

LAPPEENRANNAN TEKNILLINEN YLIOPISTO

Teknistaloudellinen tiedekunta

Ohjelmistotekniikan Laboratorio

Marko Suhonen

Verkkopohjainen palautusjärjestelmä harjoitustöille

Kandidaatintyön aihe on hyväksytty 20.09.2009

Työn tarkastaja toimii: Yliassistentti Uolevi Nikula

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Teknistaloudellinen tiedekunta
Ohjelmistotekniikan koulutusohjelma

Marko Suhonen, 0259290

Verkkopohjainen palautusjärjestelmä harjoitustöille

Kandidaatintyö, 2009

29 sivua , 3 taulukko ja 2 kuvaa

Tarkastaja: Yliassistentti Uolevi Nikula

Hakusanat: PHP, ASP.NET, tiedoston lataus, tuki

Keywords: PHP, ASP.NET, file upload, support

Tiedostojen palautus erilaisissa kursseissa yleistyy. Harjoituksia ja harjoitustöitä palautetaan erilaisissa tiedostomuodoissa, yleensä sähköpostin välityksellä. Oppilaat saattavat palauttaa tiedostoja väärin nimettynä tai tiedostotyyppi on väärä, joka hankaloittaa tarkastusta. Oppilas määrän ollessa suuri, tulee palautuksia myös paljon, joka täyttää tarkastajan sähköpostin. Tämän kandidaatintyön tarkoitus on selvittää mitkä ohjelmointikielet kävisivät verkkopohjaisen palautusjärjestelmän toteutukseen. Verkkopohjainen palautusjärjestelmä toteutetaan kirjallisuudesta löytyvän materiaalin perusteella.

ABSTRACT

Lappeenranta University of Technology
Faculty of Technology Management
Information Technology

Marko Suhonen, 0259290

Web based application for student assignment returns

Bachelor work, 2009

29 pages , 3 tables and 2 figures

Examiner: Uolevi Nikula

Keywords: PHP, ASP.NET, file upload, support

Returning files in different courses is coming more popular than before. Practices and practice assignment are returned with different file types. Sometimes students will send files that are named wrong or they are in wrong type. This will make checking the assignments difficult. On course they might be ten to hundreds of student, meaning that there will be same amount of assignments. Most of the time returning assignments are through email, and that is why teachers will get full email account in no time. In this bachelor work, we are going through some languages that will help us handle and check assignments. After this literature preview, we will implement this web application.

Alkusanat

Tutkiessani työhön liittyviä ohjelmointikieliä, etsin vain tutuista kielistä aluksi materiaalia. Työhön olisi sopinut muitakin ohjelmointikieliä kuin PHP ja ASP.NET, jotka olisivat käyneet verkkosivun toteuttamiseen. Ajan ollessa rajallinen, otin vain PHP:n ja ASP.NET:n huomioon, koska osasin ohjelmoida kyseisillä ohjelmointikielillä. Kiitän tarkastajaa Uolevi Nikulaa kandidaatintyön tarkistamisesta.

15.12.2009

Marko Suhonen

Sisällysluettelo

1 JOHDANTO.....	2
1.1 Tausta.....	2
1.2 Tavoitteet ja rajaukset.....	3
1.2.1 Vaatimusmäärittely.....	4
1.3 Työn rakenne.....	4
2 TEKNIikka.....	5
2.1 Tausta.....	5
2.2 Tutkimusongelman ratkaisutapa.....	6
3 TOTEUTUS.....	8
3.1 Työn kuvaus.....	8
3.2 Työn vaiheet.....	8
3.3 Toteuttaminen.....	9
3.4 Ongelmat.....	9
3.5 Yhteenvedo ja tulkinta.....	10
4 TESTAUS.....	12
5 POHDINTA.....	13
6 JOHTOPÄÄTÖKSET.....	14
Liite 1: Projektisuunnitelma.....	16
Liite 2: Vaatimusmäärittely.....	19

1 JOHDANTO

Kandidaatintyö käsittelee verkkosivujen toteutusta, jotta saataisiin helpotettua tehtävien ja harjoitustöiden palautusta ohjelmistotuotannon kurssiin liittyen. Työssä tehdään vaatimusmäärittely ja arkkitehtuuri verkkosivusta sekä sivu rakennetaan (ohjelmoidaan) kielellä (tekniikalla), joka on katsottu hyväksi kirjallisuudesta löytyvien tietojen perusteella.

1.1 Tausta

Tietyissä Lappeenrannan teknillisen yliopiston kurseissa palautetaan tehtäviä opettajalle. Palautus on tapahtunut sähköpostin välityksellä. Kurseissa, kuten Ohjelmistotuotanto kurssi, palautus tapahtuu PDF tai Excel muodossa [1]. Opettaja haluaa tiedoston tietyssä formaatissa ja nimettynä tietyllä tavalla, koska standardi nimeämiskäytäntö ja formaatti helpottaisivat tarkastusta. Tiedosto pitää sisällään tietyt asiat, esimerkiksi oppilaan nimi ja opiskelijanumero:

Nimi	Joku Jokunen
Opiskelijanumero	1234567

Tiedoston nimeämisessä käytetään esimerkiksi seuraavanlaista käytäntöä: ”HXX-XXXXXXXX.doc eli ensin kirjain "H" (Harjoitus) ja sen jälkeen ko. harjoituksen numero kahdella numerolla (esim. "02"), tavuviiva "-" ja lopuksi opiskelijanumero 7 merkillä; tämän jälkeen on vielä dokumentin tyyppin osoittava tarkenne (esim. doc, pdf, xls, tms.). Harjoituksen 1 tehtävät sisältävä dokumentti voisi siis näyttää seuraavalta: H01-0123456.doc. [2]”. Tiedoston nimeämiskäytäntö saattaa vaihdella kurseittain.

Tiedostossa olevat asiat pitää löytyä, jotta voidaan tarkastaa tiedosto helposti. Nykyisessä “järjestelmässä” palautukseen saattaa mennä paljon aikaa jos oppilas lähettää tiedoston, joka on puutteellinen. Tällöin opettaja joutuu olemaan yhteydessä oppilaaseen ja

kertomaan palautetun työn viat. Oppilas palauttaa tämän jälkeen korjatun tiedoston sähköpostilla takaisin opettajalle. Tämä täyttää opettajan sähköpostin, etenkin jos oppilaita on monia, ja hidastaa tarkistusta. Jossain tapauksissa sähköposti saattaa täytyä siinä määrin, että uusia sähköposteja ei voi vastaanottaa. Tämän vuoksi tiedostojen palautus tällä tavalla on huomattu hyvin huonoksi.

1.2 Tavoitteet ja rajaukset

Työn tavoitteena on tutkia mikä ohjelmointikieli sopisi tiedoston palautukseen ja sen tutkimiseen, sisällöltään ja formaatiltaan. Esimerkkinä käytetään ohjelmistotuotannon kurssissa tapahtuvaa harjoitustöiden palautusta. Tässä tapauksessa toteutettava verkkosivu antaa palauttaa vain Excel ja PDF tiedostoja, eli muiden tiedostotyyppien palautusta ei tehdä. Tavoitteena on tehdä sivusta vaatimusmäärittely ja toiminnallinen määrittely. Lisäksi toteutettavasta sivusta tehdään arkkitehtuuri. Vaatimusmäärittelyn jälkeen on tarkoitus tehdä ja tuottaa kyseinen verkkosivu ohjelmointikielellä, joka on valittu kirjallisuuden avulla. Verkkosivu tehdään korvaamaan nykyistä sähköpostin välityksen avulla tapahtuvaa tehtävien ja harjoitustöiden palautusta. Sivusta yritetään tehdä sellainen, että sitä voidaan käyttää myös muiden kurssien palautuksiin. Ulkoasu sivulle tehdään myöhemmin, jos aikaa jää. Verkkopohjaisen palautusjärjestelmässä yritetään keskittyä toiminnallisuuteen enemmän.

Valmistuvan sivun pitää tallentaa tiedostot koulun omalle Linux palvelimelle, josta opettaja voi käydä katsomassa lisättyjä tiedostoja. Verkkosivu myös tallentaa jokaisesta palautusyrityksistä tapahtuman erilliseen tiedostoon. Sivuston tuottamiseen annettua kieltä ei ole annettu, jonka myötä eri kieliä ja toteutuksia tarkastellaan tässä työssä. Verkkosivu antaa palauttajalle viestin jos jotain puutteita on ilmennyt tiedoston palautukseen liittyen. Tarkastajalle ei tehdä erillistä verkkosivua, jossa voitaisiin katsella palautettuja tiedostoja.

1.2.1 Vaatimusmäärittely

Tässä osiossa kerrotaan tärkeimpiä asioita, joita pitäisi kuulua verkkosivuun, jotta se olisi toimiva ja käytännöllinen. Eräs näistä vaatimuksista on, että oppilaat pystyvät palauttamaan tiedostoja. Oppilaiden pitää pystyä tallentamaan tiedostoja palvelimelle verkkosivun kautta. Verkkosivun pitää tarkistaa tietyt asiat tiedostosta, joita ovat tiedoston nimi, formaatti ja sisältö. Oppilailta kysytään nimi ja opiskelijanumero, johon tietoja verrataan. Enemmän vaatimusmäärittelyä liitteessä 2: Vaatimusmäärittely, jossa kerrotaan kaikki palautusjärjestelmään liittyvät ominaisuudet.

1.3 Työn rakenne

Johdannon tarkoitus on johdattaa lukija aiheeseen. Siinä kerrotaan työn tausta, laajuus ja tavoitteet. Työn suhde muihin tutkimuksiin ja lähteisiin myös kerrotaan. Tavoitteet ja rajaukset luvussa kerrotaan mitä työssä aiotaan ja mitä ei aiota tehdä. Tuotettavasta verkkosivusta kerrotaan tärkeimmät vaatimukset vaatimusmäärittely luvussa. Teoriaosuudessa kerrotaan taustatietoja tekniikasta, jolla verkkosivu toteutetaan, lisäksi käydään toista ohjelmointikieltä hieman läpi, jolla olisi voinut myös toteuttaa verkkosivun.

Toteutus osuudessa kuvataan työn vaiheet, toteuttaminen ja ongelmat. Ongelma kappaleessa kerrotaan mitä ongelmia tuli projektin etenemisessä vastaan ja miten ne ratkaistiin. Lisäksi käydään läpi tuloksia ja tulkitaan niitä. Pohdinnassa tutkitaan tulosten merkitystä ja toteutuneen verkkopohjaisen palautusjärjestelmän tulevaisuutta. Johtopäätöksissä kerrotaan mitä tapahtui ja toteutui tässä projektissa.

2 TEKNIikka

Työssä käytettäviä tekniikoita tai ohjelmointikieliä ei ollut määritetty, joten verkkosivun toteutukseen voidaan käyttää mitä tahansa ohjelmointikieltä. Ohjelmointikieliä on lukuisia, mutta tässä työssä tutkittiin etenkin PHP:n (PHP: Hypertext Preprocessor) ja ASP.NET:n ominaisuuksia tarkistaa ja tutkia tiedostoja.

2.1 Tausta

Verkkosivun halutaan tarkistavan tiedoston nimen ja tiedostotyyppin. Tiedostotyyppinä ovat Excel ja PDF tiedostot. Lisäksi verkkosivun halutaan tarkistavan tiedoston sisältö ennen sen tallentamista palvelimelle. PHP näyttää sopivan mainiosti tähän tarkoitukseen, koska tiedostonimen ja tyyppin voi tarkistaa ennen tiedoston varsinaista lataamista palvelimelle. David Power (2007) esittelee kirjassaan PHP:tä ja kuinka sitä voi käyttää hyvin tiedostojen osalta. Power esittelee ”\$_FILES”-taulukon, jonka avulla tiedostotyyppi voidaan tarkistaa Mime-tyyppien mukaan. Saman taulukon avulla voidaan tarkistaa myös tiedostonimi kätevästi. [3, 4] Mime-tyyppi saadaan käyttäjän selaimelta, jolloin selain voi lähettää MIME tyyppin vaikka tiedoston päätteen mukaan eikä tarkista tarkemmin tiedostontyyppiä. Tämä tarkoittaa myös sitä, että tiedostotyyppi voi olla PDF vaikka olisikin oikeasti kuva, eli kuva olisi vain uudelleen nimettynä PDF:ksi.

PhpSoSS (PHP Simple Online Submission System) on tarkoitettu verkkopohjaisiin hallinnointi ja palautusjärjestelmiin. Tällä phpSoSS:lla voidaan rakentaa oppilaille ja opettajalle hallinnointijärjestelmä, jossa voidaan palauttaa tiedostoja ja katsella palautettuja tiedostoja. [5]

ASP.NET on myös hyvä tarkistamaan tiedostonimiä, ja se tuo lisää dynaamisuutta sivuun

enemmän kuin PHP. Armstrong (2005) kirjoittaa teoksessaan, että ASP.NET ohjelmointia voi käyttää hyvin helposti lataamaan tiedostoja yksittäin tai monta kappaletta kerrallaan, käyttäen hyväksi myös JavaScriptiä [6]. MacDonald & Szpuszta (2005) kirjoittavat mitä kaikkea voi tehdä tiedostojen ja kansioiden kanssa. Yksi heidän esimerkeistään on näyttää kansiossa olevat tiedostot sivulla yksinkertaisella komennolla, joka näkyy listaus 1:ssä.

Listaus 1

```
string directoryName = @"c:\Temp";  
// Retrieve the list of files, and display it in the page.  
string[] fileList = Directory.GetFiles(ftpDirectory);  
foreach (string file in fileList){  
    lstFiles.Items.Add(file);  
} [7]
```

MacDonaldin & Szpusztan esittämällä tekniikalla voimme tehdä sivun opettajalle, joka näyttäisi kansiossa olevat tiedostot. Tätä sivua ei tarvita, koska tiedostot opettaja voi hakea muuta kautta. Tiedoston sisällön tarkistamisessa kumpikin, PHP ja ASP.NET ovat hyviä ja toimivat erinomaisesti.

2.2 Tutkimusongelman ratkaisutapa

PHP:lla voi helposti tarkistaa tiedostonimen ja tiedostotyypin. Sillä voi myös tarkistaa Excel ja PDF tiedoston sisällön monella tapaa. Lisäksi PHP oli tutumpi kirjoittajalle, joten PHP toteutuskielenä tuntui luonnolliselta valinnalta. Käyttäjä antaa tietonsa lomakkeen avulla, joten käyttäjän antamaa syötettä joudutaan tarkistelemaan, koska syöte saattaa olla esimerkiksi koodia, joka on haitallista palvelimelle. Tämä pitää huomioida etenkin, koska verkkopohjainen palautusjärjestelmä on suunniteltu yliopistolle. Syötettyjen tietojen tarkistus voidaan tehdä esimerkiksi filttareiden avulla kuten David Powers kirjoittaa kirjassaan PHP Object-Oriented Solutions [8].

3 TOTEUTUS

3.1 Työn kuvaus

Työnä on tehdä verkkopohjainen palautusjärjestelmä oppilaiden harjoitustöille. Palautusjärjestelmä tarkoituksena on tarkistaa tiedostonimet ja tiedostotyypit. Lisäksi tiedostot tarkistetaan sisällöltään. Jokaisessa kurssissa on eri kriteerit tarkistaa tiedostonimet ja tyypit, joten jokaiselle kurssille ja sen harjoitustöille tulee oma tiedosto, josta katsotaan tiedoston tarkistukseen liittyvät asiat. Tällöin voidaan sivua käyttää vain palautukseen jos tarkistamista ei tarvita. Lisäksi jokainen harjoitustyö tallennetaan omiin kansioihin, jotta niiden tarkastelu on helpompaa.

3.2 Työn vaiheet

Aluksi etsittiin materiaalia ja tutkittiin mikä ohjelmointikieli sopisi hyvin verkkosivun (verkkopohjaisen palautusjärjestelmän) tekemiseen. Tämän jälkeen kirjoitettiin alustava vaatimusmäärittely. Alustavan vaatimusmäärittelyn jälkeen tehtiin hahmotelma arkkitehtuurista karkealla tasolla (joka on liitteessä 2: Vaatimusmäärittely). Tämän jälkeen aloitettiin verkkosivun toteutus. Alla on taulukko alku vaiheessa tehdystä aikataulusta, joka löytyy myös liitteestä 1: Projektisuunnitelma.

Taulu 1 : Aikataulu

Sivun versio	Tarkennus	Päivämäärä
0.1	Sivu, jossa pystyy lataamaan tiedoston	08.10.2009
0.2	Tiedostotyypin tarkistus	10.10.2009
0.4	Tiedostonimen tarkistus	15.10.2009
0.4	Oppilas tiedot lomake, jossa kysytään opiskelijanumero ja oppilaan nimi	15.10.2009
0.5	Tiedoston sisällön tutkiminen PDF muodossa	25.10.2009
0.6	Tiedoston tutkiminen Excel muodossa	10.11.2009
0.7	Kurssin valinta	01.12.2009

Yllä olevasta taulukosta tutkiminen PDF ja Excel muodossa vei aikaa enemmän kuin oli oletettu. Etenkin Excel tiedostojen lukeminen ja sisällön tutkiminen tuotti ongelmia, koska jouduttiin etsimään sopivaa ratkaisua hieman enemmän kuin oletettiin. Lisäksi Excel tiedoston sisällön tutkimisen toteutus vei aikaa suunniteltua kauemmin. INI tiedoston lukeminen ja sisällön määrittämistä ei huomioitu aikataulussa, joten sen vuoksi aikataulu venyi.

3.3 Toteuttaminen

Työ, eli verkkopohjainen palautusjärjestelmä, toteutettiin PHP ohjelmointikielellä. Verkkosivu toteutettiin eri vaiheissa, versioissa, jotka testattiin jokaisen version jälkeen. Vaatimusmäärittelyssä (Liite 2) on versiot lueteltuna, ja kerrottu mitä ne tarkemmin sisältävät. Tämän lisäksi versioiden aikataulut on kerrottu Vaatimusmäärittelyssä.

3.4 Ongelmat

Työssä ilmeni hieman ongelmia, joka oli odotettavissa. Ensimmäisenä ongelmana oli ohjelmointikielen valinta, koska kumpikin ASP.NET ja PHP olivat hyviä tähän tarkoitukseen. PHP valittiin, koska testaaminen PHP:llä on helppoa omilla sivuilla, ei tarvitse koululta erillistä testausympäristöä. Lisäksi PHP:n voi asentaa sekä Windows ympäristöön sekä Linux ympäristöön [9], joten ei tarvitse huolehtia siitä, ettei verkkosivu toimisi koulun palvelimella, vaikka PHP:tä testaisi Windows koneella.

Toisena ongelmana tuli vastaan sisällön tutkiminen, koska haluttiin tutkia PDF ja Excel tiedostoja, ja niiden sisältöä. Ongelmana oli määrittellä mitä tarkistetaan, koska tarkistaminen voi tapahtua minkä suhteen vain kurssien vaihdellessa. Tässä vaiheessa INI tiedostoon tehtiin muutoksia, joita sitten voitiin käyttää tiedostojen tutkimiseen. Nämä muutokset antoivat kurssin pitäjän määrittää Excel-tiedostoissa solun ja sen sisällön, ja PDF-tiedostoissa määritettiin vain sisältö käyttäjän antamien tietojen mukaan.

Excel-tiedostojen aukaisuun ja sisällön tutkimisessa ilmeni ongelmia, koska ei voitu tutkia tiedostoa kuten haluttiin. Tähän löytyi pari ratkaisua, yksi ratkaisu oli tallentaa Excel (xls- pääte) tiedosto CSV tiedostona. Palautusjärjestelmän halutaan olevan ”automaattinen”, jolloin ei tarvita ihmistä muuttamaan tiedostoja, joten tämä ratkaisu hylättiin. Lisäksi ratkaisu toisi lisää työtä oppilaille tai opettajalle. Toinen ratkaisu oli PHP, joka tuki COM:ia (Component Object Model). Sitä ei pystytty käyttämään, koska tälle ei ollut tukea. Toinen ratkaisu oli ilmainen PHP Excel Reader, joka on avoin, ilmainen PHP koodi. Tämä koodi lukee Excel tiedoston hyvin ja sopi tähän työn vaatimukseen. Ongelma ratkaistiin ilmaisen koodin avulla, joka auttoi saamaan projektia eteenpäin. Ongelmaan löytyi myös muitakin ratkaisuja, mutta osa näytti olevan kaupallisia [10]. Tämän vuoksi näitä ratkaisuja ei käytetty.

Verkkosivun toteutuksessa ilmeni myös aikataulu ongelmia, jonka vuoksi kurssien valintaa ei voitu toteuttaa. Lisäksi ulkoasuun ei pystytty keskittymään juurikaan joten ulkoasu jäi vaisuksi. Muita palautusjärjestelmään liittyviä ominaisuuksia ei karsittu, eli kaikki muut ominaisuudet tehtiin, paitsi kurssien valintaa. Lisäksi aikataulun pettäessä jouduttiin myös verkkosivun palautus ajankohtaa muuttamaan.

3.5 Yhteenveto ja tulkinta

Verkkopohjainen palautusjärjestelmän olisi voinut tehdä monella kielellä kuten ASP.NET ohjelmointikielellä. Tällöin olisi saanut myös hieman enemmän dynaamisuutta sivulle. Esimerkiksi ASP.NET:llä olisi pystytty ilmoittamaan heti jos lomakkeissa olevat tiedot olisi annettu väärin. Toteutettavan verkkosivun ollessa kuitenkin yksinkertainen, PHP ohjelmointikieli toteutti annetut vaatimukset. Ongelmia tuli vastaan muutamia, suurin osa niistä johtui tiedoston sisällön tarkistamisesta. PHP:llä tiedostojen sisällön tarkistus yleensä luonnistuu hyvin, kun sisällön voi ottaa muuttuunaan ja tarkistaa sitä kautta onko tietty asia mainittu tiedoston sisällä. Ongelmia tuli etenkin Excel tiedostojen tarkistuksessa, koska Excelin sisältö on binääriä.

Verkkopohjainen palautusjärjestelmä toimii hyvin ja täyttää vaatimusmäärittelyssä olevat vaatimukset, joten voidaan siis päätellä että PHP on hyvä kieli tiedostonimen ja tyyppin tarkistamiseen. Tiedoston sisällön tutkimisessa oli joitain ongelmia, etenkin Excel

muodossa, jossa tarkistetaan tiedoston tietyn solun arvo. Tämä arvo voi olla esimerkiksi numero, jolle annetaan minimi ja maksimi arvot.

4 TESTAUS

Verkkopohjainen palautusjärjestelmä testattiin aluksi neljällä harjoitustyöllä vuoden 2008-2009 Ohjelmistotuotannon harjoitustyöllä. Tällä testauksella varmistettiin, että palautusjärjestelmä toimi halutulla tavalla ja tarkisti halutut asiat. Tämän jälkeen tehtiin testaus 125 tiedostolla saman vuoden palautetuilla harjoitustöillä. Alla taulukossa 2 selvitetään kuinka monta harjoitustyötä läpäisi tarkastuksen.

Taulu 2: Testaustulokset

	Testitiedostoja	Tiedostotyyppi virhe	Nimeämis virhe	Sisällönvirhe	Onnistuneet
Harjoitustyö 1	39	4	1	25	9 (=23.1%)
Harjoitustyö 2 vaihe2	47	5	27	2	13 (=27.7%)
Harjoitustyö 2 vaihe 3	39	0	5	34	0 (=0%)
Yhteensä	125	9	33	61	22 (=17.6%)

Taulukossa 2 toisessa harjoitustyössä (kolmannessa vaiheessa) annetut tiedostojen palautus epäonnistui yleisestä syystä: Tiedostojen sisällä ei ollut opiskelijanumeroita ollenkaan, joka vaadittiin läpäisyyn. Tarkistusehtoja ”löysentämällä” (eli ei tarkistettu vaadittua opiskelijanumero harjoitustyöstä) huomattiin, että 34 harjoitustyötä läpäisi tarkistuksen ja vain viidessä oli virheitä nimeämisen takia. Tällöin palautettuja harjoitustöitä olisi noin 90% tämän palautuksen harjoitustöistä. Ensimmäisessä harjoitustyössä tuli paljon virheitä sisällön takia, koska opiskelijanumero oli väärin annettu tiedoston sisällä. Jos virheisiin ei lasketa helposti muutettavia virheitä nousisi kokonaispalautus prosentti lähes 90%. Näihin helposti korjattaviin virheisiin voidaan lukea tiedostonimen, tyyppin, ja yhden solun muuttaminen. Pienessä osassa testausmateriaalissa oli virheitä monessa kategoriassa (kahdessa tai kolmessa): nimeämisessä , tyyppissä ja sisällössä.

5 POHDINTA

Verkkopohjaisen palautusjärjestelmän toteutus ei ole uusi asia. Palautusjärjestelmä kuitenkin helpottaa huomattavasti tarkastajien työtä, kun tiedostot tulevat oikeassa muodossa eikä täyty tarkastajan sähköpostia. Lokitiedosto, joka kirjaa onnistuneet ja epäonnistuneet latausyritykset, näyttää heti ketkä ovat ladanneet tiedostonsa ja mihin aikaan.

Verkkopohjaista palautusjärjestelmää voidaan tulevaisuudessa käyttää muihinkin kursseihin, eikä vain Ohjelmistotuotannon kurssiin. Tällöin hieman muuntelemalla verkkopohjaisen palautusjärjestelmän INI tiedostoja voidaan palauttaa minkä tahansa tyyppisiä tiedostoja ja sen myötä myös tavallisia harjoituksia voidaan palauttaa. Sisällön tarkistamiseen voidaan kirjoittaa tiettyjä asioita esimerkiksi tiedoston sisällä pitää olla jokin tietty arvo tietyssä solussa, esimerkiksi solussa A1 minimi arvot ja maksimi arvot voidaan määritellä INI tiedoston avulla. Näin voidaan tarkistaa tiedosto eikä tarkastajan tarvitse kuin katsoa lokitiedosto ja onnistuneet palautukset sieltä.

Verkkosivun olisi voinut myös tehdä ASP.NET:llä, mutta siinä olisi tullut samat ongelmat tiedoston sisällön tarkistamisessa. Tämän takia oli sama kumman kielen olisi valinnut. PHP:ssä tiedoston sisällön tarkistamiseen oli monta ratkaisua: kaupallisesta, omaan koodiin. Oma koodi, jota aluksi yritettiin, ei toiminut kunnolla. Koodi toimi hyvin vain vanhoilla PDF tiedostoilla. Tämän takia jouduttiin turvautumaan avoimen lähdekoodiin.

6 JOHTOPÄÄTÖKSET

Verkkopohjainen palautusjärjestelmä auttaa tarkastajia ja nopeuttaa harjoitustöiden palautusta. Tämän verkkosivun avulla tarkastajien ei tarvitse välttämättä tarkistaa itse tiedostoja yksi kerrallaan, vaan tarkastaja voi katsoa lokitiedoston, jossa on hyväksytyt ja hylätyt palautukset. Lokitiedoston saa helposti pistettyä Exceliin, kuten vaatimuksena oli. Sieltä voi myös katsoa millaisia virheitä kullakin oppilaalla ilmeni. Verkkopohjaisessa palautusjärjestelmässä toteutui suurin osa vaadituista ominaisuuksista.

Testauksen perusteella voidaan huomata, että palautusjärjestelmä tarkistaa tiedostot tarkasti tiettyjen sääntöjen perusteella. Sääntöjä muuttamalla voidaan vaikuttaa läpi päässeiden harjoitustöiden kappalemäärään. Tällöin onnistuneita palautuksia on noin 90% harjoitustöistä. Tästä voidaan päätellä, että verkkopohjainen palautusjärjestelmä toimii hyvin harjoitustöiden palautukseen ja niiden tarkistamiseen.

LÄHTEET

[1] Ohjelmistotuotanto (CT20A4001) harjoitustyöt. [Ohjelmistotuotannon www-sivu] [viitattu 02.10.2009]. Saatavissa: <http://www.it.lut.fi/kurssit/08-09/CT20A4001/projects.html>

[2] Ohjelmistotuotanto (CT20A4001) harjoitukset. [Ohjelmistotuotannon www-sivu]. [viitattu 02.10.2009]. Saatavissa: <http://www.it.lut.fi/kurssit/08-09/CT20A4001/exercises.html>

[3] Powers, David. 2007. Using PHP to manage files, PHP Solutions: Dynamic Web Design Made Easy. s.179-209 ISBN: 978-1-4302-0275-2

[4] Powers, David. 2007. Uploading Files, PHP Solutions: Dynamic Web Design Made Easy. s.151-177 ISBN: 978-1-4302-0275-2

[5] Wolff, David. 2004. A web-based tool for managing the submission of student work, Journal of Computing Sciences in Colleges. Vol. 20 No. s. 144-153. ISSN:1937-4771

[6] Armstrong, Damon. 2005. Uploading Files, Pro ASP.NET 2.0 Website Programming. s. 473-493. ISBN: 978-1-4302-0104-5

[7] MacDonald, Matthew. Szpuszta, Mario. 2005. Files and Streams, Pro ASP.NET 2.0 in C# 2005. s. 471-502. ISBN: 978-1-4302-0064-2

[8] Powers, David . 2008. Using Php filters to Validate User Input, PHP Object-Oriented Solutions. s.121-166 ISBN: 978-1-4302-1012-2

[9] Mattocks, Scott. 2006. Installing PHP-GTK, Pro PHP-GTK. s.15-23 ISBN: 978-1-4302-0152-6

[10] Excel Explorer Pro. [Excel Explorer kotisivu], [viitattu 24.11.2009]. Saatavissa <http://www.eephp.com>

Liite 1: Projektisuunnitelma

Ositus ja toteutusvaiheet

Projekti palautetaan kun palautukseen tarkoitettu verkkosivu on valmis. Toteutusvaiheet sivun ominaisuuksista ovat aikataulussa.

Aikataulu

Alla taulukossa 1, on alustava aikataulu projektin etenemisestä ja ohjelmiston ominaisuuksien lisäämisestä verkkosivuun. Verkkosivun on tarkoitus valmistua 05.12.2009. Tiedostonimen tarkistus tapahtuu oppilaalta antaman opiskelijanumero mukaan. Tämän vuoksi versioon 0.4, tulee lomake, jossa kysytään tiedot oppilaalta ja tiedoston tarkistus.

Taulu 1 : Aikataulu

Sivun versio	Tarkennus	Päivämäärä
0.1	Sivu, jossa pystyy lataamaan tiedoston	08.10.2009
0.2	Tiedostotyypin tarkistus	10.10.2009
0.4	Tiedostonimen tarkistus	15.10.2009
0.4	Oppilas tiedot lomake, jossa kysytään opiskelijanumero ja oppilaan nimi	15.10.2009
0.5	Tiedoston sisällön tutkiminen PDF muodossa	25.10.2009
0.6	Tiedoston tutkiminen Excel muodossa	10.11.2009
0.7	Kurssin valinta	01.12.2009

Riskien kartoitus

Tässä osiossa kerrotaan riskeistä mitä voi tapahtua tässä projektissa, jotka saattavat aiheuttaa verkkosivun myöhästymiseen. Eräs riski on, että verkkosivu ei valmistu ajoissa, koska määrittämiä ja ominaisuuksia pitää lisätä, jolloin aikataulu venyy. Lisäksi verkkosivun ohjelmointikieli saattaa olla huono tarkoitukseen, jolloin pitää vaihtaa ohjelmointi kieli, joka aloittaa ohjelmoinnin alusta.

Testaus

Verkkosivun testaus suoritetaan, joka version jälkeen. Testauksessa käytetään aluksi hyväksytyjä tiedostotyyppisiä kuten pdf ja Excel. Tiedostonimi annetaan oikealla tavalla. Kun näitä pystytään lataamaan, annetaan virheellisiä tiedostoja kuten tekstitiedostoja ja nimetään ne väärin. Testauksessa käydään läpi kaikki virheet yksittäin ja virheiden kombinaatiosta.

Lopullinen testaus verkkosivusta tehdään ohjelmistotuotannon edellisten vuosien varrella kertyneillä materiaaleilla eli harjoitustöillä. Lopullisessa testauksessa käytetään 125 tiedostoa (PDF ja Excel tyyppiä), jotka saadaan vuoden 2008-2009 Ohjelmistotuotannon kurssista. Näiden avulla katsotaan kuinka hyvin verkkosivu tarkistaa tiedostot ja niiden sisällön. Verkkosivun pitää antaa virhe ilmoitus virheellisissä tapauksissa.

Menetelmät ja standardit

Käytettävät ohjelmointikieliset ovat HTML ja PHP. Ohjelmoinnissa nimeämiskäytäntö tehdään englanniksi, jolloin kaikki mahdolliset muuttujat ovat englanniksi. Tällöin myös INI tiedosto, joka määrittelee mitä tarkistetaan, on myös englanniksi. Ilmoitukset mahdollisesta tiedoston palautuksesta annetaan suomeksi. Kommentointi on myös englanniksi tehty. INI tiedosto on standardin mukainen, (malli: http://my.execpc.com/~iniman/what_is.html), vain sektion takit puuttuu eli "[]" -

merkit, mutta niitä ei tarvitse, koska jokaiselle harjoitustyön palautukselle tehdään oma kansio. PHP koodissa muuttujat ovat nimetty C-kielen mukaisesti, aloittaen pienellä kirjaimella, ja funktioiden nimet ovat myös pienellä kirjoitettuna esimerkiksi ”read_excel”.

Käytettävät työkalut ovat tekstieditori (verkkosivun kirjoittamiseen) ja verkkoselain verkkosivun katselemiseen ja testaamiseen. Versioista vain viimeisen versio on tarkoitettu käyttää tiedostojen palautukseen. Viimeisessä versiossa on kaikki tarvittavat tiedoston tarkistukseen.

Projektin toimitus ja lopetus

Projekti (verkkosivut) toimitetaan opettajalle tiedostoina, jotka voidaan sitten asentaa / tallentaa koulun palvelimelle. Tällöin annetaan versio 0.6, jossa tiedostojen tallennus ja palautus toimii hyvin. Versiossa eri kursseille ja harjoitus töille palautettavat tiedostot tallentuu eri paikkoihin. Projekti lopetetaan sen jälkeen kun tuote on testattu viime vuoden tiedostoilla ja sen jälkeen toimitettu opettajalle.

Liite 2: Vaatimusmäärittely

Nykytilankuvaus

Ohjelmisto (verkkosivu) toteutetaan Lappeenrannan teknillisen yliopiston Unix palvelimelle, johon verkkosivun kautta oppilaat voivat tallentaa omia harjoitustöitään palvelimelle. Opettaja voi halutessaan käydä katsomassa tiedostoja. Verkkosivun tarkoitus on helpottaa opettajan työtä tarkistamalla tiedostojen nimet ja tiedostotyyppi ennen tiedoston varsinaista palautusta. Lisäksi tiedoston sisältö tarkistetaan, jotta tarvittavat kentät (tiedot) on täytetty. Ohjelmisto halutaan tehdä nopeuttaakseen ja helpottaakseen tehtävien / harjoitustöiden palautusta. Tiedoston ollessa puutteellinen esimerkiksi tiedostotyyppi on väärä, siitä pitää ilmoittaa oppilaalle ennen tiedoston tallennusta palvelimelle.

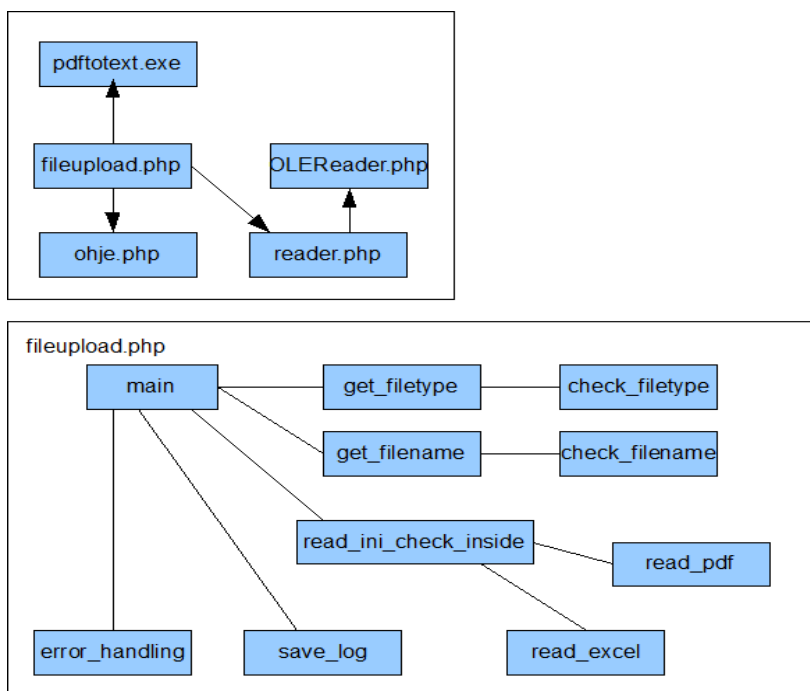
Tavoitetilan kuvaus

Oppilaat valitsevat kurssin, jolle haluavat tiedoston palauttaa. Tavoitetilassa verkkosivun kautta he pystyvät lataamaan palvelimelle tiedostoja tietyissä muodoissa ja tyyppissä. Oppilaat antavat opiskelijanumeron ja nimensä ennen tapahtuvaa tiedoston lataamista palvelimelle. Tämän jälkeen verkkosivu tarkistaa tiedostonimen ja tiedostotyypin annettujen tietojen mukaan. Tämän jälkeen sivu tarkistaa myös tiedoston sisällön. Tiedoston ollessa puutteellinen, siitä ilmoitetaan käyttäjälle.

Arkkitehtuuri

Palautusjärjestelmän arkkitehtuuri on esitetty kuvassa 1: arkkitehtuuri. Järjestelmään kuuluu kolme tiedostoa fileupload.php, ohje.php, reader.php, pdftotext.exe ja OLEReader.php. Näistä fileupload.php on pääsivu, jossa kysytään käyttäjän tietoja lomakkeen avulla ja sen jälkeen mennään oikeaan aliohjelmaan (pdftotext tai reader.php). Ohje.php:n mennään kun painetaan ”Ohje” linkkiä pääsivusta. Pdftotext ja reader.php:tä käytetään aliohjelmien kautta tarkistamaan PDF ja Excel tiedostoja.

Itse pääsivun sisällä pääfunktiossa (main), haetaan aluksi INI tiedostosta tiedostonimi ja tyyppi (get_filetype, get_filename), jotka sitten tarkistetaan (check_filetype ja check_filename). Tarkistuksen ollessa virheetön, menee palautusjärjestelmä tarkistamaan tiedoston sisällön. Save_log funktio tallentaa lokiin tapahtumat palautetuista tiedostoista sekä virheen (tai virheiden) sattuessa. Error_handling funktio antaa käyttäjälle tietoja tapahtuneista virheistä tai onnistumisen jälkeen antaa tekstin ”Palautus onnistui”.

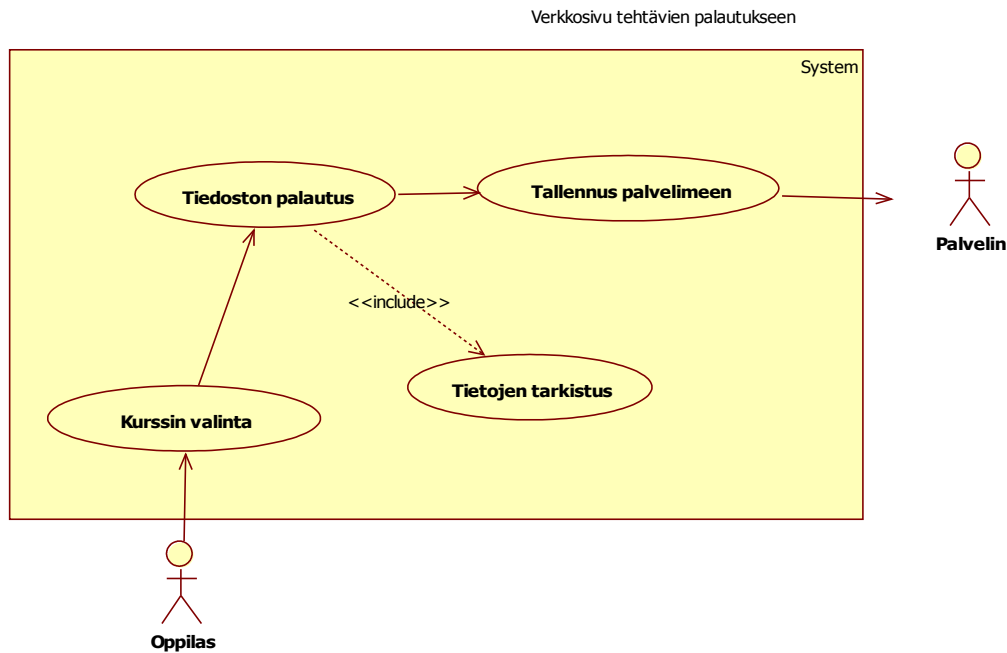


Kuva 1: Arkkitehtuuri

Käyttäjärühmät

Käyttäjärühmiä on yksi: oppilaat. Joiden erityistarve on ladata tiedostoja palvelimelle. Verkkosivu ilmoittaa mahdollisista puutteista tiedostosta oppilaille. Oppilaat voivat halutessaan palauttaa useille kursseille tiedostoja tämän sivun kautta, joten käyttäjille pitää olla kurssien valinta sivu. Mahdollinen toinen käyttäjärühmä on opettajat, jotka hakevat tiedostoja verkkosivun kautta. Tämä sivu kuitenkin toteutetaan myöhemmin.

Käyttötapauskaaviot



Kuva 2: Käyttötapauskaavio

Kuvassa 2 on hahmoteltu käyttötapauskaavio. Oppilas menee verkkosivulle ja valitsee kurssin, johon haluaa työnsä (tiedostonsa) palauttaa. Oppilas tässä vaiheessa täyttää omat tietonsa kenttään jossa kysytään nimi ja opiskelijanumero. Näitä tietoja käytetään tarkistamaan tiedoston oikeellisuus ja sen sisältö. Tämän jälkeen oppilas valitsee tiedoston omalta koneeltaan. Verkkosivu tekee tietojen tarkistuksen tiedostolle ja katsoo onko se nimetty oikein ja onko tiedostotyyppi oikea. Tarkistuksen jälkeen tallennetaan tiedosto palvelimeen kurssille ja harjoituksille tarkoitettavaan kansioon. Jokaisesta palautusyrityksestä pitää kirjata lokirivi johonkin.

Jokaisella eri kurssilla on oma hakemistonsa, johon tulee oma lokitiedosto. lokitiedostoon kirjataan kaikki palautusyritykset. Tiedosto pitää sisällään opiskelijan nimen, opiskelijanumero, tiedostonimen, virheet, päivämäärän ja kellonajan. Tiedot pistetään sellaiseen muotoon, jotta ne saadaan helposti Exceeliin auki.

Käyttötapaus 1

ID	UC-1
----	------

Nimi ja versio	Kurssin valinta
Suorittajat	Oppilas
Esiehdot	Oppilas on mennyt sivulle
Kuvaus	Oppilas valitsee halutun kurssin, jolle haluaa palauttaa tiedoston
Poikkeukset	
Lopputulos	Oppilas on valinnut kurssin
Muut vaatimukset	

Käyttötapaus 2

ID	UC-2
Nimi ja versio	Tiedoston palautus
Suorittajat	Oppilas
Esiehdot	Oppilas on mennyt sivulle, ja valinnut kurssin, jolle haluaa palauttaa tiedoston / työnsä
Kuvaus	Oppilas valitsee tiedoston koneeltansa, jonka haluaa palauttaa. Kirjoittaa opiskelijanumeronsa ja nimensä lomakkeeseen
Poikkeukset	-oppilas keskeyttää tiedoston latauksen,
Lopputulos	Oppilas on valinnut palautettavan tiedoston
Muut vaatimukset	

Käyttötapaus 3

ID	UC-3
Nimi ja versio	Tiedoston tarkistus
Suorittajat	Verkkosivu
Esiehdot	Oppilas yrittää ladata tiedostoa (tiedoston palautus)
Kuvaus	Verkkosivu tarkistaa tiedostotyypin ja nimen. Lisäksi tarkistaa osan tiedoston sisällöstä opiskelijan antamien tietojen perusteella
Poikkeukset	
Lopputulos	Tiedosto hyväksytään tai hylätään sen mukaan miten tarkistus on mennyt
Muut vaatimukset	

Käyttötapaus 4

ID	UC-4
Nimi ja versio	Tiedoston tallennus
Suorittajat	Verkkosivu
Esiehdot	Oppilas on onnistuneesti ladannut tiedoston.
Kuvaus	Tiedosto tallennetaan oikeaan hakemistoon, kurssin valinnasta riippuen
Poikkeukset	
Lopputulos	Tiedosto tallentuu koulun palvelimelle
Muut vaatimukset	

Käyttötapaus 5

ID	UC-5
Nimi ja versio	Virhe
Suorittajat	Verkkosivu
Esiehdot	Oppilas on antanut puutteellisen tiedoston
Kuvaus	Verkkosivu näyttää opiskelijalle puutelistan, jossa näkyy puutteet tiedostosta
Poikkeukset	
Lopputulos	Oppilaalle on ilmoitettu puutteista
Muut vaatimukset	

Rajoitteet

ID	Pvm	Nimi	Kuvaus
1	02.10.2009	Verkkosivu	Verkkosivun pitää toimia kaikilla selaimilla
2	02.10.2009	Verkkosivu	Verkkosivun toteutuskielenä käytetään alustavasti PHP:tä ja html:ä
3	06.10.2009	Verkkosivu	Järjestelmän tulee toimia ubuntu-linuxissa
4	20.10.2009	Verkkosivu	Järjestelmä ylläpitää loki tiedostoa
			Järjestelmä toimii INI tiedoston avulla

Olettamukset

Verkkosivun toteutuksessa käytettävää kieltä ei ole määritelty, joten oletamme että koulun palvelin tukee kaikkia mahdollisia kieliä kuten ASP.NET:iä ja PHP:tä, joilla sivu voidaan toteuttaa. Oletetaan, että tarvitaan vain oppilaille palautustyökalu (verkkosivu), joka tarkistaa tiedoston ja kertoo puutteet. Mahdollinen sivu opettajalle tehdään jos aikaa jää. Oppilaat pitää tunnistaa jollakin tavalla. Tämä toteutetaan tavalla, jossa opiskelija antaa opiskelijanumeron ja nimen jotta voidaan tarkistaa tiedoston sisältö oikein.

Toiminnalliset vaatimukset

Lataaminen

Oppilaan pitää pystyä lataamaan tiedostoja palvelimelle. Tiedoston lataaminen tehdään yksi tiedosto kerrallaan. Tiedostot joita voi ladata palvelimelle ovat joko Excel tai PDF tiedostotyyppiä.

Tarkistaminen

Tiedoston palautuksen yhteydessä tarkistetaan tarvittavat asiat kuten tiedostonimi, tiedostotyyppi ja sisältö. Nämä tarkistukseen asiat otetaan INI tiedostosta, joita on tarkoitettu olla jokaiselle harjoitustyölle omansa. Tarkistuksessa ilmenneet virheet näytetään oppilaalle, jotta oppilas voi korjata virheet. Virheet myös kirjataan lokitiedostoon, josta tarkastaja voi tarkastella mitä virheitä on tapahtunut ja kenelle.

Tapahtumien kirjaaminen

Lataamisista pidetään kirjaa, hyväksytyistä ja hylätyistä tiedostoista tehdään erillinen loki tiedosto, jossa näkyvät palauttajan nimi, opiskelijanumero, virhenumerot, päivämäärä ja aika. Virhenumerot ja niiden vastaavat virhe ilmoitukset on alla, kolme ensimmäistä virhettä ovat lomakkeeseen liittyen: Lomakkeeseen liittyviä virheitä ei kirjoiteta lokiin.

11 – Etunimi puuttuu lomakkeesta

12 – Sukunimi puuttuu lomakkeesta

13 – Opiskelijanumero puuttuu lomakkeesta

14 – Tiedostotyyppi on väärä

17 – Tiedostonimi on väärä

20 – Tiedoston sisällössä virhe. Sisältö ei vastaa annettua arvoa INI tiedostosta.

Ei toiminnalliset vaatimukset

Testattavuus

Verkkosivu testataan aluksi jokaisessa versio vaiheessa. Lopullinen testaus suoritetaan edellisten kurssien saaduista tiedostoista. Jolloin nähdään voiko verkkosivua käyttää ohjelmistotuotannon kurssin palautuksiin.

Tietoturva

Verkkosivun tullessa yliopiston palvelimelle pitää katsoa tietoturvaa tarkemmin. Tällöin esimerkiksi väriä tiedostoja ei saisi pystyä antamaan palvelimelle. Tämä tarkistus tehdään siten, että avataan tiedosto ennen tallennusta ja katsotaan alkaako tiedosto tietyillä kirjaimilla esimerkiksi PDF tiedostoissa se on <!PDF . Tämän avulla voidaan sulkea mahdollisuus siitä, että annetaan valetiedostoja. Lisäksi käyttäjän syöte lomakkeessa oleviin kentissä pitää tarkistaa. Muuten käyttäjä voi antaa lomakkeisiin esimerkiksi koodia, joka saattaisi olla haitallista koulun palvelimelle.

Ylläpidettävyys

Verkkosivun pitää olla helposti ylläpidettävä, koska tiedoston nimeämiskäytäntö voi vaihdella kursseilla vuosittain erilaiseksi. Ylläpidettävyys tehdään INI tiedoston avulla, joten koodia ei tarvitse paljoa muuttella, vain INI tiedostoa muutetaan jos tarkastukseen liittyvät nimeämiskäytännöt muuttuvat.

Muunnettavuus

Verkkosivua voidaan käyttää muissakin kursseissa harjoitustöiden tai harjoitusten palautuksiin. Tämä muunnettavuus onnistuu INI tiedoston avulla, johon kirjoitetaan tarkistettavan tiedoston nimi esimerkiksi ”{opnro}{etunimi}”, jolloin tiedoston nimi koostuu kahdesta osasta: opiskelijanumero ja etunimi. Nimeämiskäytäntöä voidaan siis muuttaa halutulla tavalla. Palautusjärjestelmä antaa tällä hetkellä palauttaa vain PDF ja

Excel tiedostoja, joten paljoo valinnan varaa ei ole tiedostotyypin muunnettavuudessa.
Tosin tämän saisi myös muutettua vain parilla rivillä koodilla itse PHP sivuihin.