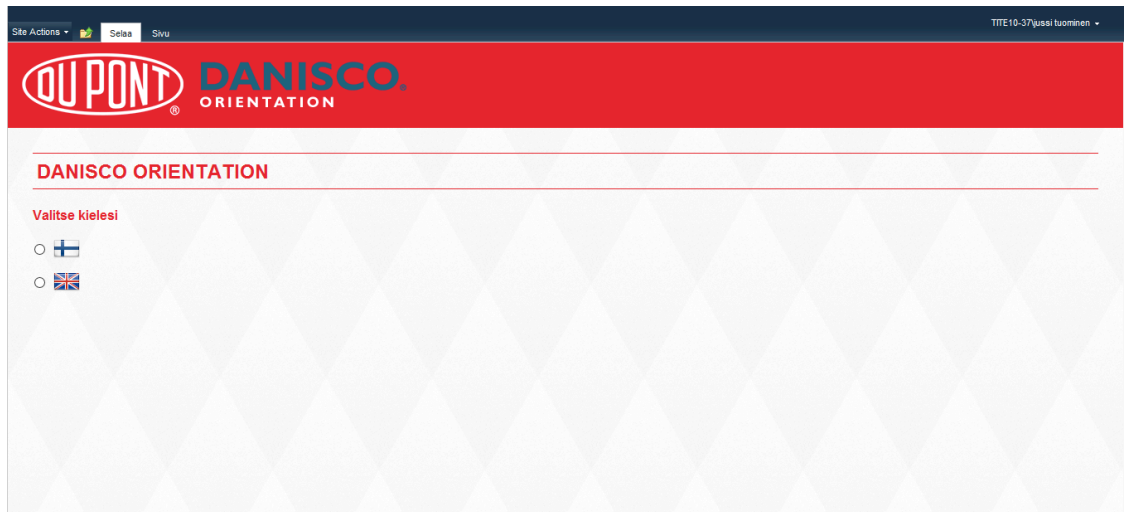
Danisco Orientation -sovellus koostuu kahdesta erillisestä sivustosta: varsinaisesta perehdytysivustosta sekä hallintasivustosta, jonne on pääsy ainoastaan myönnettyille käyttäjille. Hallintasivusto on perehdytysivuston sisällä itsenäisesti toimiva alisivusto (*Subsite*). Tässä kappaleessa selostetaan sovelluksen sisältämien sivustokokonaisuuksien rakenne ja perustoiminta.

2.1 Perehdytysivusto

Varsinainen perehdytysohjelma on avoin kaikille työasemalle kirjautuneille käyttäjille. Muutoksien tekeminen ja sivuston työkalujen käyttö vaatii kuitenkin ylläpitäjän oikeudet. Perehdytysohjelma aloitetaan valitsemalla kieli etusivulta ja tämän jälkeen

aloitetaan käyttäjien tunnistaminen. Mikäli käyttäjä on aivan uusi ja ei ole suorittanut perehdytystä vielä ollenkaan, hänen täytyy ensimmäiseksi rekisteröidä ohjelmaan omat tietonsa. Jos käyttäjä on jo aiemmin suorittanut perehdytyksen, hän voi etsiä omia tietojaan rekisteristä.

Kun perehdytysohjelma käynnistetään, sivulle ilmestyy ensimmäiseksi kielivalinta.



Kuva 1. Perehdytys sivuston etusivu

Kielen valitsemisen jälkeen siirrytään seuraavalle sivulle ruutuun ilmestyvän painikkeen avulla.



Kuva 2. Sovelluksen kielen valinta

2.1.1 Uusi käyttäjä

Mikäli käyttäjä tulee ensimmäistä kertaa suorittamaan perehdytystä, on hänen aivan ensimmäiseksi rekisteröitävä tietonsa ohjelman rekisteriin. Tämä tapahtuu aloitus sivun vasemmasta palstasta, jossa sijaitsevalle lomakkeelle hän voi syöttää kaikki tarvittavat tiedot.

Uusi käyttäjä


Ole hyvä ja rekisteröi tietosi sovellukseen oheisella lomakkeella.

Lisää perustiedot


Etunimi

Sukunimi

Syntymäaika




Yritys

--- Valitse --- 

Jos yritystä ei ole listassa, kirjoita yrityksen nimi oheiseen tekstikenttään.

Yhteyshenkilö

--- Valitse --- 

Olen tarkistanut, että yllä olevat tiedot ovat oikein.

Kuva 3. Lomake uuden käyttäjän tietoja varten

Kaikki tiedot ovat pakollisia ja ohjelma ilmoittaa siitä, mikäli kaikkia kenttiä ei ole täytetty. ”Syntymäaika”-kenttä eroaa muista siinä, että se ei ole manuaalisesti täytettävä. Tieto valitaan sitä vastoin kentän oikeassa reunassa näkyvästä kalenterikuvakkeesta, josta käyttäjälle aukeaa kalenterinäkömä päivämäärän valitsemista varten.



Uusi käyttäjä

Ole hyvä ja rekisteröi tietosi sovellu

Lisää perustiedot

Etunimi

Sukunimi

Syntymäaika

Yritys

--- Valitse ---

2006 - 2017		
2006	2007	2008
2009	2010	2011
2012	2013	2014
2015	2016	2017

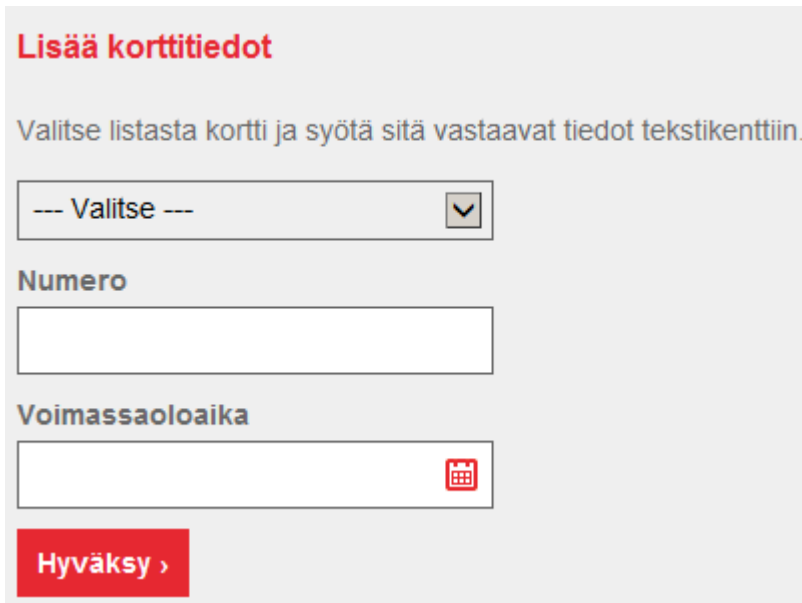
Kuva 4. Syntymäajan valinta kalenterinäkömästä

Kun käyttäjän perustiedot on syötetty, hänen on tarkistettava tietojen oikeellisuus. Tämän toimenpiteen varmistamiseksi käyttäjän on vahvistettava tietojen tarkistaminen valintaruudun avulla.

Olen tarkistanut, että yllä olevat tiedot ovat oikein.

Kuva 5. Perustietojen oikeellisuuden varmistusvalinta

Lopuksi painetaan ”Hyväksy ja jatka” -painiketta, joka vie seuraavaan vaiheeseen: korttien lisäämiseen.



Lisää korttitiedot

Valitse listasta kortti ja syötä sitä vastaavat tiedot tekstikenttiin.

--- Valitse ---

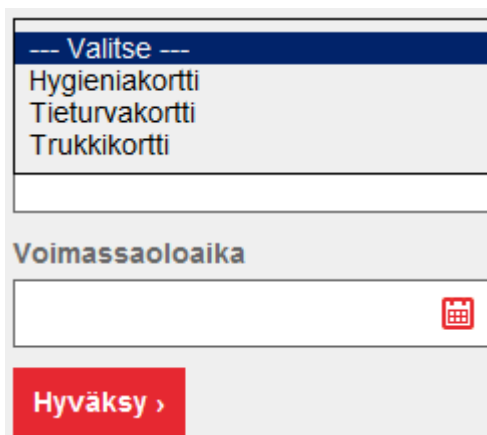
Numero

Voimassaoloaika

Hyväksy >

Kuva 6. Työturvallisuuskorttitietojen lisäämislomake uuden käyttäjän osalta

Käyttäjä valitsee kortin kerrallaan alasvetovalikosta ja syöttää sille lisätiedot tekstikenttiin. Valintalista rakennetaan tietokannasta löytyvillä tiedoilla.



--- Valitse ---

- Hygieniakortti
- Tieturvakortti
- Trukkikortti

Voimassaoloaika

Hyväksy >

Kuva 7. Korttien valintalista

Lisätyt kortit ilmestyvät lomakkeen alapuolelle sitä mukaa, kun niitä lisätään.

Nykyiset kortit

Nimi	Numero	Voimassaoloaika		
Trukkikortti	08923 423	06-2023	Muokkaa	Poista
Kattotulityökortti	2345234	11-2015	Muokkaa	Poista

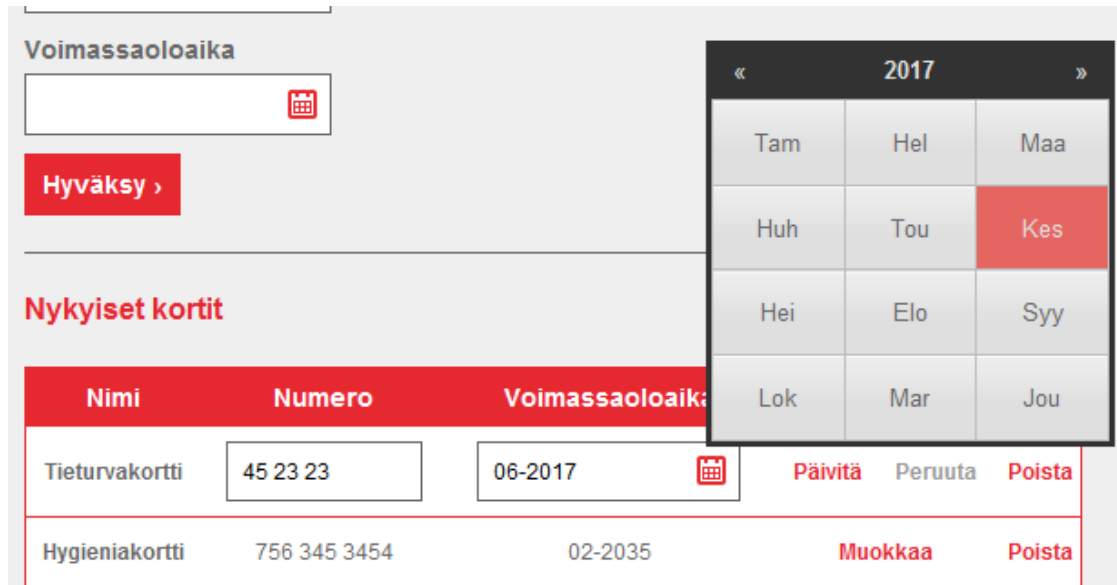
Kuva 8. Uuden käyttäjän lisäämät kortit lisäämislomakkeen alapuolella

Mikäli korttien tiedot olisi syötetty jostain syystä väärin, niitä pystyy muokkaamaan vielä jälkepäin ”Muokkaa”-linkistä. Kortin numeron ja voimassaoloajan kohdalle ilmestyy tällöin tekstikentät, johon muokatut arvot on mahdollista syöttää.

Nimi	Numero	Voimassaoloaika			
Tieturvakortti	<input type="text" value="45 23 23"/>	<input type="text" value="06-2017"/> 	Päivitä	Peruuta	Poista
Hygieniakortti	756 345 3454	02-2035	Muokkaa	Poista	

Kuva 9. Uuden käyttäjän lisäämien korttien muokkausnäkyvä

Voimassaoloaika valitaan aiemmin näytetyllä kalenterinäkyvällä. Tässä kohtaa ainoastaan kuukauden ja vuoden mukaan.



Voimassaoloaika

« 2017 »

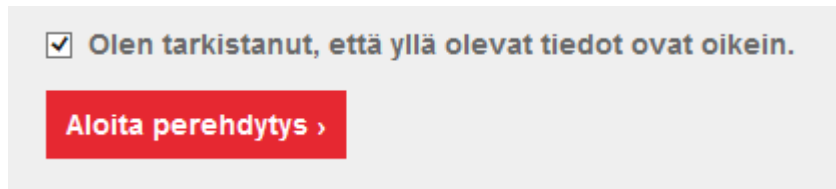
Tam	Hel	Maa
Huh	Tou	Kes
Hei	Elo	Syy
Lok	Mar	Jou

Nykyiset kortit

Nimi	Numero	Voimassaoloaika	
Tieturvakortti	45 23 23	06-2017	Päivitä Peruuta Poista
Hygieniakortti	756 345 3454	02-2035	Muokkaa Poista

Kuva 10. Kortin voimassaoloajan lisääminen kalenterinäköymästä

Kun korttien tiedot ovat oikein, käyttäjän on vielä varmistettava niiden oikeellisuus. Tämä tapahtuu valitsemalla valintaruutu, jonka jälkeen sen alapuolelle ilmestyvästä painikkeesta hän on valmis aloittamaan perehdytyksen.



Olen tarkistanut, että yllä olevat tiedot ovat oikein.

Aloita perehdytys

Kuva 11. Korttitietojen oikeellisuuden varmistaminen

2.1.2 Palaava käyttäjä

Mikäli käyttäjä on suorittanut perehdytyksen aikaisemmin, löytyvät hänen tietonsa ohjelman rekisteristä. Kun käyttäjä tulee suorittamaan ohjelmaa uudelleen, valitsee hän perehdytyksen aloitussivun oikean palstan, missä sijaitsevan tekstikentän avulla hän voi etsiä nimen mukaan (etu- tai sukunimi tai koko nimi) tietojaan ohjelmasta.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013

Palaava käyttäjä

Voit etsiä tietojasi **nimen mukaan** oheisella lomakkeella.

Kuva 12. Palaavan käyttäjän valinta

Hakutermi on pakollinen tieto ja sovellus ilmoittaa siitä, mikäli sitä ei ole annettu. Mikäli haulle löytyi käyttäjiä, tiedot ilmestyvät hakulomakkeen alapuolelle.

Nimi	Syntymäaika	Toiminnot
Cyberman Show	25.09.1959	<input type="button" value="Muokkaa"/> <input type="button" value="Valitse ja jatka >"/>

Kuva 13. Järjestelmästä löytyneet käyttäjätiedot

Käyttäjälle annetaan mahdollisuus joko tarkastella ja muokata olemassa olevia tietojaan tai siirtyä jo tässä kohtaa suoraan perehdytykseen. Käyttäjän omalla vastuulla on tiedostaa, mitkä hänen informaatiostaan ovat vanhentuneet ja toimia sen mukaisesti tässä vaiheessa.

Tietojen muokkaamiseen pääsee ”Toiminnot”-sarakkeen ”Muokkaa”-painikkeesta. Tämän jälkeen käyttäjän on syötettävä henkilökohtainen, viisinumeroinen varmennusnumerosa, jonka hän on saanut edellisen perehdytysosuituksen yhteydessä. Tämä numero on tulostettu perehdytystodisteen takapuolelle oikeaan alareunaan.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013

KÄYTTÄJÄN VARMISTUS

Jatkaaksesi eteenpäin, syötä henkilökohtainen varmennuskoodisi oheiseen kenttään.

Hyväksy ja jatka »

Kuva 14. Palaavan käyttäjän varmistusikkuna

Mikäli numero syötetään väärin, ohjelma ilmoittaa siitä käyttäjälle ja vaatii, että se syötetään oikein. Tekstikenttään ei voi syöttää viittä numeroa pitempää numerosarjaa. Kenttä hyväksyy syötteiksi ainoastaan numeroita.

Hyväksytyn varmistusnumeron syöttämisen jälkeen käyttäjä ohjataan seuraavalle sivulle, missä hän voi muokata omia perustietojaan. Järjestelmästä löytyvät tiedot näkyvät valmiiksi lomake-elementeissä.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013

Etunimi
Paavo

Sukunimi
Pesusieni

Syntymäaika
1982-08-26 

Yhteyshenkilö
Kalevi Rantanen 

Yritys
Testing Oy 

Jos yritystä ei ole listassa, kirjoita yrityksen nimi oheiseen tekstikenttään.

Hyväksy ja jatka >

Kuva 15. Käyttäjän perustietojen muokkauslomake

Syntymäajan muokkaus tapahtuu klikkaamalla tekstikentässä oikealla näkyvää kalenterikuvaketta. Kuvakkeen klikkauksen jälkeen sivulle aukeaa kalenteri-ikkuna, mistä voidaan valita haluttu päivämäärä. Kalenterin otsikkopalkin painikkeita klikkaamalla voidaan selata vuosia ja vaihtaa näkymiä vuosista kuukausiin ja kuukausista päiviin. Tässä kohtaa on hyvä huomata se, että itse tekstikenttä on ”vain luku” -muodossa, joten sen päivittäminen manuaalisesti (kirjoittamalla siihen tekstiä) ei ole mahdollista. Tällä varmistetaan se, että syöte menee joka tapauksessa oikeassa muodossa eteenpäin.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013

The screenshot shows a registration form for 'DANISCO ORIENTATION'. The form includes fields for 'Etunimi' (First name) with the value 'Paavo', 'Sukunimi' (Last name) with the value 'Pesusieni', and 'Syntymäaika' (Date of birth) with the value '1982-08-26'. A calendar pop-up is open over the date field, showing a grid of years from 1975 to 1986. The year 1982 is highlighted in red, indicating it is the selected year.

Kuva 16. Syntymäajan (ja muidenkin ajallisten tietojen) valintaan käytettävä kalenteri-ikkuna

Kaikki tiedot lomakkeella on täytettävä. Jos jokin tieto puuttuu, ohjelma antaa siitä virheilmoituksen. Kun päivitettyt tiedot on syötetty ohjelmaan, painetaan ”Tallenna ja jatka” -painiketta. Tässä kohtaa on hyvä mainita sekin, että jos mitään tietoja ei ole tarvetta muuttaa, voidaan siirtyä suoraan seuraavalle sivulle (olettaen, että mikään tieto ei ole lomakkeella tyhjä).

Käyttäjätietojen muokkauksen toisessa vaiheessa voidaan tarkastella ja päivittää työturvallisuuskorttitietoja. Sivulla näkyvät käyttäjän ohjelmaan rekisteröimät korttitiedot: kortin nimi, numero ja voimassaoloaika. Hänen on mahdollista päivittää olemassa olevien korttien tietoja tai poistaa kortteja.

Nykyiset kortit

Nimi	Numero	Voimassaoloaika		
Tieturvakortti	324 23	07-2017	Muokkaa	Poista

Kuva 17. Käyttäjän nykyiset työturvallisuuskortit

Kunkin kortin muokkaaminen tapahtuu klikkaamalla oikealla olevaa ”Muokkaa”-linkkiä.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013

Nykyiset kortit

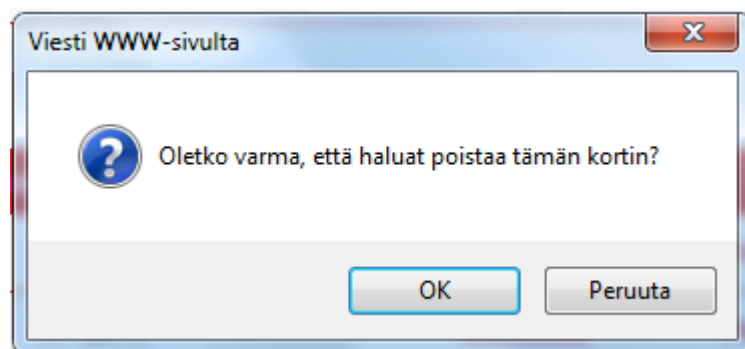
Nimi	Numero	Voimassaoloaika	
Tieturvakortti	324 23	07-2017	Päivitä Peruuta Poista

Kuva 18. Olemassa olevan kortin muokkaaminen

Näkymään ilmestyy kortin numeron ja voimassaolon muokkaamiseksi kaksi tekstikenttää, joihin käyttäjä voi päivittää uudet tietonsa. Tietojen päivittäminen järjestelmään tapahtuu oikealla puolella näkyvän ”Päivitä”-linkin avulla. Päivitystapahtuman voi myös peruuttaa ”Peruuta”-linkillä, jolloin näkymä palautuu alkuperäiseen muotoonsa mitään tietoja päivittämättä.

Voimassaolon syöttäminen tapahtuu samalla tavoin kalenterin avulla kuin edellisessä vaiheessa tapahtunut syntymäajan päivittäminen. Erona on se, että tässä vaiheessa valitaan ainoastaan vuosi ja kuukausi. Muuten toimintaperiaate on sama.

Korttia poistettaessa käyttäjälle pyydetään siitä vielä varmistus. Mikäli käyttäjä klikkaa ”OK”-painiketta, kortti poistetaan hänen tiedoistaan.



Kuva 19. Työturvallisuuskortin poistamisen varmistusikkuna

Jos käyttäjällä on olemassa jokin uusi kortti, jota ei sovelluksen rekisteristä vielä löydy, voi hän lisätä sen sinne samalta sivulta löytyvällä lomakkeella. Lomakkeen kaikki tiedot ovat pakollisia. Kortin valintavalikossa näytetään ainoastaan järjestelmästä löytyvät kortit, joita ei käyttäjällä vielä ole olemassa. Eli kahta samanlaista korttia ei voi olla yhdellä ja samalla käyttäjällä.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013

Nykyiset kortit

Nimi	Numero
Tieturvakortti	324 23

Lisää uusi kortti

--- Valitse ---

Kattotulityökortti

Trukkikortti

Tulityökortti

Kuva 20. Uuden työturvallisuuskortin lisääminen sovellukseen. Alasvetovalikko näyttää ainoastaan mahdolliset lisättävät kortit, jotka löytyvät sovelluksen rekisteristä.

Kun kaikki kortit on saatu lisätyksi järjestelmään ja päivitettyä ajan tasalle, klikataan oikeassa alakulmassa näkyvää valintaruutua. Tällä varmistetaan se, että käyttäjä on varmasti tarkistanut ja hyväksynyt omat työturvallisuuskorttitietonsa. Kun valintaruutu on valittuna, sen alapuolelle ilmestyy painike, jota painamalla aloitetaan perehdytys.

Olen tarkistanut, että korttitietoni ovat ajan tasalla.

Aloita perehdytys »

Kuva 21. Työturvallisuuskorttitietojen varmistaminen ja perehdytyksen aloituspainike

HUOMAA: Sovellukseen on rakennettu suojaus estämään sen, että kuka tahansa ei voi mennä muokkaamaan toisten tietoja, eikä sivuille ole suoraa pääsymahdollisuutta.



2.1.3 Perehdytys

Itse perehdytystä aloitettaessa ensimmäinen vaihe on suoritettavien moduulien valitseminen. Niitä voi valita yhden tai useamman.



Kuva 22. Suoritettavien moduulien valintanäkymä.

Kun suoritettavat moduulit on valittu, siirrytään eteenpäin ”Aloita perehdytys” -painikkeesta. Moduulit tulevat käyttäjän suoritettaviksi **siinä järjestyksessä, kun ne listassa näkyvät.**

Aloita perehdytys >

Perehdytyksen aikana on nähtävissä sen hetkinen tilanne (missä moduulissa ja missä materiaalissa ollaan menossa), materiaalin avauspainike sekä painike kysymyksien ja vastausvaihtoehtojen tarkastelua varten.

Näytä kysymys ja vastaukset >

HUOMAA: Mikäli kyseessä on materiaalin ensimmäinen kysymys, kysymyksen / vastausvaihtoehtojen tarkastelupainike on poissa käytöstä. Tällä varmistetaan se, että käyttäjä on varmasti tutustunut materiaaliin ennen siihen liittyvien kysymysten katsomista.

Materiaali aukeaa ruudun kattavaan ikkunaan, jonka voi sulkea joko ESC-näppäimestä, ikkunan ulkopuolella klikkaamalla tai varsinaisesta sulkemispainikkeesta ikkunan oikeassa alakulmassa.



Käyttäjä valitsee vastausvaihtoehdoista sopivimmat ja tarkastaa valintansa ”Tarkista vastaukset” -painikkeella. Oikeita vastauksia voi olla yhdestä useampiin, mutta aina vähintään yksi kappale.

Tarkista vastaukset >

HUOMAA: Ohjelma hyväksyy vastauksen oikeaksi vain silloin, kun ainoastaan oikeat vastaukset on valittu. Näin ollen, mikäli käyttäjän valitsemien oikeiden vastausten joukossa olisi myös väriä vastauksia, vastaus on tällöin väärin.

Sovellus tarkastaa vastaukset ja antaa palautetta käyttäjälle sen mukaan. Palaute näkyy tarkastuspainikkeen oikealla puolella eri väreillä (punainen = väärin, vihreä = oikein).

Mikäli vastaus on mennyt oikein, vastausvaihtoehdot ja tarkastuspainike poistuvat käytöstä ja näytölle ilmestyy painike seuraavaan vaiheeseen etenemiseksi (kyseinen vaihe voi olla seuraava moduuli / materiaali / kysymys tai lopetussivu).

Tarkista vastaukset >

Vastasit oikein!

Siirry eteenpäin >

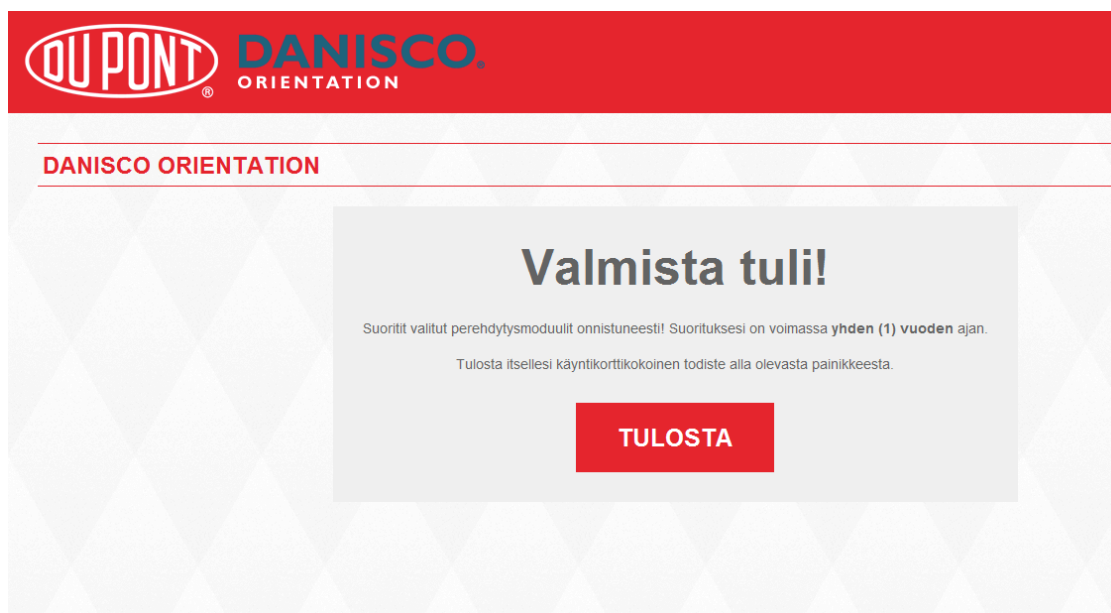
HUOMAA: Perehdytysprosessin aikana ohjelma tallentaa tiedon järjestelmään ilman erillistä ilmoitusta aina silloin, kun käyttäjä on saanut tietyn moduulin suoritettua päätökseen.

Mikäli käyttäjä on vastannut kysymykseen väärin, hän voi avata materiaalin uudelleen tarkentaakseen tietojaan ja vaihtaakseen valitsemiaan vastauksia. Kun kaikkien valittujen moduulien materiaalit ja niihin liittyvät kysymykset on suoritettu, käyttäjä ohjataan lopuksi uudelle sivulle, missä häntä tiedotetaan tilanteesta ja mistä hän voi tulostaa itselleen todisteen suoritetusta perehdytyksestä.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013



Kuva 23. Perehdytyksen lopetussivu

Tulostuspainikkeesta avataan tulostettava PDF-tiedosto ja samalla sivulle ilmestyy ”Palaa etusivulle” -linkki.

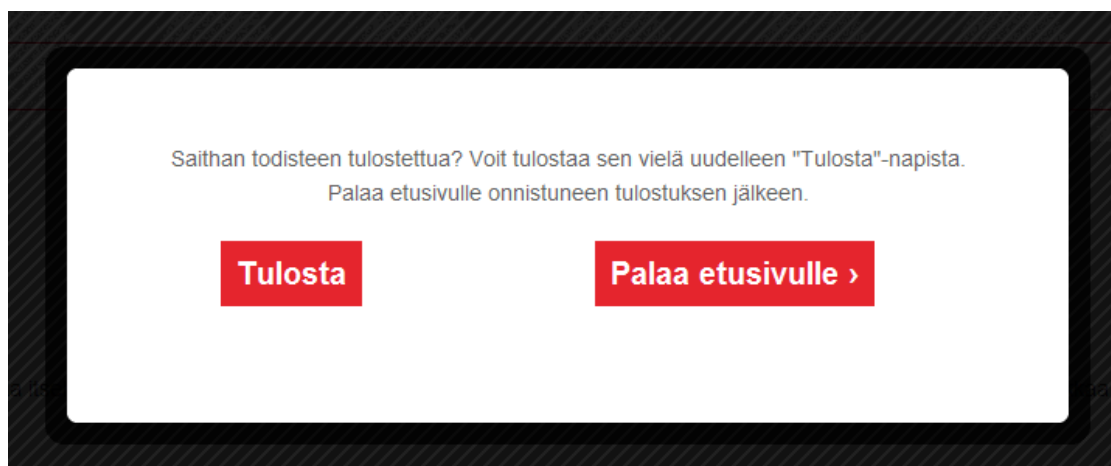


Tätä linkkiä painamalla avautuu ruutuun vielä varmistusikkuna, joka varmistaa, että käyttäjä on onnistuneesti saanut tulostettua perehdytystodisteen. Ikkuna sisältää painikkeen, jonka kautta on vielä mahdollisuus tulostaa se. Lopuksi ”Palaa etusivulle” -painiketta painamalla käyttäjä palaa takaisin perehdytysohjelman etusivulle. Perehdytys on suoritettu päätökseen.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013



Kuva 24. Käyttäjältä varmistetaan vielä, että tulostaminen on onnistunut. Tästä ikkunasta palataan myös takaisin etusivulle.

HUOMAA: Sovellukseen on määritetty yhden perehdytystapahtuman maksimikestoksi 1 tunti (60 minuuttia). Yksi sessio kestää siis maksimissaan tämän verran, kunnes väliaikaiset tiedot pyyhkiytyvät muistista. Ne pyyhitään normaalisti perehdytystapahtuman lopussa.

2.2 Hallintasivusto

Hallintasivustolla voi käsitellä monin tavoin sovelluksen sisältämiä tietoja kuten kortteja, yrityksiä, yhteyshenkilöitä ja perehdytykseen liittyviä aineistoja. Lisäksi käyttäjien työturvallisuuskorttien ja perehdytysasuoritusten vanhentumista voi seurata keskitetysti. Hallintasivustolle on pääsy ainoastaan tiettyyn käyttäjäryhmään lisätyillä henkilöillä.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013

Tämä on Danisco Orientation -sovelluksen admin-paneeli. Tervetuloa!

Vanhentuneet 10 perehdytysuoritusta

Työntekijän nimi	Moduuli	Suorituspvm
DJ Renato	Urakoitsijat	19.05.2012
Cyberman Show	Vierailijat	13.09.2012

Vanhentuneet 10 korttia

Työntekijän nimi	Kortin nimi	Kortti voimassa
Paavo Pesusieni	Tieturvakortti	09-2012
Test Man	Hygieniakortti	02-2013

Kuva 25. Danisco Orientation -sovelluksen hallintasivuston etusivu

Etusivulla on tilanteen mukaan kaksi oleellista informaatioelementtiä koskien käyttäjiä:

- Käyttäjät ja heidän perehdytysuorituksensa, jotka ovat menneet vanhaksi
- Käyttäjät ja heidän korttinsa, jotka ovat menneet vanhaksi

Valikon linkkejä klikkaamalla pääsee muokkaamaan eri kategorioita. Kukin sivu sisältää loogisuuden painottamiseksi pääosin samanlaisen, taulukkomuotoisen tavan esittää dataa.

2.2.1 Käyttäjiin liittyvän datan hallinta

Seuraavissa kappaleissa keskitytään sovelluksen käyttäjiin liittyvän informaation (työturvallisuuskortit, käyttäjät, yritykset, yhteyshenkilöt) hallintaan.

2.2.1.1 Tietojen tarkastelu

Hallintasivun eri kategorioiden etusivulla näytetään näkymä kaikista sovelluksesta löytyvistä tiedoista. Esimerkkinä tässä ohjeessa käytetään ”Käyttäjät”-sivua.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013

KÄYTTÄJÄT

Sovelluksesta löytyvät käyttäjät

Näytä vain käyttäjät, joilla on vanhentuneita suorituksia Näytä vain käyttäjät, joilla on vanhentuneita kortteja

Etunimi	Sukunimi	Synt.pv.	Yhteyshenkilö	Yritys	Suoritukset	Kortit	Avainnro	Kaukolaukaisin
Ari	Tuomi	22.05.1986	Kalevi Rantanen	Socialnerd Oy	Suoritukset »	Kortit »		Muokkaa Poista Tuloste
Olli	Opiskelija	25.09.1959	Steven Bannister	Turpaan Veto Oy	Suoritukset »	Kortit »		Muokkaa Poista Tuloste
Paavo	Pesusiemi	26.11.1987	Kalevi Rantanen	Socialnerd Oy	Suoritukset »	Kortit »		Muokkaa Poista Tuloste
Kimmo	Ollila	13.05.1971	Antti Soukka	Nogia Oy	Suoritukset »	Kortit »		Muokkaa Poista Tuloste
Elena	Hietanen	20.11.1957	Antti Soukka	Maansiirto Oy	Suoritukset »	Kortit »		Muokkaa Poista Tuloste
Jason	Satham	22.05.1971	Veikko Yrjölä	Turpaan Veto Oy	Suoritukset »	Kortit »		Muokkaa Poista Tuloste
Käyttäjä	Yksi	24.07.1985	Kalevi Rantanen	UusiYritys1111 Oy	Suoritukset »	Kortit »		Muokkaa Poista Tuloste

Näytä kaikki

Kuva 26. Sovelluksesta löytyvien tietojen esitysnäkymä hallintasivulla

Taulukon tietoja voi järjestää klikkaamalla otsikkorivejä. Tämä on yleinen ominaisuus kaikissa hallintapaneelin sivuissa, joissa data esitetään sivulla taulukkomuodossa. Oletuksena tiedot on sivutettu niin, että listassa näkyy yhdellä sivulla 25 riviä kerrallaan. ”Näytä kaikki” -painikkeella (taulukon oikea alareuna) saadaan kaikki tiedot näkyviin yhdelle sivulle. Sivutuksen voi laittaa takaisin päälle samasta painikkeesta.

Käyttäjäraporttien tulostamiseksi on olemassa erillinen sivu, jonne pääsee ”Käyttäjät”-sivun alareunasta ”Tulosta raportteja käyttäjistä” -linkillä.

[Tulosta raportteja käyttäjistä →](#)

PDF-tiedoston tulostus (tai tallennus) tapahtuu vasemmassa yläkulmassa olevan painikkeen takaa.

Tallenna / Tulosta PDF

Käyttäjätietojen näkymää voi muokata mieleisekseen seuraavilla tavoilla:

- Näytetään ainoastaan käyttäjät, joilla on vanhentuneita perehdytys-suorituksia
- Näytetään ainoastaan käyttäjät, joilla on vanhentuneita työturvallisuuskortteja
- Näytetään ainoastaan tietyn yrityksen käyttäjät

Nämä työkalut löytyvät käyttäjätietotaulukon yläpuolelta. Vain yksi edellä mainituista suodattimista voi olla päällä kerrallaan.



Danisco Sweeteners Oy
Sokeritehtaantie 10
PL 213, 48101 KOTKA, Finland

DANISCO ORIENTATION
Perehdytysohjelma
Käyttöohje | 29.10.2013



Kuva 27. Työkalut tulostettavan taulukon muokkaamiseksi siihen muotoon, kuin se halutaan tulostettavan.

Mikäli käyttäjällä on tiedoissaan vanhentuneita tietoja joko suoritusten tai työturvallisuuskorttitietojen osalta (tai sekä että), hänen rivillään näytetään kyseisessä sarakkeessa punainen huomautusmerkki.

Suoritukset »	Kortit »
Suoritukset » !	Kortit »
Suoritukset »	Kortit » !
Suoritukset » !	Kortit »
Suoritukset »	Kortit » !
Suoritukset »	Kortit »

Kuva 28. Huomautusmerkit käyttäjälistassa

Käyttäjien tietoja voi suodattaa niin, että sivulla näytetään ainoastaan käyttäjät, joilla on vanhentuneita suorituksia tai vanhentuneita kortteja. Suodatus tapahtuu informaatiotaulukon yläpuolen valintaruutujen avulla.

Näytä vain käyttäjät, joilla on vanhentuneita suorituksia Näytä vain käyttäjät, joilla on vanhentuneita kortteja

Kuva 29. Käyttäjätietojen suodatusvalinnat

2.2.1.2 Tietojen muokkaaminen ja poistaminen

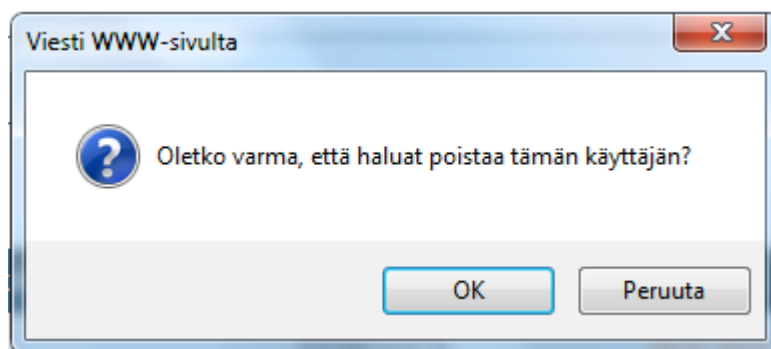
Tietoja voi muokata ja poistaa taulukkonäkymien oikean reunan ”Muokkaa”- ja ”Poista”-linkkien avulla. ”Muokkaa”-linkkiä klikattaessa kyseessä olevan taulukon rivi siirtyy muokkaustilaan. Muokattavien tietojen tilalle ilmestyy tekstikenttä, johon uuden tiedon voi päivittää. Käyttäjien kohdalla muokattavissa olevia tietoja ovat avainnumero ja kaukolaukaisimen numero.



s	Suoritukset	Kortit	Avainnro	Kaukolaukaisin	
Inerd Oy	Suoritukset »	Kortit »			Muokkaa Poista
ritys Oy	Suoritukset » !	Kortit »			Muokkaa Poista
ng Oy	Suoritukset »	Kortit » !			Muokkaa Poista
ng Oy	Suoritukset » !	Kortit »	<input type="text"/>	<input type="text"/>	Päivitä Peruuta Poista
	Suoritukset »	Kortit » !			Muokkaa Poista

Kuva 30. Muokkaustila päällä käyttäjätaulukossa

Tieto päivitetään klikkaamalla ”Päivitä”-linkkiä. Muokkauksen voi myös ottaa pois päältä tallentamatta muutoksia ”Peruuta”-linkistä. Tiedon poisto tapahtuu punaisesta ”Poista”-linkistä. Käyttäjältä vielä varmistetaan, haluaako hän tiedon poistettavan.



Kuva 31. Tiedon poistamisen varmistaminen hallintapaneelissa

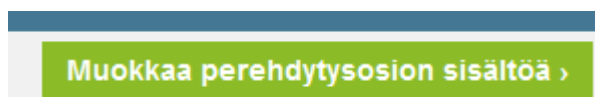
HUOMAA: Jos poistat käyttäjän, poistuvat sovelluksen rekisteristä myös kaikki hänen työturvallisuuskorttitietonsa ja perehdytysasuorituksensa!

Taulukon riveillä näkyvät myös ”Tuloste”-linkit, joista voi tulostaa tietylle käyttäjälle perehdytystodisteen, mikäli se olisi esimerkiksi hävinnyt tai vioittunut. Käyttäjiä voi etsiä tietokannasta joko nimen tai yrityksen mukaan sivulla näkyvän lomakkeen avulla.



2.2.2 Perehdytykseen liittyvän datan hallinta

Seuraavissa kappaleissa keskitytään sovelluksen perehdytykseen liittyvän informaation (moduulit, perehdytysaineistot, kysymykset, vastaukset) hallintaan. Perehdytysosioon liittyvän datan muokkaustyökaluihin pääsee navigaation oikean reunan painikkeesta.



Kuva 32. Painike perehdytysosion tietojen hallintasivulle siirtymiseksi

2.2.2.1 Tietojen tarkastelu

Perehdytysosion tietojen hallintasivulla on neljä kategoriaa: moduulit, materiaalit, kysymykset ja vastaukset. Kunkin sivun rakenne noudattaa pääasiassa samanlaista asetelua: ensin sovelluksesta löytyvät tiedot esitetään taulukossa ja sen alapuolelta löytyy lomake, josta uuden tiedon voi lisätä. Sivulla näkyvistä alavetovalikoista on valittu haluttu moduuli ja sinne lisätty aineisto, jonka kysymykset halutaan näytettävän.

Alavetovalikot näytetään sivuilla porrastetusti:

- Moduulit Ei alavetovalikoita.
- Materiaalit ”Valitse moduuli”
- Kysymykset ”Valitse moduuli” → ”Valitse materiaali”
- Vastaukset ”Valitse moduuli” → ”Valitse materiaali” → ”Valitse kysymys”



PEREHDYTYSSOSION HALLINTA

Moduulit Materiaalit Kysymykset Vastaukset

Hallitse kysymyksiä

Urakoitsijat Tulitöiden sähköistäminen erinäisillä menetelmillä

Kysymys	Kysymys (EN)
Mikä tämä on?	

Lisää uusi kysymys

Kysymys

Kysymys (EN)

Lisää uusi kysymys >

Kuva 33. Kysymysten muokkaussivun näkymä tietyn aineiston osalta

2.2.2.2 Tietojen muokkaaminen ja poistaminen

Tietoja voi muokata ja poistaa taulukoiden oikeassa reunassa sijaitsevien linkkien avulla.



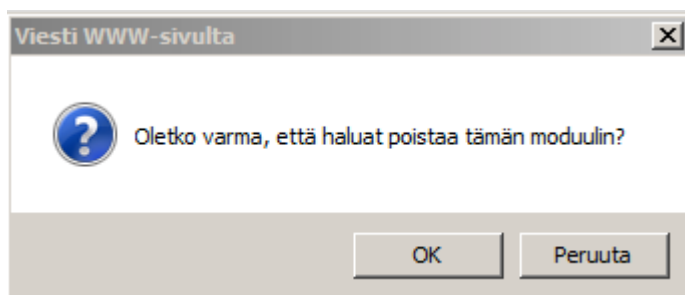
Kuva 34. Tietojen muokkaus- ja poistotyökalu perehdytysosion tietojen muokkaussivulla

Kun painetaan ”Muokkaa”-painiketta, kyseinen rivi muuttuu editointitilaan, jolloin sen sisältöjä voidaan muokata.

Moduuli	Moduuli (EN)	Päivitä	Poista
Urakoitsijat	Contractors		

Kuva 35. Muokkaustila päällä perehdytysosion hallintasivulla

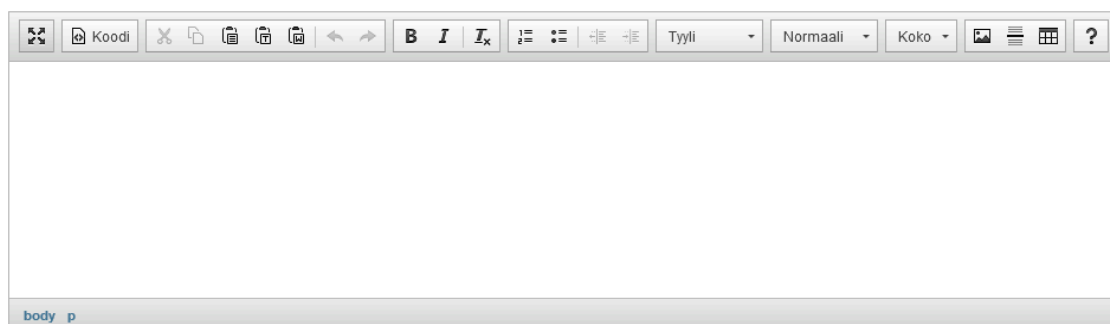
”Päivitä”-painikkeesta tieto luonnollisesti päivitetään ja ”Peruuta”-painikkeesta muokkaustilasta poistutaan tallentamatta. ”Poista”-painikkeesta tieto voidaan poistaa. Poisto varmistetaan käyttäjältä ennen kuin se suoritetaan.



Kuva 36. Tiedon poistamisen varmistaminen

HUOMAA: Jos poistat moduulin, poistuvat järjestelmästä kaikki siihen liittyvät aineistot, niihin liittyvät kysymykset ja kysymysten vastaukset. Lisäksi kaikki kyseisen moduulin suorittaneiden henkilöiden suoritustiedot pyyhkiytyvät. Jos poistat aineiston, poistuvat kaikki siihen liittyvät kysymykset ja vastaukset ja niin edelleen.

Perehdytysosion hallintasivujen muokkaustyökaluista eroaa materiaalien hallinta, jossa sisällön syöttämiseen käytetään rikastekstieditoria (ns. WYSIWYG-editori), joka tarkoittaa sitä, että sisältö voidaan esittää muotoiltuna tekstinä (HTML-muodossa) ja siihen voidaan näin ollen lisätä erilaisia tehostuksia. Materiaalin sisältöön voidaan lisätä myös kuvia.



Kuva 37. Materiaalien sisältöjen muokkaukseen käytetty tekstieditori.



Tekstieditorin työkalurivillä on saatavissa seuraavia painikkeita (vasemmalta oikealle):

- Suurena editori koko näytön kokoiseksi
- Näytä syötetty sisältö HTML-muodossa
- Kopioi / Liitä / Liitä tekstinä / Liitä teksti Microsoft Wordista (säilytä tekstin muotoilut)
- Peru / Tee uudelleen
- Tekstin lihavointi, kursivointi, muotoilujen poisto
- Numeroitu lista, numeroimaton lista, tekstin sisennys, sisennyksen poisto
- Tekstin tyyli
- Tekstin muotoilu
- Tekstin koko
- Lisää kuva sisältöön
- Lisää vaakaviiva
- Lisää taulukko
- Näytä tietoja editorista

Kuvan lisääminen

Mikäli materiaalin sisältöön halutaan lisätä kuvia, se tapahtuu seuraavan painikkeen takaa:





Kuva 38. Kuvan lisäämisikkuna

Kuva voidaan joko valita kirjastossa olemassa olevista tai lisätä kokonaan uusi kuva. Uusi kuva lisätään ”Lisää kuva” -välilehdeltä. Levyiltä valitaan haluttu kuva ja klikataan ”Lähetä palvelimelle”-painiketta.

Kuva 39. Kuvan lisääminen materiaaliin levyiltä



Tämän jälkeen näkymä siirtyy takaisin ”Kuvan tiedot” -välilehdelle. Sieltä kuvan asetuksia on mahdollista muokata. Näitä ovat:

- Kuvan teksti
- Kuvan koko
- Kehys ja sen paksuus
- Kuinka paljon tilaa asetetaan kuvan ympärille
- Kuvan sijoittaminen: tekstin lomaan oikealle tai vasemmalle puolelle tai tekstin väliin

Kuvan osoite lisätään ”Osoite”-kenttään automaattisesti, eikä sitä näin ollen tarvitse muokata. Kun kuva sijaitsee halutussa sijainnissa, painetaan ”OK”-painiketta. Tapah-tuman voi myös peruuttaa ”Peruuta”-painikkeesta.

Kuvan tiedot
Lisää kuva
Lisäominaisuudet

Osoite

Vaihtoehtoinen teksti

Leveys

Korkeus

Kehys

Vaakatila

Pystytila

Kohdistus

Esikatselu

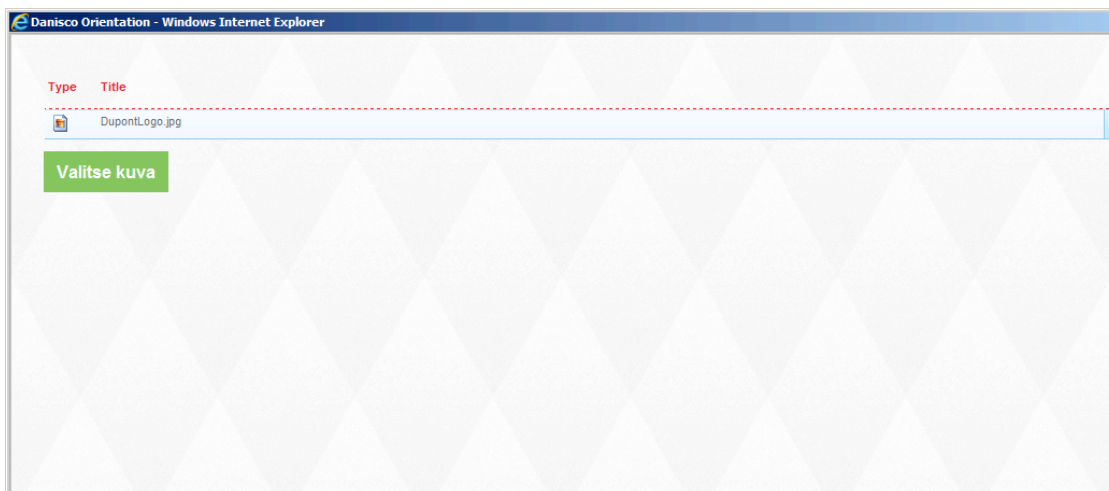
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas feugiat consequat diam. Maecenas metus. Vivamus diam purus, cursus a, commodo non, facilisis vitae, nulla. Aenean dictum lacinia tortor. Nunc iaculis, nibh non iaculis aliquam, orci felis euismod neque, sed ornare massa mauris

Kuva 40. Lisätyn kuvan tietojen muokkaus



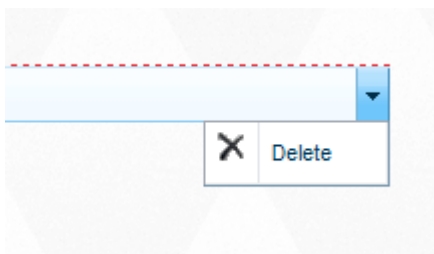
Lisätyn kuvan asettelua voidaan muokata myös jälkeinpäin kuvaa kaksoisklikkaamalla.

Kuva voidaan hakea editoriin myös olemassa olevasta kirjastosta. Tämä tapahtuu kuvan lisäämisikkunan oikeassa reunassa näkyvästä ”Selaa palvelinta” -painikkeesta, josta avautuu lista kirjastosta löytyvistä kuvista.



Kuva 41. Ikkuna olemassa olevan kuvan valitsemiseksi

Näkymästä valitaan haluttu kuva ja tämän jälkeen painetaan ”Valitse kuva” -painiketta. Sen jälkeen siirrytään takaisin edelliseen näkymään, jossa voidaan muokata kuvan asetuksia. Kuvan poistaminen onnistuu myös avautuneesta näkymästä: vietään hiiren kursori kuvan päälle ja valitaan näkyviin tullut alasvetovalikon avaamiseen tarkoitettu nuoli oikeasta reunasta, jonka takaa poistaminen onnistuu.



Kuva 42. Olemassa olevan kuvan poistaminen kirjastosta



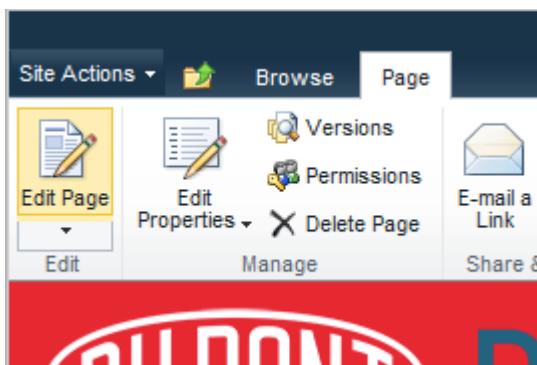
3 SIVUJEN MUOKKAAMINEN

Sovellus on toteutettu niin, että valmiita elementtejä ei ole tarvetta muokata: ne toimivat sellaisenaan ja on asetettu oikeille paikoilleen. Joillekin perehdytysosion sivulle on kuitenkin mahdollista lisätä tekstiä ja kuvia valmiina olevien elementtien rinnalle. Tämä toiminnallisuus on ohjelmaan toteutettu niin sanottujen web-osien eli Web Partien avulla. Tässä kappaleessa selostetaan pintapuolisesti, kuinka sisällön lisääminen sivulle onnistuu.

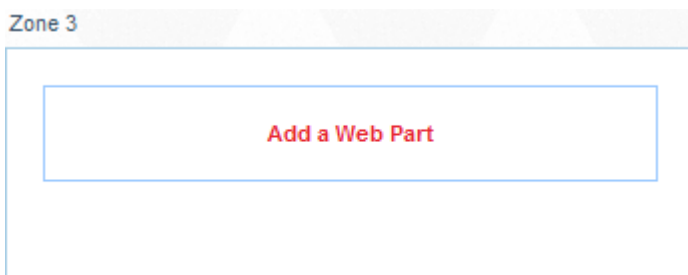
Muokataksesi sivun sisältöä, sinun tulee olla kirjautuneena sivulle admin-käyttäjänä. Sisältöä lisätään sivulle yläpalkin työkaluista (*Ribbon*). Valitaan *Page*-välilehti ja klikataan kohtaa *Edit Page*.

Sisällön lisääminen sivulle web-osien (Web Parts) avulla

Klikataan ”Edit Page”.

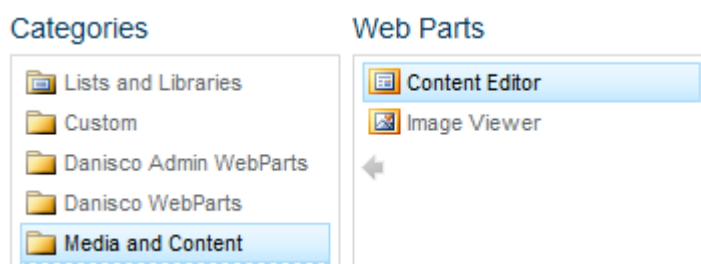


Sivulle ilmestyy ”Add Web Part” -linkkejä, joista klikkaamalla sivuille voi lisätä web-osia.

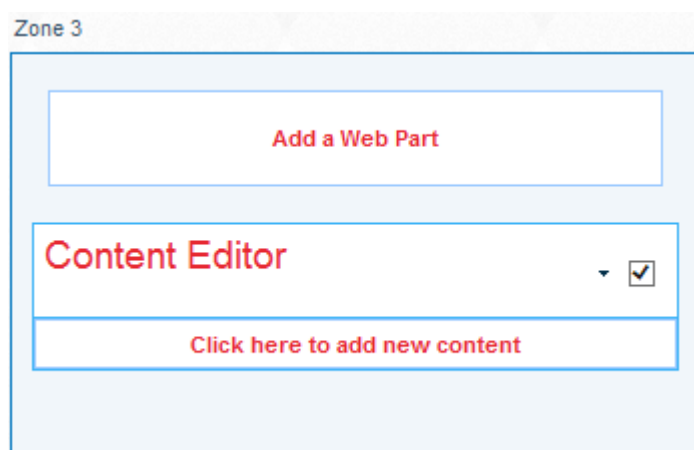




Teksti- ja kuvaweb-osat löytyvät kansioista "Media and Content".

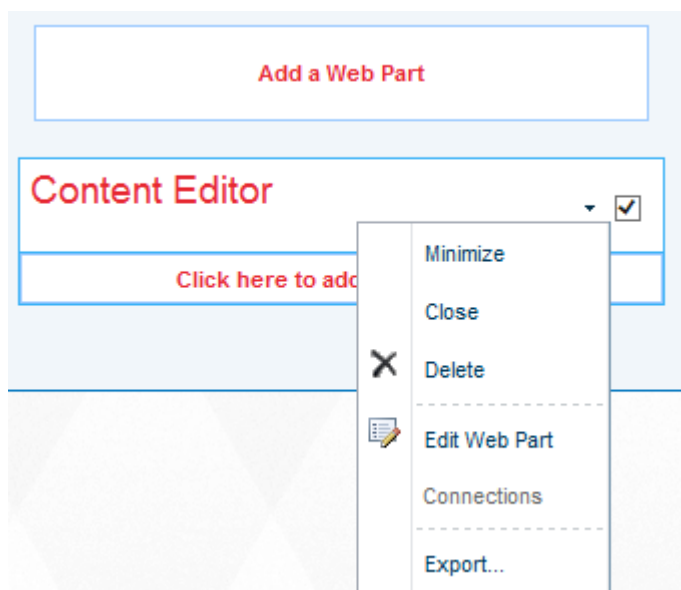


Web-osa lisätään sivuille valitsemalla sitä ja klikkaamalla "Add"-painiketta oikeasta reunasta. Tämän jälkeen web-osa muuttuu sivulla tämän näköiseksi. Esimerkkinä tässä on "Content Editor", joka mahdollistaa esimuotoillun tekstin lisäämisen sivulle.

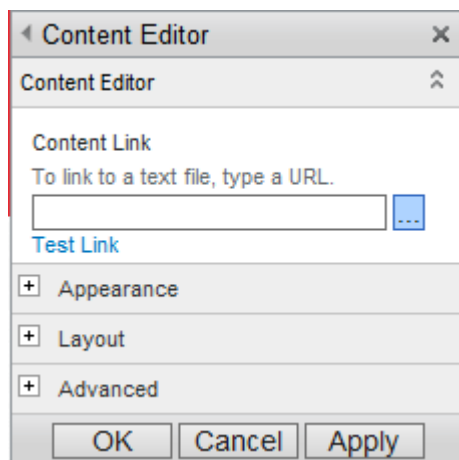


Tämän jälkeen klikataan "Click here to add new content" -kohtaa ja sitten siihen voidaan lisätä sisältöä. Mikäli web-osan asetuksia haluaa muokata, klikataan oikealla ylhäällä näkyvää pientä nuolta ja valitaan "Edit Web Part". Kyseisestä paikasta löytyy myös muita web-osaan liittyviä toimintoja:

- "Delete" – poistaa web-osan kokonaan
- "Close" – poistaa web-osan sivuilta, mutta lisää sen "Closed Web Parts" -galleriaan ja säilyttää sen asetukset myöhempää käyttöä varten
- "Minimize" – minimoi web-osan
- "Restore" – palauttaa minimoidun web-osan



Asetusikkuna näyttää seuraavanlaiselta. Pieniä +-merkkejä klikkaamalla voi avata eri kategorioita.



Kun web-osa on saatu sivulle halutun näköiseksi, klikataan ”Stop Editing” -kohtaa yläpalkin työkaluista.

