

Jarkko Laukkanen

**Web-osion rakentaminen SEAMK-opiskelijaportaaliin**

Opinnäytetyö

Kevät 2010

Yksikön virallinen nimi esim. Tekniikan yksikkö

Tietojenkäsittely

Sovellustuotanto



## SEINÄJOEN AMMATTIKORKEAKOULU

## OPINNÄYTETYÖN TIIVISTELMÄ

Koulutusyksikkö: Tekniikan yksikkö  
Koulutusohjelma: Tietojenkäsittelyn koulutusohjelma  
Suuntautumisvaihtoehto: Sovellustuotannon suuntautumisvaihtoehto

Tekijä: Laukkanen, Jarkko

Työn nimi: Web-osion rakentaminen SEAMK-opiskelijaportaaliin

Ohjaaja: Salmenjoki, Kimmo

Vuosi: 2010 Sivumäärä: 41 Liitteiden lukumäärä: 4

---

Seinäjoen ammattikorkeakoulussa on käytössä Microsoftin SharePoint-palvelut. SharePoint on tällä hetkellä vain henkilökunnan käytössä, mutta sen käytön laajentamista ollaan suunnittelemassa myös opiskelijoille. Kehitteillä olevaan opiskelijaportaaliin haluttiin saada uutta sisältöä. Tavoitteena oli rakentaa SharePoint-sivulla toimiva web-osio, joka ottaa yhteyttä opiskelijatietokantaan ja hakee sieltä tiedot opiskelijoiden suorituksista selattavaan taulukkoon.

Teoriaosassa tutkittiin hieman mikä SharePoint on, millaisia erillaisia versioita ja lisenssejä siitä on saatavilla, miten ne eroavat toisistaan ja mihin SharePointia voidaan käyttää. Kalliiden lisenssikustannuksien vuoksi pohdittiin hieman myös vaihtoehtoisia palveluita portaalien rakentamiseen.

Web-osion kehittäminen toteutettiin C#-kielellä. Ohjelmoinnissa käytettiin Microsoft Visual Studio 2008 -sovelluskehittäjä. Testiympäristönä toimi Microsoft Virtual PC -ohjelmalla luotu virtuaalikone. Virtuaalikoneen käyttöjärjestelmänä oli Windows Server 2008, johon oli asennettu SQL Server 2005 -tietokantaohjelmisto ja Microsoft Office SharePoint Server 2007 -ympäristö.

Rakennettu web-osio ottaa yhteyden tietokantapalvelimelle ja tuo sillä hetkellä sisään kirjautuneen opiskelijan tiedot taulukkonäkymään. Taulukkoon lisättiin muutamia asetuksia, joilla voidaan vaikuttaa taulukon esitystapaan ja ulkonäköön. Työn lopussa on lyhyt ohjeistus helpottamaan web-osion käyttöönottoa.

Asiasanat: SharePoint, Web-osio, Opiskelijaportaali

## SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

**Thesis abstract**

Faculty:	School of Technology	
Degree programme:	Business Information Technology	
Specialisation:	Application Software Production	
Author/s:	Laukkanen, Jarkko	
Title of the thesis:	Building the Web-part in the SeAMK Student Portal	
Supervisor(s):	Salmenjoki, Kimmo	
Year: 2010	Number of pages: 41	Number of appendices: 4

---

Seinäjoki University of Applied Sciences is currently using Microsoft's SharePoint -services. SharePoint is currently only used by the personnel, but plans are being made to extend the use of it to the students, as well. New content was sought into the student portal under development. The aim of the final project is to build a Web-part which works on the SharePoint-pages and connects to the student database and gets information about the students' grades in a table that can be browsed.

SharePoint was studied in the theoretical part of the study in order to find out information about, for example, what kind of different versions and licenses are available on it, what kind of differences they are having and where SharePoint can be used. Because of expensive license costs different kind of services available to build portals were also considered.

In the development of the web-part C#-language was used. Microsoft's Visual Studio 2008 -application generator was used for programming. A virtual machine created by Microsoft's Virtual PC -program worked as a test environment. The operating system in the virtual machine was Windows Server 2008, which had SQL server 2005 -database program and Microsoft's Office SharePoint Server 2007 -environment installed on it.

The designed Web-part connects to the database and displays student information of currently logged in users in a data table. A few custom-properties were added to the table, which makes it possible to change the outfit of the table and how the data are being displayed. At the end of this study there is a short guideline to make initialization of the web-part easier.

Keywords: SharePoint, Web-part, student portal

## SISÄLTÖ

### OPINNÄYTETYÖN TIIVISTELMÄ

### THESIS ABSTRACT

<b>SISÄLTÖ .....</b>	<b>4</b>
<b>KÄYTETYT TERMIT JA LYHENTEET .....</b>	<b>5</b>
<b>KUVIO- JA TAULUKKOLUETTELO.....</b>	<b>6</b>
<b>1 JOHDANTO .....</b>	<b>7</b>
1.1 Työn tausta.....	7
1.2 Työn tavoite.....	7
1.3 työn rakenne .....	7
<b>2 SHAREPOINTIN NYKYTILANNE SEAMKISSA.....</b>	<b>8</b>
2.1 WinhaWille .....	9
<b>3 SHAREPOINT OFFICE -PALVELUT .....</b>	<b>10</b>
3.1 Windows SharePoint Services 3.0 (WSS) .....	12
3.2 Microsoft Office SharePoint Server (MOSS) .....	13
3.3 Web-osiot .....	16
<b>4 SHAREPOINT-PALVELUT OPIKELIJAN NÄKÖKULMASTA....</b>	<b>18</b>
4.1 Muiden palveluiden vertailu .....	19
<b>5 TEKNINEN TOTEUTUS.....</b>	<b>23</b>
5.1 Web-osion luonti.....	24
5.2 Tietokantayhteyden muodostaminen .....	26
5.3 Custom-valikoiden luonti.....	30
5.4 Opiskelijatunnuksen hakeminen Active Directorysta.....	32
5.5 Ohje winhaWP-web-osion käyttöönottamiseen.....	34
<b>6 LOPUKSI .....</b>	<b>37</b>
<b>LÄHTEET .....</b>	<b>38</b>

## KÄYTETYT TERMIT JA LYHENTEET

<b>GAC-kansio</b>	(Global assembly cache) GAC-kansio sisältää DLL-tiedostoja .NET sovelluksille.
<b>DLL</b>	(Data link layer) DLL-tiedostot sisältää ajettavaa koodia.
<b>Bin-kansio</b>	Bin-kansio on tarkoitettu sisältämään käännettyjen kokoonpanojen tiedostoja ASP.NET sovelluksia varten esim. DLL-tiedostoja.
<b>Portaali</b>	Portaali on palvelu, joka tarjoaa pääsyn yhteen tai useampiin palveluihin.
<b>CMS</b>	(Content Management Server) on sisällönhallintajärjestelmä, jolla voidaan toteuttaa julkiset www-sivut tehokkaasti ja luotettavasti.
<b>SQL</b>	(Structured Query Language) standardoitu kyselykieli.
<b>XML</b>	(eXtensible Markup Language) merkintäkieli.
<b>WorkFlow</b>	Microsoftin kehittämä sovellus työnkulun hallintaa varten.
<b>Parametri</b>	Parametrit ohjelmoinnissa tarkoittaa funktiolle tai käskylle välitettäviä tietoja.
<b>MOSS</b>	Microsoft Office SharePoint Server.
<b>WSS</b>	Windows SharePoint Services.

## KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. MOSS ja WSS lisennöinti (Roine 2007, 113).....	12
Kuvio 2. MOSS 2007:n ominaisuudet lohkoittain. (Wallinger 2009.) .....	14
Kuvio 3. Googlen etusivu .....	19
Kuvio 4. Google wave (Google 2010c.).....	21
Kuvio 5. WinhaWP-web-osio perusasetuksilla .....	27
Kuvio 6. WinhaWP yksinkertaistetussa näkymässä .....	30
Kuvio 7. WinhaWP:n asetusvälilehti.....	31
Kuvio 8. SharePointin web-osio valikoima .....	35
Kuvio 9. WinhaWP web-osio listassa.....	35
Taulukko 1. WSS-version ja MOSS-version eroja (Gajendran 2008).....	11
Taulukko 2. Mahdolliset Custom propertyt (Microsoft 2010b).....	30

# 1 JOHDANTO

## 1.1 Työn tausta

Seinäjoen ammattikorkeakoulun Tekniikan yksikössä SharePoint on tällä hetkellä käytössä ainoastaan henkilökunnalla. Käytössä on SharePointin molempia versioita WSS ja MOSS, mutta WSS:n käytöstä ollaan vähitellen luopumassa. Opiskelijaportaali ei ole vielä käytössä, mutta sitä kehitetään SharePoint-ympäristöön.

## 1.2 Työn tavoite

Opinäytetyön aiheena on tehdä Seinäjoen ammattikorkeakoulun tekniikan-yksikön opiskelijaportaaliin web-osio, jonka tulee olla yhteydessä koulun opiskelijatietokantaan WinhaWilleen. Tietokannasta halutaan saada näkyviin opiskelijan suoritukset ja web-osion haluttaisiin ilmoittavan uusista suorituksista.

## 1.3 työn rakenne

Tässä dokumentissa selvitetään hieman mikä SharePoint on ja mihin sitä voidaan käyttää. SharePointin kalliiden lisenssimaksujen vuoksi pohditaan hieman vaihtoehtoisia palveluja. Dokumentin myöhemmässä osiossa käydään läpi web-osion luontia ja lopussa on pieni käyttöohje, joka on tehty järjestelmänhallitsijan näkökulmaa ajatellen helpottamaan web-osion käyttöönottoa.

## 2 SHAREPOINTIN NYKYTILANNE SEAMKISSA

Seinäjoen ammattikorkeakoululla on käytössään Microsoftin Sharepoint-palvelut, joita hyödynnetään eri yksiköissä eri tavoin. Erilaisten Microsoft-palveluiden integrointi Sharepointiin on mielekästä ja helppoa, koska kyseessä on Microsoftin tuote. Lisäksi Microsoft-työkalut ovat tuttuja henkilökunnalle. (Sivula 2010a.)

Sharepoint on koulukeskusyhtymällä käytössä hierarkkisessa rakenteessa. Jokaiselle yksikölle on luotu omat sivunsa, joihin on lisätty tarpeelliset toiminnot. Sharepointtia hyödynnetään myös työryhmäkäytössä (Sivula 2010a.)

Sharepoint on Seinäjoen ammattikorkeakoulussa on laajasti käytössä. Eniten SharePointista hyödynnetään tiedon jakamista sekä kalenteri- ja tapahtumatoimintoja. Kaikista Sharepointiin lisätyistä dokumenteista saadaan automaattinen versiointi. Tämä helpottaa henkilökunnan työtä sekä lisää historiatiedon saatavuutta, koska kaikista tiedostoista, joihin tehdään muutoksia, jää tallelle vanhemmat versiot. (Sivula 2010a.)

Helpdeskin SharePoint-sivut poikkeavat hieman muista koulun Sharepoint-sivustoista. Helpdesk-henkilön tehtävä on ratkoa erilaisia tietoteknisiä ongelmia, joita helpdesk-henkilö saa puhelimitse tai sähköisesti. Ongelmista luodaan aina vikailmoitustiketti, joka tarvittaessa välitetään eteenpäin yksikön atk-henkilöstölle RSS-syötteenä. Kun ongelma on selvitetty, kirjataan ratkaisu vikaselostukseen. Näin ratkaisu on tallella, mikäli vika toistuisi uudelleen. (Sivula 2010a.)

Opiskelijaportaali ei ole tällä hetkellä käytössä, koska se on kehitysvaiheessa. Tällä hetkellä opiskelijoilla on käytettävissään Seinäjoen ammattikorkeakoulun sisäinen Intranet. Intrassa ei olla hyödynnetty SharePointin toimintoja. Tämä tarkoittaa muun muassa sitä, että opiskelija ei voi vaikuttaa siihen sisältöön, mitä hänelle intrassa esitetään. Seinäjoen ammattikorkeakoululla on tavoitteena luoda opiskelijoille toimiva portaali, jonka sisältöä he voisivat itse muokata



haluamukseen. Tämä toteutettaisiin erilaisilla web-osoilla. Sharepointissa on paljon hyviä valmiita web-osia, mutta siitä ei löydy valmiiksi web-osoita, joita voitaisiin suoraan hyödyntää Seinäjoen ammattikorkeakoulun tarpeessa. Portaalissa pitäisi olla erilaisia ominaisuuksia, kuten esimerkiksi mahdollisuus seurata opiskelujen etenemistä, kuten ilmoittamalla opiskelijalle uusista suorituksista. (Sivula 2010.)

## 2.1 WinhaWille

Seinäjoen ammattikorkeakoulu käyttää tällä hetkellä WinhaWille-järjestelmää. Winha on web-pohjainen palvelu, joka sisältää tietoa opiskelijoista, opintosuunnitelman ja ilmoitustaulun. Winhaan kirjataan kaikkien kurssien suoritukset ja opiskelijat ilmoittautuvat kursseille sekä tekevät läsnäoloilmoituksen lukukaudelle Winhan kautta. (Seamk Willen käyttöohjeita.)

Hops, eli henkilökohtainen opintosuunnitelma on WinhaWilleissä jaettu neljään osaan; suunnitellut opinnot, ilmoittautumiset, osallistumiset ja suoritukset. Hops sisältää kaikki siihen kuuluvat kurssit. Winhawillen kautta voidaan kursseista katsoa lisätietoa sekä lisätä pakollisia ja vapaasti valittavia kursseja opintosuunnitelmaan. (Seamk Willen käyttöohjeita.)

WinhaWille-järjestelmä kehitti Novogroup, joka myöhemmin integroitui WM-dataan tämän ostettua sen. Vuonna 2006 VM-data myytiin kuitenkin Logicaan, eli nyt Logica on vastuussa WinhaWillestä.

Opinnäytetyön käytännön osuuden keskeisempiä vaatimuksia oli saada tietokantayhteys muodostettua Winha-järjestelmään. Työn tekoon ei saatu riittäviä käyttöäoikeuksia SQL-lauseiden rakentamista varten. Aluksi oletettiin, että SQL-lauseet voitaisiin ottaa Winhan raportointipuolelta, mutta tämäkin osoittautui myöhemmin hankalaksi tietoturvan vuoksi. Ratkaisuksi ongelmaan Seinäjoen ammattikorkeakoulun atk-suunnittelijalta saatiin valmis SQL-kysely, joka käsitteli opinnäytetyön tekijän henkilökohtaisia opintosuorituksia. Näiden tietojen perusteella toteutuksen tekeminen oli mahdollista. (Sivula 2010b.)

### 3 SHAREPOINT OFFICE -PALVELUT

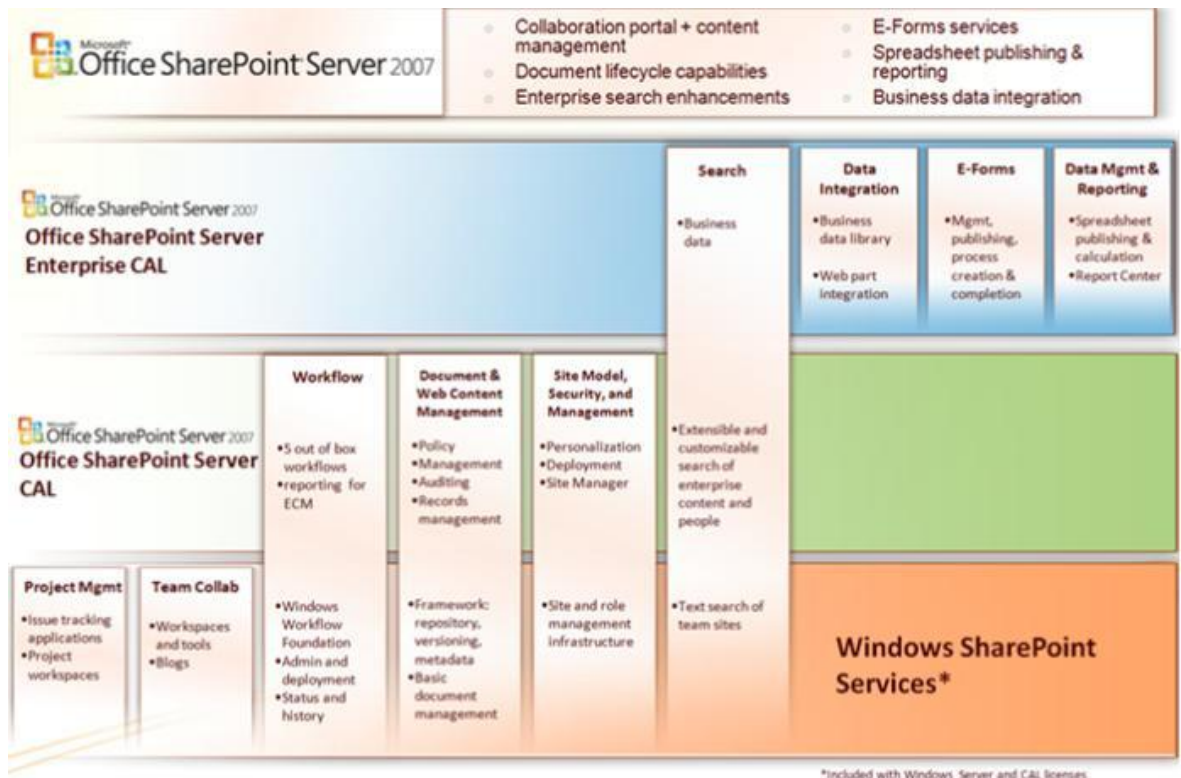
Tässä kappaleessa selvitetään hieman mikä SharePoint on, mitä sillä voi tehdä, millaisia toimintoja siitä löytyy ja selvitetään hieman millaisia lisenssivaihtoehtoja SharePointista on saatavilla.

SharePoint on verkkopohjainen työkalu, jonka avulla voidaan luoda erilaisia web-pohjaisia palveluita kuten intranet, extranet, sekä portaaleja erilaisiin tarkoituksiin. Nimensä mukaisesti SharePointin pääpaino on tiedostojen jakamisessa ja tämän kautta helpottaa työorganisaatoiden yhteistyötä, sekä kommunikointia. Sovelluksen avulla esimerkiksi organisaation projektin tiedostot ovat kaikkien henkilöiden ulottuvilla vaivattomasti. SharePoint tekee tiedostoista automaattista versiointia. Kaikista tiedostoista, joihin tehdään muutoksia jää vanhemmat versiot tallelle. Tämä vähentää huomattavasti tiedonsäilytykseen liittyviä riskejä. SharePointilla on mahdollista luoda edellämainittuja palveluita ilman koodaustaitoja. Tarjolla on lukuisia valmiiksi rakennettuja mallipohjia, joita on helppo konfiguroida asetusvalikkojen kautta. Mikäli valmiista ratkaisusta ei ole mahdollista tuottaa halutunlaista lopputulosta, on mahdollista koodata omaa sisältöä esimerkiksi web-osioiden muodossa. Räätelöityjä ratkaisuja voidaan ostaa myös kolmannilta osapuolilta. MS SharePoint-office -tuoteperheestä on saatavilla kaksi versiota WSS ja MOSS. Lisätietoja MOSS-version ja WSS-version eroista löytyy liitteestä 1. (Murphy & Perran 2007, 3.)

Taulukossa 1 selvitetään WSS-version ja MOSS-version eroja.

Taulukko 1. WSS-version ja MOSS-version eroja (Gajendran 2008)

<b>WSS</b>	<b>MOSS</b>
Saatavilla ilmaiseksi Windows Server 2003:sen kanssa	On maksettava serveri- ja -CAL -lisenssit
Tarjoaa valmiit mallipohjat erillaisten sivujen ja blogien rakentamiseen	Käytetään myös todella tehokkaana sisällönhallinta työkaluna
Sivuihin voidaan liittää Access 2007, Exel 2007, Outlook 2007, Word 2007 ja PowerPoint 2007 – palveluja	MOSSiin voidaan liittää myös Workflow ja paljon muista sovelluksia
Site Collectionin luonti tarjoaa minimaalisen määrän eri mallipohjia kuten Collaboration ja Meetings	Site Collectionin luonti tarjoaa enemmän valmiita mallipohjia kuten Collaboration, Meetings, Enterprise, Custom and Publishing



Kuvio 1. MOSS ja WSS lisennöinti (Roine 2007, 113)

Kuvio 1 on tarkoitettu selvittämään lisensiointiä, mutta siitä näkee hyvin myös kuinka MOSS rakentuu ja millaisia eroja sillä on Windows SharePoint Serviceen. Kuvion alimmalla tasolla on Windows SharePoint Services ja siitä edetään pykälittäin MOSS CAL- ja Enterprise CAL -versioon, jotka sisältävät myös kaikki alemmilla portilla olevat ominaisuudet.

### 3.1 Windows SharePoint Services 3.0 (WSS)

Windows SharePoint Servicestä löytyy dokumentin hallinnan ja yhteistyöalustojen ydintoiminnot. WSS:llä on siis mahdollista tuottaa intranet- ja ekstranet palveluita, sekä julkisia www-sivustoja helposti. Valmiiden mallipohjien ansiosta koodaustaidot eivät ole välttämättömiä. Suurin etu WSS-versiossa on sen ilmaisuus. WSS voidaan asentaa Windows Server 2003- tai Small Business Server 2003 -käyttöjärjestelmiin. (Murphy & Perran 2007, 3, 5.)

Web-osiot toimivat sekä WSS-versiossa, että MOSS-versiossa. Heikkouksia löytyy kuitenkin mm. tiedon haun ja sen indeksoinnin osalta. Näitä puutteita voidaan kuitenkin korjata yhdistämällä Microsoft-palvelujen eri lisenssejä. Esimerkiksi yhdistämällä CMS + WSS saataisiin käyttöön paljon järeämpi extranet-sivusto. Tämä tietenkin lisää kustannuksia huomattavasti, joten organisaatioiden tulee harkita tarkkaan mitä lisenssejä todellakin tarvitaan. (Roine 2007, 33,45.)

WSS on yksittäisille käyttäjille ilmainen, mutta yrityksillä täytyy olla Windows-lisenssi ja erikseen hankittava Windows-Connector-lisenssi. WSS:n mukana tulee kevyt WMSDE-tietokanta, sen käyttö ei kuitenkaan ole kovin suotavaa isoille organisaatioille, sillä se sisältää paljon koko- ja palvelinrajoituksia SQL Server 2005:een verrattuna. WSS:n on kuitenkin mahdollista kytkeä myös erillinen SQL-palvelin, sen myötä saadaan myös paremmat hakutoiminnot käyttöön. (Roine 2007 18.)

### **3.2 Microsoft Office SharePoint Server (MOSS)**

Kun yhdistetään kaksi eri SharePoint-tuotetta samalle alustalle, Office Sharepoint Server 2007 ja Windows SharePoint Services 3.0, syntyy siitä MOSS 2007 (Roine 2007, 94). MOSS 2007 on siis uudempi versio SharePoint Portal Server 2003:sta.

Ulkoisesti MOSS 2007 näyttää samalta kuin WSS, valikot ja toiminnot löytyvät samoista paikoista. Tuotteiden toiminnallisuuksissa on kuitenkin suuria eroja. MOSS perii kaikki ominaisuudet mitä WSS pitää sisällään, sekä lisää siihen uusia tai kehittyneempiä vaativampaan käyttöön tarkoitettuja toimintoja (Kuvio 1). Näistä tärkeimmät ovat kehittyneemmät hakutoiminnot, hakujen indeksointi ja Excel-palvelut. MOSS:n mukana tulee enemmän mallipohjia sekä valmiita web-osiota kuin WSS:ssa. Miinuspuolena ovat kuitenkin kalliit lisenssi-maksut. MOSSin mukana ei tule minkäänlaista tietokantapalvelinta, joten se vaatii toimiakseen SQL-serverin. Lisenssikustannuksia voi karsia hankkimalla ilmaisen SQL-Server Expressin. Tuote on ilmainen, mutta se sisältää erittäin suuria rajoituksia, minkä takia sitä ei voi suositella isoille organisaatioille, joiden tietokannassa liikkuu paljon tietoa. (Roine 2007, 114.)

Jokaiseen palvelimeen, johon MOSS asennetaan, on hankittava tuhannen euron hintainen palvelinlisenssi. Jokaiselle henkilölle, joka käyttää MOSSia, on hankittava CAL-lisenssi. Core CAL -lisenssi on tarkoitettu peruskäyttöön ja Entepriice CAL suuren yrityksen tarpeisiin. Jos MOSS-palvelu ulkoistetaan, on organisaation ulkopuolelta palvelua käyttäville henkilöille hankittava tarkoitukseen sopiva CAL-lisenssi. Julkisia palveluita varten järkevin vaihtoehto on hankkia kymmenien tuhansien eurojen hintainen SharePoint Connector for Internet Sites -lisenssi. Lisäksi Microsoft Office Search Server 2007, Forms Server, Project Server, Groove Server ja Portfolio Server vaativat oman lisenssinsä, jos näitä halutaan käyttää. (Roine 2007, 114-116.) Lisätietoa MOSS 2007 lisensoinnista on liitteessä 2.

Kuvassa 2 MOSS 2007 ominaisuudet on lohkottuna kukin omalle osa-alueelleen.



Kuvio 2. MOSS 2007:n ominaisuudet lohkoittain. (Wallinger 2009.)

Collaboration tarkoittaa yhteistyötä. MOSS tarjoaa käyttäjilleen turvallisen ympäristön luoda, selata ja julkaista tietoa verkossa niin, että se voidaan jakaa helposti muiden käyttäjien kanssa. Käyttäjät voivat lisätä dokumenttejaan useisiin erillaisiin kirjastoihin kuten dokumentti- ja kuvakirjastoihin. MOSS tarjoaa

käyttäjille työtilan (Workspaces), jossa käyttäjät voivat työstää yhdessä samaa dokumenttia kunnes se on valmis julkaistavaksi. Käyttäjien on mahdollista kommunikoida mm. blogien, wikien, RSS-syötteiden ja keskustelufoorumeiden välityksellä, kun taas jotkin työkalut on tarkoitettu projektin hallintaa varten kuten työlista, kalenteri ja ongelmalista. (Wallinger 2009.)

Portaalilla tarkoitetaan verkkosivuja, joka tarjoaa omien toimintojensa lisäksi pääsyä myös muihin sivuihin ja palveluihin. MOSS on suunniteltu toimimaan muun Microsoft Office -tuoteperheen kanssa, jotenka kaikki kalenterimerkinnät sekä työtehtävät voidaan liittää Outlookiin. MOSSIin on sisään rakennettu Excel, jottei käyttäjällä, joka haluaa tarkastella SharePointilla olevia Excel-tiedostoja, tarvitsisi olla omalla päätteellään Exceliä asennettuna. Halutessaan luoda oman henkilökohtaisen SharePoint-sivun käyttäjällä on mahdollisuus tehdä se My sites -toimintoa hyväksikäyttäen. (Wallinger 2009.)

Hakutoimintoa on pyritty kehittämään paljon MOSSissa, sillä ei ole ainoastaan mahdollista hakea tieto omasta SharePoint-ympäristöstä, vaan se voidaan myös konfiguroida hakemaan tietoa muilta sivuilta. Muita hakumahdollisuuksia ovat mm. ihmiset (käyttäjäprofiilit), SharePoint-dokumenttikirjastot, sähköposti, jopa SQL- tai Oracle-tietokantapalvelimet voidaan määritellä niin, että niistä voidaan hakea tietoa. Tietoturvasyistä haku ei näytä sivuja, joihin haun tekijällä ei ole käyttöoikeutta. (Wallinger 2009.)

Sisällönhallinta (Content Management) on MOSSissa jaettu kahteen kategoriaan, dokumentin hallintaan ja web-sisällön hallintaan. Valmiita dokumentteja voidaan ladata SharePoint-listoihin tai uusia dokumentteja voidaan luoda valmiiden SharePoint-mallipohjien avulla. Metadataa voidaan kerätä kaikkien liitetiedostojen sisällön tyyppinä (Content Type) hyödyntämällä. (Wallinger 2009.)

Business Process Management (BPM) osassa hyödynnetään lähinnä WorkFlow- ja InfoPath-työkaluja. Esimerkkinä tästä ominaisuudesta on työntekijä, joka täyttää loma-anomuksen SharePoint-palvelussa, jonka WorkFlow kuljettaa työnantajalle. Työnantajan joko hyväksyttyä tai hylättyä anomuksen työntekijälle välittyy siitä viesti. (Wallinger 2009.)

MOSSissa on valmiita mallipohjia, joiden avulla voidaan rakentaa sivustoja, joihin kerätään tietoa yrityksestä. Sivulla on mahdollista esittää esimerkiksi erillaisia raportteja sekä Excel-taulukoita ja kuvioita. (Wallinger 2009.)

Kuviossa 2 keskus, Platform Services huolehtii turvallisuudesta ja kaikkien erillisten SharePoint-sovellusten kuvauksesta, niin että kaikki näyttäisi olevan yhtä ja samaa tuotetta. (Wallinger 2009.)

### **3.3 Web-osiot**

Web-osiot muistuttaa Windows Vistan widgettejä. Ne ovat erikseen ohjelmoituja toiminnallisuuksia, jotka voidaan raahata niille varattuihin kohtiin portaalia. Käyttäjä saa siis itse määrätä sivustojen toiminnallisuuksien järjestyksen ja koon. Web-osioiden muodostuu myös automaattisesti esimerkiksi kun palveluun luodaan uusi asiakirjakirjasto. SharePoint lisää tällöin web-osion joka esittää asiakirjakirjaston sisältöä, näin ollen käyttäjät löytävät sen helposti eikä se pääse uppoutumaan SharePointin syövereihin. MOSSin mukana toimitetaan n. 70 web-osiota, joista useimmat ovat periaatteeltaan hyvin samankaltaisia. Valmiita web-osioiden käytetään usein ainoastaan pohjana uutta toimintoa luodessa. (Roine 2007, 151-153.)

Web-osioiden luodessa tai muokatessa käytetään usein Microsoft Visual Studio-sovelluskehittäjä C#- tai Visual Basic -ohjelmointikielillä (Roine 2007, 153). Visual Studioon on mahdollista ladata Microsoftin sivuilta Visual Studio 2008 Extensions for Windows SharePoint Services 3.0 -laajennuspaketti. Paketin asentamalla Visual Studioon saadaan Web-parts -template. Template luo valmiin mallipohjan web-osiosta, sekä luo web-osion XML-kuvaustiedoston automaattisesti. Paketti voidaan kuitenkin asentaa vain Windows Server 2003- tai 2008 -käyttöjärjestelmiin. Vastaavanlaisia paketteja löytyy internetistä muitakin (epävirallisia). Template-pakettien käyttäminen ei ole välttämätöntä, mutta niistä saattaa olla suuri apu kokemattomalle sovelluskehittäjälle. (Microsoft 2010c.)



Tämän opinäytetyön tavoitteena on luoda Seinäjoen ammattikorkeakoulun Tekniikan yksikölle SharePointissa toimiva web-osio, joka ottaa yhteyden koulun opiskelija-tietokantaan ja hakee sieltä tiedot suorituksista.

WinhaWille hakee suorituksista tietoa mm. seuraavista kentistä; Opinnon nimi ja tunnus, ylempi opinto, laajuus, Vir (virtuaalisuus), T&K (tutkimuksellisuus), Arv (arviointi) Arvioija, arviointipvm ja Huom (huomioitavaa). Käyttäjä pääsee muokkaamaan web-osioiden asetuksia SharePoint portaali-sivun kautta web-osion lisäämisen jälkeen, näihin asetuksiin on mahdollista lisätä ohjelmoitaessa omia määrittämiä. Web-osiota suunniteltaessa harkittiin erillaisia asetuksia, kuten valintamahdollisuuden yksinkertaistetulle näkymälle. Kun tämä olisi valittuna, aktivoituisi yksinkertaistettu SQL-lause, joka hakisi tietoa ainoastaan kaikista tärkeimmistä kentistä kuten opinnon nimi, arviointi ja arviointipvm. Tämä saattaisi rasittaa vähemmän hakumootoria, jolloin taulukko latautuisi hieman nopeammin ja web-osion sisältämästä taulukosta tulisi samalla siistimmän näköinen. Opiskelija näkisi helposti onko hän päässyt kurssista läpi.

Toinen harkittu lisäasetus määrittäisi kuinka monta tietuetta näytettäisiin per sivu. Esimerkiksi jos asetukseen määritettäisiin viisi, taulukossa näytetään vain viisi tietuetta ja loput on selattavissa eri sivuilta 1, 2, 3, 4, 5. Koska taulukko hakisi vain pienen osan tietoa kerralla, se latautuisi nopeammin, eikä pitkä taulukko ahmisi koko portaalisivua. Opiskelija saattaisi muutenkin olla kiinnostunut esimerkiksi vain viidestä uusimmasta suorituksesta.

Kolmas web-osioon suunniteltu lisäasetus saattaisi olla hieman turha mikäli edellä mainittu sivuamisasetus toteutetaan. Eräs asetumahdollisuus voisi olla kuitenkin, että halutessaan käyttäjä voisi määrittää taulukkoon haettavaksi tiedot vain uusimmasta suorituksesta tai viidestä uusimmasta. Halutessaan käyttäjä voi lisätä portaalisivulleen useammankin suoritustaulukko web-osion, joissa olisi eri asetukset määriteltynä. Esimerkiksi käyttäjällä voisi olla sivun yläkulmassa yksinkertaistettuna taulukkona tieto ainoastaan uusimmasta suorituksesta ja jossain muualla olisi sitten hieman tarkempi taulukko suorituksista.

## 4 SHAREPOINT-PALVELUT OPISKELIJAN NÄKÖKULMASTA

Opiskelijoilla on usein tarvetta paremalle tiedostonjakamiselle esimerkiksi ryhmätöissä. Nykytilanne on sellainen, että ryhmätyön tiedostontallennus jää usein yhden opiskelijan vastuulle. Hän tallentaa ryhmätyön tiedostot omalle henkilökohtaisille verkkoasemalle, johon ei ole kellään muulla pääsyä. Joskus kuitenkin käy niin, että henkilö, jonka verkkoasemalle tiedostot on tallennettu, ei pääsekkään paikalle, eikä muille opiskelijoille jää muuta vaihtoehtoa kun lähteä nyrpeänä takaisin kotiin. Olisi muutenkin hyödyllistä jos tiedostoihin päästäisiin käsiksi myös muualta kuin koulun tietokoneilta. Ongelmaa esiintyy varsinkin suurissa ryhmätöissä, joissa tiedonjako olisi tärkeätä. Ongelma pystyttäisiin ratkaisemaan SharePointin tiedostonjaolla, kaikilla olisi pääsy tiedostoihin ja kuka tahansa ryhmänjäsen pystyisi tekemään niihin muutoksia. Kaikista muutoksista SharePoint hoitaisi automaattisesti versioinnin, eli vanhatkin tiedostot jäävät tallelle. Ehkä olisi mahdollista myös perustaa ryhmiä koko luokalle, tätä kautta opettajat voisivat jakaa materiaalia opiskelijoille, mistä ilmoitettaisiin opiskelijaportaalin pääsivulla. Tämä korvaisi Moodlen ainakin osittain.

SharePointin keskusteluryhmät ja blogit sen sijaan saattaisivat parantaa etätyöskentely mahdollisuuksia ja helpottaa kommunikaatiota opiskelijoiden välillä. Varsinkin pikaviestin, joihin opiskelijat saisivat rakentaa keskusteluryhmiä, saattaisi olla hyödyksi joissain koulutehtävissä.

Toinen asia mitä opiskelijaportaaliin voitaisiin tuoda on kursseille ilmoittautuminen. Mutta valitettavasti se ei ole mahdollista WinhaWilleen asetettujen kieltojen takia. Jos WinhaWillen tietokantoihin olisi laajemmat käyttöoikeudet koko WinhaWillen käyttö voitaisiin korvata SharePoint puolella ehkä hieman selkeämällä käyttöliittymällä, kuin mitä sillä nyt on.

Näin saataisiin kaikki opiskelijan tarvitsemat toiminnot yhteen paikkaan. Yksi web-osoista voisikin olla linkkilista, johon olisi vakiona määritelty kaikki tarpeelliset toiminnot, mutta käyttäjällä itsellään olisi myös mahdollista lisätä siihen omia linkkejä. Lisäarvoa SharePoint-palveluille tuo pääsy mobiililaitteilla.

## 4.1 Muiden palveluiden vertailu

SharePoint-tuoteperheen kalliiden lisenssimaksujen vuoksi on kuitenkin syytä miettiä myös vaihtoehtoisia ratkaisuja. Vertailukohdaksi valittiin Googlen palvelut, koska sen käyttäminen vaatii ainoastaan gmail-tilin. Koska Google-palvelut ovat web-pohjaisia ja toimii internetissä valmiilla palvelimilla, tietokoneelle ei tarvitse asentaa mitään lisäohjelmia. Googlella on useita uusia palveluja kuten iGoogle, Google sivustot ja Google Wave. Nämä palvelut muistuttavat hyvin paljon SharePointin tarjoamia palveluja.

iGoogle on Googlen tarjoama uusi etusivu, joka toimii portaalina muihin internetpalveluihin. iGoogleen voidaan lisätä sisältöä erillaisten gadgettien kautta. Gadgetit vastaavat SharePointin web-osioita. Gadgetit ovat HTML- ja JavaScript -sovelluksia, joita voidaan lisätä, muokata ja poistaa oman halun mukaisesti. (Google 2010a.)

The screenshot shows the iGoogle homepage with a search bar at the top and several widgets. The search bar contains the text "iGoogle" and "Tarkennettu haku" and "Kokkeilun onneani". Below the search bar, there are buttons for "Google-haku" and "Kokkeilun onneani". The main content area is divided into several sections:

- Tietoviikko**: A section for news and updates.
- Telkku.com TV-opas**: A TV schedule widget showing programs like YLE TV1, YLE TV2, and MTV3.
- Reittiopas**: A travel planning widget with fields for "Mistä:" and "Mihin:" and a "Hae" button.
- Talentum.com- uutiset**: A news widget with links to "Jenkkidata jäi odotuksista", "Mainosmies vie rahat Ruotsissa", and "Häroöksintö houkuttaa mediataloja".
- Urheiluportaali Sportti.com**: A sports news widget with links to "Messi jätti Beckhamin taakseen", "Ferrari-pomo vaatii nöyryyttä", "Vain Selänne ja Jokinen varmuudella ulkona", "Vancouver ei verota MM-kisojen yleisömäärää", and "Jalonen harmittelee SM-liigan aikataulun venymistä".
- Gmail (29)**: A Gmail widget showing a list of emails from "Postilaatikko (29)", "x10Hosting", "Bidwii.com", "noreply (2)", and "Funcom".
- Saa**: A weather widget showing a sun icon and the text "Saat kotiseutusi sääennusteen ja haluamiesi kohteiden ennusteita ympäri maailmaa."

Kuvio 3. Googlen etusivu

Kuva 3 on otettu iGooglen etusivusta. iGoogle koostuu pääosin gadgeteista. Useimmat kotimaiset gadgetit ovat tunnettujen organisaatioiden luomia kuten esimerkiksi Itasanoma ja telkku.com, tarjolla on kuitenkin useita erillisiä palveluita

ja toimintoja. Sivun vasemmalla laidalla on linkkilista, josta klikkaamalla kyseinen palvelu aukeaa iGoogleen koko sivun suuruisena. Googlessa on oma pikaviestin, johon voidaan lisätä kavereita sähköpostiosoitteen perusteella. Pikaviestin-sovellus löytyy myös iGooglen etusivulta. (Google 2010a.)

Google-sivustot muistuttavat SharePointin sivustoja. Google-sivujen luonti on nopeata ja vaivatonta. Tarjolla on useita valmiita mallipohjia, kuten esimerkiksi extranet, wiki, club ja projekti. Sivuille voidaan lisätä mm. kuvia, tekstiä, videoita, kalentereita, laskentataulukkoa, gadgettejä ja tietoarkistoja, joihin käyttäjät voivat lisätä tekemiään dokumentteja. Jopa uusimpien tiedostomuutosten seuraaminen on mahdollista gadgettien avulla. (Google 2010b.)

Googlen Wave on ryhmätyösovellus, joka yhdistää sähköpostin, pikaviestimen, wikin ja sosiaalisen median toimintaperiaatteita. Waven kehittäjät kuvaavat palvelua sähköpostin seuraavaksi sukupolveksi. Viestien sijaan palvelulla luodaan ”aaltoja” joihin voidaan liittää uusia käyttäjiä missä tahansa vaiheessa. Kuka tahansa osallistuja voi lisätä, muokata ja kommentoida ”aallolla” olevaa tietoa. Wave-viestiin osallistuneiden henkilöiden on mahdollista kirjoittaa viestille samanaikaisesti ja kaikkien kirjoitus on näkyvässä reaaliaikaisesti jokaisen näyttöpäätteellä. Waven on kerrottu julkaistavan avoimena protokollana, jolloin kuka tahansa voi halutessaan perustaa Wave-palvelimen. Toistaiseksi Wave-palveluun pääsee tutustumaan ainoastaan kutsun saaneet henkilöt. (Google 2010c.)



Kuvio 4. Google wave (Google 2010c.)

Kuten kuvasta 4 voi nähdä, Google waven näkymä on jaettu kolmeen osioon. Vasemmassa reunassa on työkalu, yhteystieto- ja navigointi -palkki. Palkin vieressä on kenttä jossa esitetään kaikki "aallot". Oikeanpuoleisessa alueessa tapahtuu kaikki aaltoihin liittyvät toiminnot kuten, lukeminen, kirjoittaminen ja uusien aaltojen luonti. Kaikki kolme osiota on mahdollista skaalata haluamansa kokoisiksi.

Google tarjoaa hyvin samankaltaisia palveluita kuin SharePoint, mutta Googlen palvelut ei kuitenkaan sovellu SEAMK tekniikan-yksikössä viralliseen käyttöön. Googlen palvelinten sijainnista ei ole mitään tietoa joten vaara, että palveluun tallennetut tiedot katoaa on suurempi kuin SharePointilla. Vaikka SharePoint on suunniteltu yrityksen tarpeisiin on siinä tiedonjakaminen yms. palvelut selkeämmin ja helppokäyttöisemmin esillä, sitä vartenhan koko tuote on suunniteltu. Portaalin perustaminen MOSS 2007:ssa vaatii enemmän työtä kuin Googella, mutta tämä on järjestelmänvalvojen työtä, eikä varsinaisille käyttäjille tule sen asian suhteen ongelmia. Googlen käyttö on kuitenkin ilmaista ja sitä voisi suositella opiskelijoille epävirallisena ratkaisuna helpottaa ryhmätöiden tiedostonjakoa. Esimerkiksi opiskelijat voisivat perustaa keskenään oman googletilin jotain kurssia varten ja

luoda sinne sivun jonka kautta he jakavat tiedostoja kaikille ryhmätyöhön osallistuneille henkilöille. Kunhan Google Wave-palvelu julkaistaan kaikkien saataville, saattaisi siinä olla mielenkiintoisia ominaisuuksia ryhmätöitä ajatellen. Kuvittele kolmen hengen ryhmä, jonka tehtävänä on kirjoittaa dokumentti jostain aiheesta. Sen sijaan että kaikki kolme työskentelisivät saman näytön äärellä, jokainen on omalla näyttöpäätteellään kirjoittamassa jotain osaa työstä. Samalla kun he kirjoittavat omaa osuuttaan, näkisivät he reaaliajassa miten muiden työ edistyy. Kuka tahansa heistä voisi muuttaa mitä tahansa työn osaa tai kirjoittaa kommenttiin muutosehdotuksia. Työstä tulisi yhdenmukaisempi vaikka sen olisi kirjoittanut kolme eri henkilöä kotoaan käsin. Kun työ on valmis, aallon sisältämät tekstit voitaisiin kopioida esimerkiksi Word-asiakirjaan.

## 5 TEKNINEN TOTEUTUS

Testiympäristön pystyttämiseksi ensiksi asennettiin koneelle Microsoft Virtual PC-ohjelma, jolla virtuaalinen palvelin emuloitiin. Käyttöjärjestelmäksi valittiin Windows Server 2003, koska se oli windows-palvelimista tutuin. Windows palvelimeen asennettiin MOSS 2007. MOSS valittiin, koska siinä on enemmän toimintoja ja valmiita mallipohjia. MOSS ei voi toimia ilman tietokanta-serveriä, joten täytyi palvelimelle asentaa myös tietokanta-palvelin. Tietokantapalvelimeksi valittiin Microsoft SQL-server 2005:sen, koska se tuntui helpommalta konfiguroida ja asentaa kuin SQL-server 2008. Web-osioon tarvittavan koodin tehtiin Microsoft Visual Studio 2008-sovelluskehittimellä, joka asennettiin myös palvelin-koneeseen, että käyttöön saataisiin Microsoftin virallinen web-osio laajennuspaketti. Myöhemmin myös WSS asennettiin eri tietokoneeseen, että voitaisiin verrata eroja MOSSIin.

Opinäytetyössä tehty web-osio ottaa yhteyden tietokanta-palvelimeen ja hakee sieltä tiedot opiskelija suorituksista DataGridille. DataGrid on ohjelmoinnissa listaus-kontrolli, jonka avulla voidaan esittää tietolähteen sisältämä tieto taulukkomuodossa. DataGridiin lisättiin ominaisuus, jolla voi määrittää kuinka monta tietuetta esitetään per/sivu. Alunperin koodi oli kirjoitettu niin, että tietokantayhteys asetukset täytyy määrittää SharePointin puolella web-osion ominaisuuksista ennen yhteyden muodostamista. Ominaisuudesta kuitenkin luovuttiin, sillä tässä tapauksessa on parempi jos tietokantayhteys on ainoastaan kovakoodattu web-osion koodiin. Myös luvussa 3.3 Web-osiot mainitussa ominaisuudesta luovuttiin, jonka aktivoimalla haettaisiin tieto ainoastaan uusimmasta suorituksista. Kyseisen toiminnon korvasi sivuamis-ominaisuus, jolla määritetään kuinka monta tietuetta esitetään yhdellä sivulla.

## 5.1 Web-osion luonti

Web-osioiden luonnin helpottamiseksi asennettiin Microsoft Visual Studio myös palvelinkoneeseen, näin ollen käyttöön saatiin Visual Studio 2008 extensions for Windows SharePoint Services 3.0 -laajennuspaketti. Laajennuspaketti tuo tullessaan Visual Studioon web-osio -templaten, joka luo perus "hello world" web-osion automaattisesti uuden projektin luonnin yhteydessä. Templaten käyttäminen ei ole välttämätöntä web-osioita rakentaessa, mutta se huolehtii alkumäärittämisistä jotka olisi muuten tehtävä manuaalisesti, kuten vahvan avaimen luonnista, jota joissain tapauksissa tarvitaan web-osion tunnistamiseen SharePointin puolella. Laajennuspaketti luo Microsoft.Sharepoint-viittauksen automaattisesti joka on välttämätöntä web-osioita luodessa. AssemblyInfo.cs rakentuu myös automaattisesti. AssemblyInfo pitää sisällään rakennettavan web-osion määrittämiä, laajennuspaketin mallipohja määrittelee sinne automaattisesti versionumeroksi 1.0.0.0-version. Versionumero on syytä määrittää manuaalisesti, sillä muuten joka kerran kuin web-osion koodi käännetään, tulisi sille eri versio. Joka tarkoittaisi sitä että web.config-tiedostoon jouduttaisiin joka kääntö kerralla muokkaamaan oikea web-osion versionumero, että se voitaisiin tunnistaa myös SharePointin puolella. Ennen kuin SharePoint hyväksyi web-osioni jouduin lisäämään AssemblyInfoon [assembly: AllowPartiallyTrustedCallersAttribute]-määrittäksen.

Kun web-osio käännetään siitä syntyy DLL-tiedosto, joka sisältää kaiken tehdyn koodin. Jotta koodi saataisiin näkymään SharePoint-palvelimella sen käyttöön ottamista varten DLL-tiedosto on sijoitettava oikeaan kansioon. Mahdollisia sijainteja on kaksi: Global Assembly Cache (GAC) tai WSS-sivun bin-kansio. Kansion valintaan liittyy lähinnä tietoturva seikkoja. GAC-kansiossa, joka sijaitsee Windows Server 2003 -käyttöjärjestelmässä polun C:\WINDOWS\assembly takana, .dll-tiedostoilla on täydet oikeudet toimia, eikä niiden oikeuksia tarvitse määrittää erikseen missään muualla (Microsoft, 2010a). Kun taas bin-kansioon sijoitetuilla tiedostoilla on oltava koodissa määriteltynä riittävät oikeudet toimiakseen. Bin-kansio sijaitsee yleensä polun C:\Inetpub\wwwroot\wss\VirtualDirectories\[PortinNumero]\bin takana. Tästä syystä sovelluksenkehittäjät suosittelivat, että web-osion DLL-



tiedosto olisi sijoitettuna bin-kansioon. GAC-kansion käyttö on kuitenkin aloittelijalle helpompi. GAC:ssa olevilla tiedostoilla on oltava vahva avain (Strong key). Strong key on ohjelmoinnissa käytettävä nimeämiskäytäntö (Roth 2008.)

Kun DLL-tiedosto on lisätty oikeaan kansioon, täytyy web.config-tiedostoon lisätä web-osion tiedot SaveControlliin, jotta se voitaisiin näyttää SharePointin puolella. Alla olevassa esimerkissä on SaveControl-määrytykset tehtynä WinhaWP-web-osiolle.

Web-config-tiedosto löytyy polun  
 C:\Inetpub\wwwroot\wss\VirtualDirectories\[PortinNumero]\  
 takaa.

#### Esimerkki 1.

```
<SafeControls>
<SafeControl Assembly="WinhaWP, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=9f4da00116c38ec5" Namespace="WinhaWP"
TypeName="*" Safe="true" />
```

Toinen mahdollinen tapa lisätä web-osio SharePoint-serverille on lisätä se XML-tiedoston avulla. Nämä ovat muotoa .dwp (Windows Sharepoint Services 3.0) tai .webpart (ASP. NET 2.0). Dwp-tiedostontyyppi on vanhentunut, mutta sitä yhä tuetaan SharePointissa (Roth, 22.10.2008). XML-tiedostossa kuvataan DLL-tiedosto SharePointille ja kerrotaan missä se sijaitsee. DLL-tiedoston on joka tapauksessa oltava oikeassa kansiossa ja web.config-tiedostoon on tehtävä SaveControl-määrytykset. (English 2007, 13)

Esimerkki 2 on XML-tiedostosta, jonka SharePoint loi itsestään web-osion käyttöönottamisen jälkeen.

### Esimerkki 2.

```
<?xml version="1.0" encoding="utf-8"?>
<webParts>
  <webPart xmlns="http://schemas.microsoft.com/WebPart/v3">
    <metaData>
      <!--
        The following Guid is used as a reference to the web part class,
        and it will be automatically replaced with actual type name at
        deployment time.
      -->
      <type name="be07f994-3c9e-47e2-9ba8-dd0a14c4b493" />
      <importErrorMessage>Cannot import WebPart1 Web
Part.</importErrorMessage>
    </metaData>
    <data>
      <properties>
        <property name="Title" type="string">WebPart1 Web Part</property>
        <property name="Description" type="string">WebPart1
Description</property>
      </properties>
    </data>
  </webPart>
</webParts>
```

## 5.2 Tietokantayhteyden muodostaminen

Keskeisempiä vaatimuksia web-osiolle oli, että sen täytyy ottaa tietokantayhteys koulun opiskelijatietokantaan, johon on kirjattu tiedot opiskelijoista. Ennen varsinaisen tietokantayhteyden rakentamisen aloittamista luotiin uusi luokka nimeltään DBAcces.cs, jota käytetään apuna tietokantayhteyttä muodostaessa. DBAcces-luokassa luodaan StrConnection-olio, sekä GetSQIResult-methodi, joka palauttaa DataTableen ja käsittelee SQL-lauseen parametrinä. Itse tietokantayhteys määritellään web-osion koodissa.

Esimerkeissä olevan koodin tietokantayhteys on määritetty testipalvelimeen. Palvelimen tietokantaan on lisätty tekijän henkilökohtaiset opiskelijatiedot, jotka saatiin koulun järjestelmänhallitsijalta Excel-tilukossa. Tieto ei siirtynyt palvelimelle täydellisesti, arviointi kentän tiedot jäivät puuttumaan suurimmilta

osin. Tätä ei kuitenkaan pidetty ongelmana, sillä arviointikenttä ei sisällä web-osion kehittämisen kannalta tärkeitä tietoja. Kun toteutus siirretään koulun ympäristöön arviointi kenttä täyttyy normaalisti koulun tietokannasta, mutta yhteyslauseet kuten strConnection ja SQL-lauseet on muutettava koulun ympäristöön sopivaksi. Tietokannan kirjautumistiedot annetaan strConnection-stringissä. Connection-string sijaitsee koodin alkupäässä, ettei sitä tarvitse kirjoittaa joka kerta uudestaan kun tietokantapalvelimelle halutaan ottaa yhteys.

### Esimerkki 3.

```
string strConnection = "UID = testi;PWD = testi;Initial
Catalog=testi;Data Source = MURHA-CC832AD1D;";
```

UID merkitsee käyttäjätunnusta, jolla kirjaudutaan SQL-palvelimelle, PWD:llä annetaan käyttäjän salasana, Initial Catalog -kohtaan kirjoitetaan tietokanta, josta haussa olevien taulujen tulisi löytyä ja Data Source -kohdassa annetaan palvelimen nimi, johon yhteys otetaan parametriin. strConnectionin tunnistaminen on mahdollista hoitaa myös Windows-autentikoinnin perusteella, jolloin käyttäjätunnusta tai salasanaa ei tarvitse syöttää koodissa. Tällöin yhteyslauseen tulisi olla muotoa Data Source=Tietokantapalvelin;Initial Catalog=Tietokanta;Integrated Security=SSPI;

Tunnus	Ylempi_opinto	Opinnon_nimi	Opinnon_tyyppi	Toteutus	Arviointi	Arviointi_txt	Arviointi_pvm	Laajuus	Laajuudenyksikkö
600714	KL17BTTT240	Systemityöprojekti	OJ	8	Hyvä		20100114	7.00	OP
600714	KL17BTEL200	Palvelinohjelmointi	OJ	5	Hyvä		20091009	4.00	OP
600714	KL17DWT06	3DS MAX perusteet	OJ	2	Kiitettävä		20090604	2.00	OP
600714	KL17BTTT285	Open Source-ohjelmistot	OJ	3	Hyvä		20090522	3.00	OP
600714	KL17BTTT111	Ohjelmistotuotanto	OJ	6	Hyvä		20090508	3.00	OP
1 2 3 4 5 6 7 8 9 10 ...									

### Kuvio 5. WinhaWP-web-osio perusasetuksilla

Kuviossa 5 on rakennetun web-osion oletusnäkyminen. Tietueet järjestetään arviointipäivämäärän mukaan uusin ensin. Alalaidassa olevista numeroista voidaan vaihtaa sivua. WinhaWP:n asetuksista on mahdollista säätää erillisiä määrittäjiä, kuinka tieto esitetään taulukossa. Säätömahdollisuuksia ovat yksinkertaistettu näkymä, kuinka monta riviä näytetään yhdellä sivulla, sekä taulukon ulkomuotoon liittyviä asetuksia. Asetuksista kerrotaan lisää luvussa 5.3 Custom-valikoiden luonti.

## Esimerkki 4.

```

//Yhteyden ottoon liittyviä
DbAccess objDB = new DbAccess();
objDB.StrConnection = strConnection;
DataTable dt = new DataTable();
string sql = sqlquery;

//Tarkistetaan yksinkertaistetun näkymän arvo
if (simpleSQL == true)
{
    sqlquery = SQLqSimple;
}
else
{
    sqlquery = SQLq;
}

dt = objDB.GetSqlResult(sql);

Controls.Add(gv);
//määritellään vakio arvo
if (datarows == 0)
{
    datarows = 5;
}
if ( cellPadding == 0)
{
    cellPadding = 5;
}
//GridViewiin liittyviä asetuksia
gv.AllowPaging = true;
gv.PageSize = datarows;
gv.PageIndexChanging += new
GridViewPageEventHandler(this.gv_PageIndexChanging);
gv.Load += new EventHandler(this.gv_Load);
gv.DataSource = dt;
gv.AlternatingRowStyle.BackColor =
System.Drawing.Color.WhiteSmoke;
gv.GridLines = GridLines.Horizontal;
gv.AutoGenerateColumns = true;
gv.CellPadding = cellPadding;
gv.CellSpacing = cellSpacing;
gv.DataBind();
gv.HeaderStyle.BackColor =
System.Drawing.Color.LightGray;
//gv.AllowSorting = true;
//gv.Sorting += new GridViewSortEventHandler(gv_Sorting);

}
catch (Exception ex)
{
    Label lbl = new Label();
    lbl.Text = "Error Occured: ";
    lbl.Text += ex.Message;
    Controls.Add(lbl);
}

```

Kuvion 5 taulukko on rakennettu esimerkki 4 koodiesimerkillä. Koodin alussa poimitaan yhteysasetuksien tieto ja määritellään DataTable, joka toimii taulukon tiedonlähteenä. SQL-lause on määritelty koodissa etukäteen SQLqSimple- ja SQLq-atribuutteihin, if-lauseella tarkastetaan kumpaa käyttäjä on halunnut käyttää. Jos SQL-lause täytyisi syöttää useammassa kuin yhdessä kohdassa koodia, täytyisi SQL-lause vaihtaa jokaiseen erikseen kun web-osio otetaan käyttöön. Tästä syystä SQL-lause haluttiin syöttää attribuutteihin, ne löytyvät helposti yhdestä sijainnista.

### Esimerkki 5.

```
SQLq = "Tunnus, Ylempi_opinto, Opinnon_nimi, Opinnon_tyyppi, Toteutus,
Arviointi, Arviointi_txt, Arviointi_pvm, Laajuus, Laajuudenyksikkö FROM
koulu WHERE (Tunnus = '" + employeeID + "') ORDER BY Arviointi_pvm " +
vanhinUusin + """;
```

```
SQLqSimple = "SELECT Opinnon_nimi, Arviointi, Arviointi_pvm FROM koulu
WHERE (Tunnus = '" + employeeID + "') ORDER BY Arviointi_pvm " +
vanhinUusin + """;
```

Esimerkissä 5 esitetään SQL-lauseet. SQLq hakee kaiken tiedon liittyen opiskelijoiden suorituksiin, kun taas SQLqSimple hakee ainoastaan kaiken välttämättömän tiedon suorituksista. Tieto haetaan kannasta opiskelijatunnuksen perusteella, joka saadaan employeeID:stä. EmployeeID:stä kerrotaan lisää luvussa 5.4 Opiskelijatunnuksen hakeminen Active Directorystä. VanhinUusin-attribuutti sisältää tiedon tehdäänkö järjestys uusimman (DESC) vai vanhimman (ASC) tiedon perusteella. Selkeyden vuoksi on SQL-lauseet esimerkkiin kirjoitettu yksinkertaistetusti.

Koska lisättyjen asetusten vakioarvo oli DefaultValue-määrityksestä huolimatta nolla, tarkistettiin sen koodissa ja määritettiin tätä kautta uudet vakioarvot. Esimerkin 4 loppupuolella annetaan gridView:n arvot ja lisätään erillaisia toiminnallisuuksia. Kommentteissa oleva AllowSorting määritys mahdollistaisi Kuviossa 5 olevan taulukon tietojen järjestämisen riviotsikkaa klikkaamalla. Ominaisuutta ei kuitenkaan pidetty hyvänä, koska silloin käyttäjä joutuisi

napsauttamaan haluamansa järjestyksen aina uudestaan sivulle tullessaan. Ominaisuuden saa käyttöön poistamalla koodista kommentti merkinnät ja SQL-lauseesta ORDER BY -määrityksen.

Opinnon_nimi	Arviointi	Arviointi_pvm
Systeemityöprojekti	4	20100114
Palvelinohjelmointi	4	20091009
3DS MAX perusteet	5	20090604
Open Source-ohjelmistot	4	20090522
Ohjelmistotuotanto	4	20090508
1 2 3 4 5 6 7 8 9 10 ...		

Kuvio 6. WinhaWP yksinkertaistetussa näkymässä

Kuviossa 6 on WinhaWP yksinkertaistetussa näkymässä. Tähän näkymään haetaan tietokannasta vain kaikki välttämätön tieto, mikä selkeyttää taulukkoa huomattavasti ja tekee siitä helposti luettavamman. Web-osio ei vie SharePoint sivustolta kovin paljon tilaa.

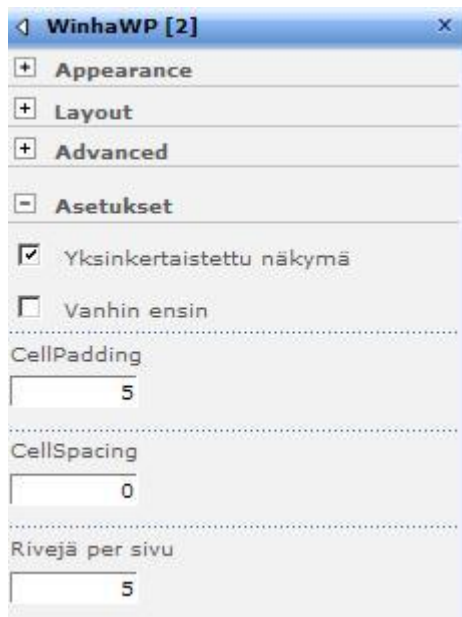
### 5.3 Custom-valikoiden luonti

Jokaiselle SharePoint-sivulle lisätyllä web-osiolla on oma asetusvalikko, johon pääsee helposti käsiksi pienestä nuolesta, joka sijaitsee web-osion vierellä. Asetusvalikkossa on vakiona tietyt asetukset, joilla voi hallita esimerkiksi web-osion kokoa. Asetusvalikkoon on mahdollista lisätä myös omia asetuksia, näitä asetuksia kutsutaan Custom Propertyiksi.

Taulukko 2. Mahdolliset Custom propertyt (Microsoft 2010b).

Property type	Displayed in property plane as
Boolean	Check box
Enum	Drop down
Integer	Text box
String	Text box
DateTime	Text box

Taulukossa 2 on lueteltuna mahdolliset propertyt, joita asetusvalikkoon on mahdollista lisätä. Boolean arvo voi olla vain tosi tai epätosi (rasti ruutuun -laatikko). Enumiin voidaan antaa useita eri arvoja (pudotusvalikko). Integeriin voidaan antaa vain numeerisia arvoja (tekstilaatikko). Stringiin on mahdollista kirjoittaa tekstiä ja numeroita (tekstilaatikko). DateTime käsittelee päivänmääriä (tekstilaatikko).



Kuvio 7. WinhaWP:n asetusvälilehti

Kuviossa 7 esitetään WinhaWP:n asetusvälilehti. Rastittamalla Yksinkertaistettu näkymä -vaihtoehdon käyttöön aktivoituu yksinkertaisempi SQL-lause ks. 5.3 Tietokantayhteyden muodostaminen. Rastittamalla Vanhin ensin -valinnan WinhaWP:n taulukon tiedot järjestetään päivänmäärän mukaan nousevaan järjestykseen. Cellpadding ja CellSpacing ovat taulukon ulkonäköön liittyviä asetuksia. Niillä määritellään taulukon kenttien väliä toisiinsa. Oletuksena nämä säädöt ovat CellPadding 5 ja CellSpacing 0. Rivejä per sivu -toiminnolla määritetään kuinka monta tietuetta esitetään WinhaWP-web-osion sivulla kerrallaan. Mikäli käyttäjä haluaa nähdä kaikki tietueet kerralla, kenttään voidaan syöttää arvoksi esimerkiksi 100.

### Esimerkki 6.

```

private Boolean simpleSQL;
[Personalizable(), WebBrowsable(true),
WebDisplayName("Yksinkertaistettu näkymä"),
Category("Asetukset")]

public Boolean SimpleSQL
{
    get { return simpleSQL; }
    set { simpleSQL = value; }
}

```

Yllä olevassa esimerkissä luodaan Yksinkertaistettu näkymä (simpleSQL) -custom property. WebBrowsable-arvolla kerrotaan esitetäänkö kenttä asetusvälilehdellä vai ei. Categoryyn kirjoitetaan välilehti johon toiminto halutaan. Muiden asetusten koodia ei esitetä, koska ne ovat kaikki hyvin saman tyyppisiä. Valikot esitetään SharePointissa samassa järjestyksessä kuin ne on kirjoitettu koodiin.

## 5.4 Opiskelijatunnuksen hakeminen Active Directorysta

Active Directory(AD) on Windows Server -käyttöjärjestelmissä oleva käyttäjätietokanta, jonka avulla serverille voidaan lisätä uusia käyttäjiä ja muokata vanhoja. Luonnollisesti AD:ssa on mahdollista hallita käyttäjien käyttöoikeuksia erilaisiin palveluihin, luoda käyttäjäryhmiä ja hallita niitä. Active Directoryyn voidaan lisätä muutakin tietoa käyttäjistä kuten nimen lisäksi mm. työpaikka, osoite, puhelinnumero, sähköpostiosoite, maa jne.

Seinäjoen ammattikorkeakoulun Tekniikan yksikön opiskelijatunnukset on sijoitettu Active Directoryyn employeeID-atribuuttiin. Jotta saadaan SQL-lause hakemaan tietoa oikealta käyttäjältä, on ensin tarkistettava kuka on kirjautunut sisään, ja sen jälkeen haetaan sisäänkirjautuneen käyttäjän opiskelijatunnus employeeID-atribuutista. Testipalvelimelta puuttui employeesID, joten se jouduttiin ensin luomaan manuaalisesti Visual Basic scriptin avulla.



## Esimerkki 7.

```

//Tarkistetaan kirjautunut käyttäjä
WindowsPrincipal p = Thread.CurrentPrincipal as WindowsPrincipal;
string strLoggedInUser = p.Identity.Name;

//Poistaa domainin nimen stringin alusta(6 ekaa kirjainta, lasku alkaa
nollasta)
string kirjautunut = strLoggedInUser.Remove(0, 6);

// otetaan yhteys Active directoryyn
DirectoryEntry entry = new DirectoryEntry("LDAP://MURHA");
DirectorySearcher Dsearch = new DirectorySearcher(entry);

Dsearch.Filter =
"(&(objectClass=user)(sAMAccountName=*"+kirjautunut+"*))";

foreach (SearchResult sResultSet in Dsearch.FindAll())
{
employeeID = (GetProperty(sResultSet, "employeeID"));

}

public static string GetProperty(SearchResult searchResult, string
PropertyName)
{

    if(searchResult.Properties.Contains(PropertyName))
    {
return searchResult.Properties[PropertyName][0].ToString() ;

    }
else
    {
return string.Empty;
    }

}
}

```

SharePointtiin kirjautuneen käyttäjän tarkastaminen onnistuu helposti lyhyellä koodilla. Koska käyttäjätunnuksen tulos kyseisellä tavalla on muotoa <domain>/<käyttäjätunnus>, se täytyi käsitellä niin, että jäljelle jää ainoastaan käyttäjätunnus. Haku AD:sta epäonnistuisi, jos käyttäjätunnuksen etuliitteenä olisi domainin nimi. Koodiesimerkissä tekstin alusta poistetaan kuusi ensimmäistä kirjainta, tässä tapauksessa se merkitsee MURHA/-kirjoituksen poistamista.

Kirjautuneen käyttäjän tarkastamisen jälkeen voidaan ottaa yhteys Active directoryyn, josta poimitaan employeeID. EmployeeID lisätään SQL-lauseen

hakuehtoihin ja sen perusteella ohjelma osaa etsiä tietokannasta ainoastaan sisäänkirjautuneen käyttäjän tiedot.

## 5.5 Ohje winhaWP-web-osion käyttöönottamiseen

Tämä lyhyehkö ohje on kirjoitettu Seinäjoen ammattikorkeakoulun järjestelmänhallitsijoita ajatellen. Tässä kappaleessa käydään läpi WinhaWP-web-osion käyttöönottoa vaihe vaiheelta, sekä käydään läpi yhteyslauseet, jotka täytyy muuttaa web-osion koodissa, että WinhaWP voi toimia uudessa ympäristössä.

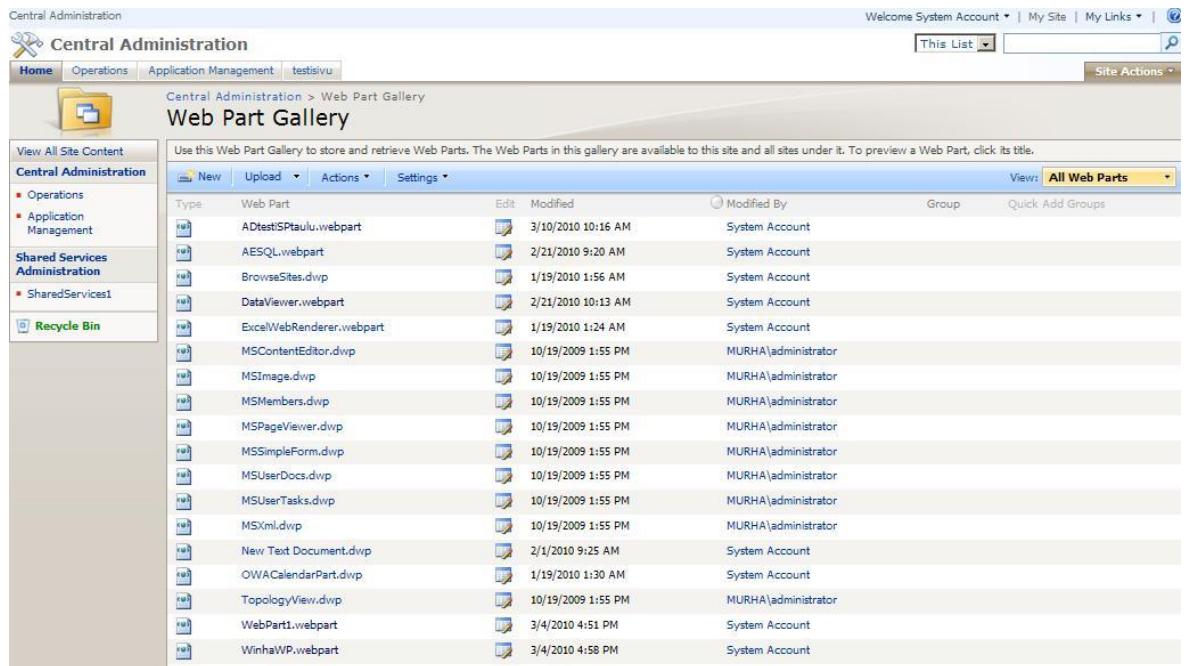
**Vaihe 1. WinhaWP.dll-tiedoston kopioiminen oikeaan kansioon.** dll-tiedosto voidaan kopioida kahteen eri kansioon. Global assembly catcheen (GAC) tai WSS-sivun bin-kansioon. Bin-kansio sijaitsee polun `C:\Inetpub\wwwroot\wss\VirtualDirectories\<portin numero>\bin` takana. Windows server 2003 -käyttöjärjestelmässä GAC sijaitsee osoitteessa `C:\WINDOWS\assembly`. Tässä tapauksessa suositellaan dll-tiedoston lisäämistä GAC-kansioon, muutoin tiedostolle joudutaan säätämään riittävät oikeudet jotka mahdollistaa tietokantayhdeyden ottamisen. Dll-tiedoston lisääminen GAC-kansioon onnistuu helposti raahaamalla se sinne.

**Vaihe 2. web.config-tiedoston muokkaaminen.** Web.Config-tiedosto löytyy polun `C:\Inetpub\wwwroot\wss\VirtualDirectories\<portin numero>` takaa. Config-tiedoston voi avata esimerkiksi Notepadissä. Etsi config-tiedostosta `<SafeControls>`-määrittymiset ja lisää sinne seuraava rivi

```
<SafeControl Assembly="WinhaWP, Version=1.0.0.0, Culture=neutral, PublicKeyToken=9f4da00116c38ec5"
Namespace="WinhaWP" TypeName="*" Safe="true" />
```

**Vaihe 3. web-osion lisääminen SharePoint-palveluun.** Vaiheiden 1 ja 2 jälkeen voidaan web-osio ottaa käyttöön SharePointin puolella. Käynnistä SharePointin keskitetty hallinta. Sivuston toiminnot -pudotusvalikosta valitse sivuston asetukset.

Kohdasta Valikoimat(galleries) napsauta web-osat, tämän jälkeen avautuu kuvion 8 näkymä.



Kuvio 8. SharePointin web-osio valikoima

Kuvion 8 näkymän kautta voidaan lisätä, poistaa ja tarkastella web-osiota. Web-osiota lisätään klikkaamalla Uusi-painiketta. Tämän jälkeen aukeaa lista kaikista käytettävissä olevista web-osiosta. Mikäli WinhaWP ei näy tässä listassa vaihe 1 tai 2 on epäonnistunut. Etsi listasta WinhaWP ja rastita sen vieressä oleva ruutu kuten kuvion 9 esimerkissä esitetään. Tämän jälkeen palaa sivun alkuun ja napsauta Täytä valikoima -painiketta ja WinhaWP-web-osiota siirtyy kuvion 8 esittämään web-osiota valikoimaan, josta sen tietoja voidaan tarkastella, muokata ja katsoa web-osion esikatselu.



Kuvio 9. WinhaWP web-osiota listassa.

Vaiheen 3 suorittamisen jälkeen lisätty web-osio voidaan lisätä mille tahansa SharePoint-sivulle, jossa web-osiot ovat käytössä. Vaiheet 1 - 3 käsittelevät kuitenkin vain web-osion lisäämistä SharePointin listoille. Ennen kuin WinhaWP toimii on sen koodiin muutettava uuden ympäristön yhteysosoitteet. Koodin muokkaaminen käy helposti WinhaWP Visual Studio -projektin avaamalla. Tämä on luovutettu koulun käyttöön.

Kohta 1. Tällä lauseella otetaan yhteys tietokantaan.

```
string strConnection = "UID = käyttäjäTunnus;PWD = salasana;Initial
Catalog=tietokanta;Data Source = serverin nimi;";
```

Kohta 2. Muuta SQLq- ja SQLqSimple-attribuuttien arvoihin haluamasi SQL-lause. Katso lisätietoja kohtiin 1 - 2 kappaleesta 5.2 Tietokantayhteyden muodostaminen.

Kohta 3. Varmista että alla oleva lause poistaa oikean määrän kirjaimia, eli domainin nimen tunnuksen alusta.

```
//Poistaa domainin nimen stringin alusta(6 ekaa kirjainta, lasku alkaa
nollasta)
string kirjautunut = strLoggedInUser.Remove(0, 6);
```

Kohta 4.

```
// otetaan yhteys Active dirjectoriin
DirectoryEntry entry = new DirectoryEntry("LDAP://MURHA");
```

Kirjoita DirectoryEntryyn domainin nimi, josta halutaan hakea käyttäjien tietoja.

Kohta 5. Varmista että employeeID-attribuuttiin haetaan oikea arvo AD:stä. Lisätietoja kohtiin 3 - 5 katso kappale 5.4 Opiskelijatunnuksen hakeminen Active Directorystä.

Kun tarvittavat muutokset on tehty, rakenna WinhaWP-web-osio Visual Studiassa klikkaamalla Build-valikosta Build WinhaWP. Tämän jälkeen siirrä siitä syntynyt .dll-tiedosto GAC- tai bin-kansioon kuten vaiheessa 1 neuvotaan. Jos web-osio on jo otettu käyttöön SharePointissa, on myös resetoitettava IIS ennenkuin muutokset näkyvät SharePointissa. Tämä onnistuu avaamalla Käynnistä-valikon Suorita...-toiminto ja antamalla komento IISReset.

## 6 LOPUKSI

Opinnäytetyön tavoitteena oli luoda Seinäjoen ammattikorkeakoulun Tekniikan yksikölle web-osio, joka mahdollisesti joskus otettaisiin käyttöön opiskeljiaportalissa. Web-osion haluttiin hakevan ja esittävän tietoa WinhaWillen tietokannoista. Tavoite onnistui tekijän mielestä hyvin

Työ ja kaikki siihen liittyvät tiedostot sekä testiympäristönä toiminut virtuaalikone luovutettiin Seinäjoen ammattikorkeakoululle. Toimeksiantajani oli tyytyväinen työn lopputulokseen, heidän sanojensa mukaan web-osio tekee sen mitä siltä haluttiinkin.

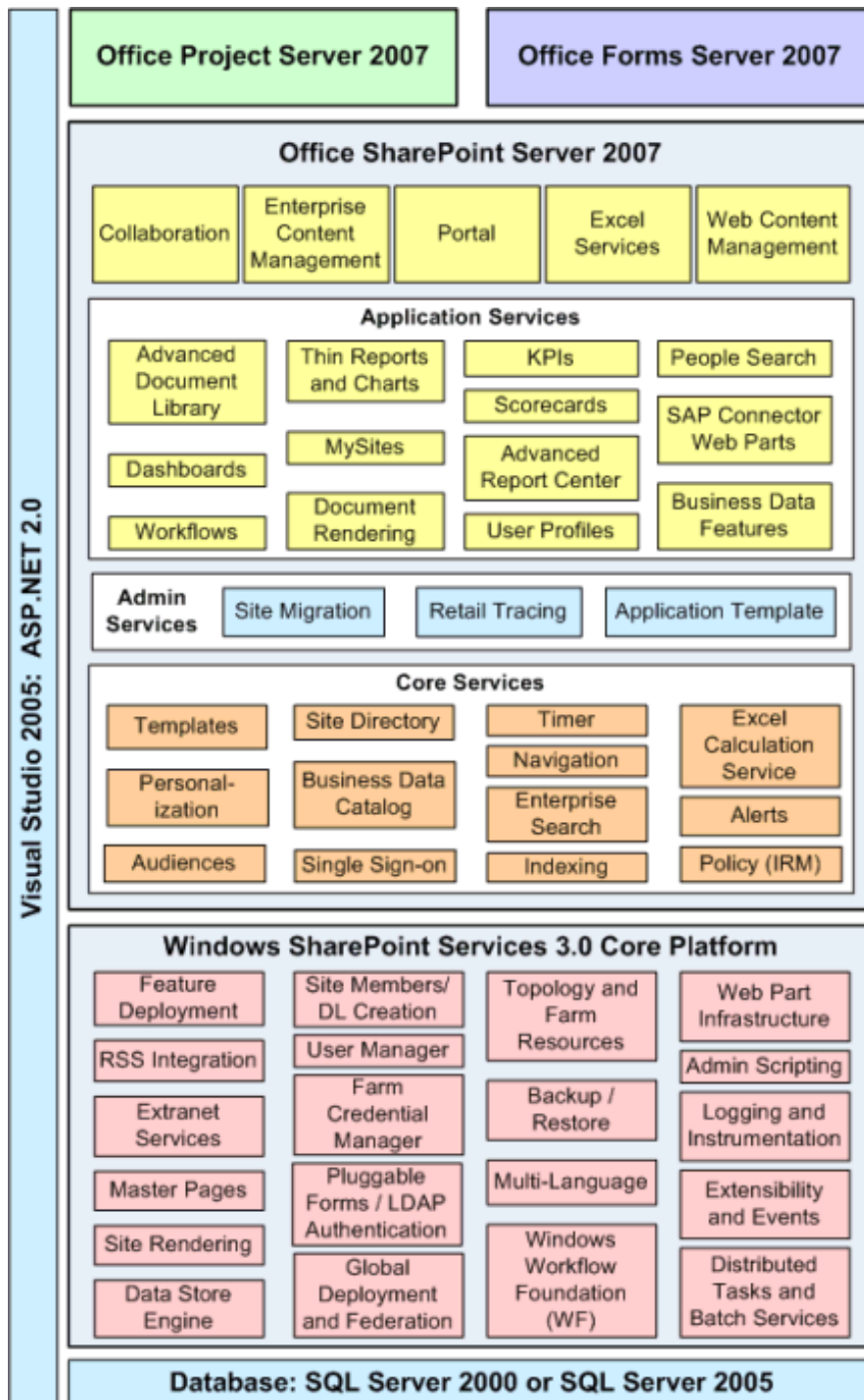
## LÄHTEET

- English, B. 2007. Office sharepoint server 2007: Administrator's Companion. Washington: Microsoft press
- Gajendran, S. 2008. Features of WSS 3.0 and MOSS 2007. 3.4.2008. [WWW-dokumentti]. C#-corner. [Viitattu: 9.3.2010]. Saatavana: <http://www.c-sharpcorner.com/UploadFile/sarav82/FeaturesofWSSandMOSS02282008005632AM/FeaturesofWSSandMOSS.aspx>
- Google. 2010a. Gadgets API. [WWW-dokumentti]. [Viitattu: 24.3.2010]. Saatavana: <http://code.google.com/intl/fi/apis/gadgets/index.html>
- Google. 2010b. Google Sites Help. [WWW-dokumentti]. [Viitattu: 15.4.2010]. Saatavana: <http://sites.google.com/support/?hlrm=fi>
- Google. 2010c. Google wave. [WWW-dokumentti]. [Viitattu: 24.3.2010]. Saatavana: <http://wave.google.com/about.html>
- Itec IT education center. Microsoft Office Sharepoint Server Hakupalvelut. [WWW-dokumentti]. Ei päiväystä. [Viitattu: 1.4.2010]. Saatavana: [http://www.iteducate.fi/infopiste/files/MOSS2007\\_Search%20Services.pdf](http://www.iteducate.fi/infopiste/files/MOSS2007_Search%20Services.pdf)
- Microsoft. 2010a. Global Assembly Cache. [WWW-dokumentti]. [Viitattu: 9.3.2010]. Saatavana: <http://msdn.microsoft.com/en-us/library/yf1d93sz%28VS.71%29.aspx>
- Microsoft. 2010b. CustomPropertyToolPart Class. [WWW-dokumentti]. [Viitattu: 20.3.2010]. Saatavana: <http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.webpartpages.custompropertytoolpart.aspx>
- Microsoft. 2010c. Windows SharePoint Services 3.0 Tools: Visual Studio 2008 Extensions, Version 1.2. [WWW-dokumentti]. [Viitattu: 14.4.2010]. Saatavana: <http://www.microsoft.com/downloads/details.aspx?familyid=7BF65B28-06E2-4E87-9BAD-086E32185E68&displaylang=en>

- MOSS WSS Components Architecture in a single diagram. 26.2.2008. [Verkkosivu]. Sharenotes. [Viitattu: 8.2.2010]. Saatavana: <http://sharenotes.wordpress.com/2008/02/26/moss-wss-components-architecture-in-a-single-diagram/>
- Murphy, A. & Perran, S. 2007. Beginning SharePoint 2007: Building Team Solutions with MOSS 2007. Indianapolis: Wiley Publishing.
- Roine, J. 2007. Microsoft 2007 Office system: Uudet ominaisuudet. Jyväskylä: Readme.fi.
- Roth, C. 22.10.2008. Intro to SharePoint Development: How to Build and Deploy a Web Part. [WWW-dokumentti].. Corey Roth [Viitattu: 28.1.2010]. Saatavana: <http://www.dotnetmafia.com/blogs/dotnettipoftheday/archive/2008/10/22/intro-to-sharepoint-development-how-to-build-and-deploy-a-web-part.aspx>
- Seamk. Willen käyttöohjeita. [WWW-dokumentti]. Ei päiväystä. [Viitattu: 1.5.2010]. Saatavana: <https://winha.seamk.fi/WilleOhje.pdf>
- Sivula, A. 2010a. Atk-suunnittelija. Seinäjoen ammattikorkeakoulu. Haastattelu 15.2.2010
- Sivula, A. 2010b. <xxx.xxx@xxx.fi> 24.2.2010. SQL-kyselyt. [Henkilökohtainen sähköpostiviesti]. Vastaanottaja: Jarkko Laukkanen. [Viitattu 26.2.2010]
- Waller, A. 2009. Microsoft Office SharePoint Server 2007. [WWW-dokumentti]. The SharePoint User Group(TheSUG). [Viitattu: 11.3.2010]. Saatavana: <http://www.thesug.org/mossasaurus/Wiki%20Pages/Microsoft%20Office%20SharePoint%20Server%202007.aspx>

## LIITTEET

LIITE 1. MOSS- ja WSS- ohjelmistojen toiminnallisuudet (MOSS WSS Components Architecture in a single diagram 2008)





Liitteen 1 kuvassa esitetään yhdessä kuvassa MOSS ja WSS -versioiden eroja. Kuvan allimmalla tasolla on tietokanta-ohjelmistot, joiden käyttäminen SharePoint-palveluissa on välttämätöntä. Toiseksialimmassa laatikossa esitetään Windows SharePoint Service (WSS) ja sen sisältämät ominaisuudet. Microsoft Office Sharepoint Server (MOSS) ominaisuudet on jaettu kolmeen osioon. Application Services, Admin Services ja Core Services. MOSS pitää sisällään myös samat ominaisuudet, joita on WSS-versiossa.

## LIITE 2. Office SharePoint Server 2007 lisenssivaihtoehdot (Itec IT education center)

Features	Office SharePoint Server 2007 for Search (Standard Edition)	Office SharePoint Server 2007 for Search (Enterprise Edition)	Office SharePoint Server 2007 (Standard CAL)	Office SharePoint Server 2007 (Standard plus Enterprise CAL)
File shares	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Web sites	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SharePoint sites	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Microsoft Exchange Server public folders	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lotus Notes databases	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Third party document repositories <sup>1</sup>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Secure content access control	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Enhanced search center user interface			<input type="radio"/>	<input type="radio"/>
Search for people and expertise			<input type="radio"/>	<input type="radio"/>
Business Data Catalog (BDC)				<input type="radio"/>
Search structured data sources				<input type="radio"/>
Document limit	500,000	No Limit <sup>2</sup>	No Limit <sup>2</sup>	No Limit <sup>2</sup>

Kuvassa esitetään Microsoft Office SharePoint Server 2007 -ohjelmiston mahdolliset lisenssivaihtoehdot. Palloilla on merkitty lisensseillä käyttöönsaatavat ominaisuudet.

### Liite 3. Lähdekoodi: WinhaWP.cs.

```
using System;
using System.Runtime.InteropServices;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Serialization;

using Microsoft.SharePoint;
using Microsoft.SharePoint.WebControls;
using Microsoft.SharePoint.WebPartPages;
using System.Data;
using System.Reflection;
using System.ComponentModel;
using System.Security.Principal;
using System.Threading;
using System.Text;

using System.DirectoryServices;
using System.DirectoryServices.ActiveDirectory;
using System.Collections;

namespace WinhaWP
{
    [Guid("be07f994-3c9e-47e2-9ba8-dd0a14c4b493")]
    public class WinhaWP : System.Web.UI.WebControls.WebParts.WebPart
    {

        GridView gv = new GridView();

        string vanhinUusin = "";
        string employeeID = "";

        Label ba = new Label();

        private Boolean simpleSQL;
        [Personalizable(), WebBrowsable(true),
        WebDisplayName("Yksinkertaistettu näkymä"),
        Category("Asetukset")]

        public Boolean SimpleSQL
        {
            get { return simpleSQL; }
            set { simpleSQL = value; }
        }
    }
}
```

```

private Boolean vanhinEnsin;
[Personalizable(), WebBrowsable(true),
WebDisplayName("Vanhin ensin"), Category("Asetukset"),
Description("")]

public Boolean VanhinEnsin
{
    get { return vanhinEnsin; }
    set { vanhinEnsin = value; }
}

private string sqlquery;
[Personalizable(), WebBrowsable(false),
WebDisplayName("sqlquery"), Category("Asetukset")]

public string SQLQuery
{
    get { return sqlquery; }
    set { sqlquery = value; }
}

private int CellPadding;
[Personalizable(), WebBrowsable(true),
WebDisplayName("CellPadding"), DefaultValue("5"),
Category("Asetukset")]

public int cellPadding
{
    get { return CellPadding; }
    set { CellPadding = value; }
}

private int CellSpacing;
[Personalizable(), WebBrowsable(true),
WebDisplayName("CellSpacing"), Category("Asetukset")]

public int cellSpacing
{
    get { return CellSpacing; }
    set { CellSpacing = value; }
}

private int datarows;
[Personalizable(), DefaultValue("5"), WebBrowsable(true),
WebDisplayName("Rivejä per sivu"), Category("Asetukset"),
Description("Kuinka monta riviä näytetään per rivi"), ]

public int DataRows
{
    get { return datarows; }
    set { datarows = value; }
}

string strConnection = "UID = testi;PWD = testi;Initial
Catalog=testi;Data Source = MURHA-CC832AD1D;";
string SQLq = "";
string SQLqSimple = "";

```

```

        protected override void OnLoad(EventArgs e)
        {
            base.CreateChildControls ();
            try
            {
                if (vanhinEnsin == true)
                {
                    vanhinUusin = "ASC";
                }
                else
                {
                    vanhinUusin = "DESC";
                }

                //Tarkistetaan kirjautunut käyttäjä
                WindowsPrincipal p = Thread.CurrentPrincipal as
WindowsPrincipal;
                string strLoggedInUser = p.Identity.Name;

                //Poistaa domainin nimen stringin alusta(6 ekaa
                kirjainta, lasku alkaa nollasta)
                string kirjautunut = strLoggedInUser.Remove(0, 6);

                DirectoryEntry entry = new
DirectoryEntry("LDAP://MURHA");
                DirectorySearcher Dsearch = new DirectorySearcher(entry);

                Dsearch.Filter =
"(&(objectClass=user)(sAMAccountName=*"+kirjautunut+"*)");

                foreach (SearchResult sResultSet in Dsearch.FindAll())
                {
                    employeeID = (GetProperty(sResultSet, "employeeID"));
                }

                // SQLqSimple = sqlquery = "SELECT
                R_OPINTO#OPINTO_NIMI AS Opinnon_nimi, R_HOPSIN_OPINTO#ARV AS Arviointi,
                R_HOPSIN_OPINTO#ARVIOPVM AS Arviointi_pvm FROM koulu WHERE
                (R_HOPSIN_OPINTO#OPRLI = '"' + employeeID + "') AND R_OPINTO#OPINTO_NIMI
                LIKE '%" + hakusana + "%' ORDER BY R_HOPSIN_OPINTO#ARVIOPVM " +
                vanhinUusin + ";

                SQLq = "SELECT R_HOPSIN_OPINTO#OPRLI AS Tunnus,
                R_HOPSIN_OPINTO#OPINTO AS Ylempi_opinto, R_OPINTO#OPINTO_NIMI AS
                Opinnon_nimi, R_HOPSIN_OPINTO#OPITYYP AS Opinnon_tyyppi,
                R_HOPSIN_OPINTO#TOTEUTUS AS Toteutus, R_HOPSIN_OPINTO#ARV AS Arviointi,
                R_HOPSIN_OPINTO#ARV_TXT AS Arviointi_txt, R_HOPSIN_OPINTO#ARVIOPVM AS
                Arviointi_pvm, R_HOPSIN_OPINTO#LAAJ AS Laajuus, R_HOPSIN_OPINTO#LAAJYKS
                AS Laajuudenyksikkö FROM koulu WHERE (R_HOPSIN_OPINTO#OPRLI = '"' +
                employeeID + "') ORDER BY R_HOPSIN_OPINTO#ARVIOPVM " + vanhinUusin + ";
                SQLqSimple = "SELECT R_OPINTO#OPINTO_NIMI AS
                Opinnon_nimi, R_HOPSIN_OPINTO#ARV AS Arviointi, R_HOPSIN_OPINTO#ARVIOPVM

```

```

AS Arviointi_pvm FROM koulu WHERE (R_HOPSIN_OPINTO#OPRLI = '' +
employeeID + '') ORDER BY R_HOPSIN_OPINTO#ARVIOPVM '' + vanhinUusin + '';

/*
//Poistaa KAIKKI kirjaimet stringistä
string winhaTunnus = "";
foreach (char c in strLoggedInUser)
{
    if (char.IsDigit(c))
    {
        winhaTunnus += c;
    }
}
*/

//yhteyden otto alkaa
//Connection string siirretty toisalle ettei sitä
tarvitse mainita joka kerta uudestaan
DbAccess objDB = new DbAccess();
objDB.StrConnection = strConnection;
DataTable dt = new DataTable();
string sql = sqlquery;

if (simpleSQL == true)
{
    sqlquery = SQLqSimple;
}
else
{
    sqlquery = SQLq;
}

dt = objDB.GetSqlResult(sql);

Controls.Add(gv);
if (datarows == 0)
{
    datarows = 5;
}
if ( cellPadding == 0)
{
    cellPadding = 5;
}

gv.AllowPaging = true;
gv.PageSize = datarows;
gv.PageIndexChanging += new
GridViewPageEventHandler(this.gv_PageIndexChanging);
gv.Load += new EventHandler(this.gv_Load);
gv.DataSource = dt;
gv.AlternatingRowStyle.BackColor =
System.Drawing.Color.WhiteSmoke;
gv.GridLines = GridLines.Horizontal;
gv.AutoGenerateColumns = true;
gv.CellPadding = cellPadding;
gv.CellSpacing = cellSpacing;
gv.DataBind();
gv.HeaderStyle.BackColor =
System.Drawing.Color.LightGray;

```

```

        //gv.AllowSorting = true;
        //gv.Sorting += new GridViewSortEventHandler(gv_Sorting);

        ba.Text = employeeID;
        Controls.Add(ba);

    }
    catch (Exception ex)
    {
        Label lbl = new Label();
        lbl.Text = "Error Occured: ";
        lbl.Text += ex.Message;
        Controls.Add(lbl);
    }

    base.OnLoad(e);
}

/*
void gv_Sorting(object sender, GridViewSortEventArgs e)
{
    DataTable dataTable = gv.DataSource as DataTable;

    if (dataTable != null)
    {
        DataView dataView = new DataView(dataTable);
        dataView.Sort = e.SortExpression + " " + vanhinUusin;

        gv.DataSource = dataView;
        gv.DataBind();
    }
}

*/

private void gv_Load(object sender, EventArgs e)
{
    //Bindataan uudestaan täällä datagridin päivitysongelmien
    vuoksi

    DbAccess objDB = new DbAccess();
    objDB.StrConnection = strConnection;

    DataTable dt = new DataTable();
    string sql = sqlquery;

    dt = objDB.GetSQLResult(sql);
    gv.DataSource = dt;
    gv.DataBind();
}

protected void gv_PageIndexChanging(object sender,
GridViewPageEventArgs e)
{

```

```
gv.PageIndex = e.NewPageIndex;
DbAccess objdb = new DbAccess();
objdb.StrConnection = strConnection;
DataTable dt = objdb.GetSqlResult(sqlquery);
gv.DataSource = dt;
gv.DataBind();
```

```
}
```

```
public static string GetProperty(SearchResult searchResult, string
PropertyName)
```

```
{
```

```
    if (searchResult.Properties.Contains(PropertyName))
```

```
    {
```

```
        return searchResult.Properties[PropertyName][0].ToString();
```

```
    }
```

```
    else
```

```
    {
```

```
        return string.Empty;
```

```
    }
```

```
}
```

```
}
```

```
}
```



#### Liite 4. Lähdekoodi: DBAcces.cs.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data.SqlClient;
using System.Data;
using System.Web.UI.WebControls;

namespace WinhaWP
{
    public class DbAccess
    {
        private string strConnection;

        public string StrConnection
        {
            get { return strConnection; }
            set { strConnection = value; }
        }
        public DataTable GetSQLResult(string sql)
        {
            DataTable dt = new DataTable();

            try
            {
                SqlConnection cnn = new SqlConnection(strConnection);
                SqlDataAdapter da = new SqlDataAdapter(sql, cnn);
                da.Fill(dt);
                return dt;
            }

            catch (Exception ex)
            {
                return dt;
            }
        }
    }
}
```