

eMo  
eAssari Moodle-ympäristössä  
Testaussuunnitelma

Helsinki 15.12.06  
Ohjelmistotuotantoprojekti  
HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteenlaitos

## Kurssi

581260 Ohjelmistotuotantoprojekti (9+1op/6ov)

## Projektiryhmä

Järviniitty Juho

Karkulahti Ossi

Katainen Riku

Tverin Teemu

Vainio Ville-Pekka

## Asiakas

Laine Harri

## Vastuhenkilö

Taina Juha

## Ohjaaja

Halko Antti

## Kotisivu

<http://www.cs.helsinki.fi/group/emo/>

## Versiohistoria

<i>Versio</i>	<i>Päiväys</i>	<i>Tehdyt muutokset</i>
0.1	01.11.2006	runko
0.2	04.11.2006	1,2,3,5, 6,7
0.4	06.11.2006	8 (ensimmäinen esitelty versio)
0.5	10.11.2006	Korjauksia, 9
0.6	13.11.2006	EUCT, korjauksia, toinen esitelty versio
1.0	20.11.2006	Kattavuus, virheiden korjaus
1.01	8.12.2006	Muotoilu.Testiraportti
1.02	12.12.2006	Testiraportit
1.1	15.12.2006	Lopullinen versio

# Sisällys

1 Johdanto.....	1
2 Sanasto.....	1
3 Yksikkötestaus.....	1
3.1 Testausstrategia.....	2
3.2 Testauksen kattavuus.....	2
3.3 Toiminta virhetilanteessa.....	3
4 Integrointitestaus.....	3
4.1 Testausstrategia.....	4
4.2 Toiminta virhetilanteissa.....	4
5 Järjestelmätestaus.....	5
5.1 Testausstrategia.....	5
5.2 Järjestelmätestauksen toteutus.....	5
5.3 Laajennetut (EUCT) käyttötapaukset.....	5
5.4 Toiminta virhetilanteissa.....	6
6 Hyväksymistestaus.....	6
7 Työskentelytavat.....	6
7.1 Testauksen dokumentointi.....	6
7.2 Työkaluista.....	7
8 Testausaikataulu.....	7
9 Testausraportti.....	9
9.1 Yleistä.....	9
9.2 Testiympäristö.....	9
9.3 Yksikkötestaus ja integrointitestaus.....	9
9.3.1 Tehtävätyyppi (OKREGEXP).....	10
9.3.2 Tehtävätyypin lisääminen.....	12
9.3.3 Tehtävätyypin poistaminen.....	14
9.3.3.1 showremovetype.php.....	14
9.3.3.2 processremovetype.php.....	16
9.3.4 Käyttöoikeudet.....	17
9.3.4.1 Laatija.....	17
9.3.4.2 Opettaja.....	18
9.3.4.3 Opiskelija.....	18
9.4 Järjestelmätestaus.....	18
9.4.1 Opiskelijan käyttötapaukset.....	18
9.4.1.1 UCS1: Tehtäviin tutustuminen.....	18
9.4.1.2 UCS2: Tehtävään vastaaminen.....	19
9.4.2 Opettajan käyttötapaukset.....	19
9.4.2.1 UCT1: Tehtävän lisääminen.....	19
9.4.2.2 UCT2: Tehtävän muokkaaminen.....	20
9.4.2.3 UCT3: Tehtävän poistaminen.....	20
9.4.2.4 UCT4: Tehtävien selailu.....	21
9.4.3 Laatijan käyttötapaukset.....	21
9.4.3.1 UCT5: Uuden tehtävätyypin lisääminen.....	21
9.4.3.2 UCT6: Tehtävätyypin muokkaaminen.....	22
9.4.3.3 UCT7: Tehtävätyypin poistaminen.....	22
9.5 Muita järjestelmätestejä.....	23
9.6 Vaatimusten toteutuminen.....	24
9.6.1 Järjestelmävaatimukset.....	24
9.6.2 Käyttäjävaatimukset.....	25
9.7 Hyväksymistestaus.....	25
9.8 Testauksen yhteenveto.....	25



# 1 Johdanto

Tämä dokumentti on testaussuunnitelma eMo-ryhmän ohjelmistotuotantoprojektille, jonka tarkoituksena on siirtää eAssari-järjestelmä moodle-ympäristöön. Tarkoituksena on luoda järjestelmä, joka tekee uusien eAssari-tehtävätyyppien lisäämisen Moodleen helpoksi. Ohjelmisto laaditaan kurssien vetäjille sekä opiskelijoille.

Testaussuunnitelmasta käy ilmi, miten projektissa tuotettavan järjestelmän testaus suoritetaan. Testauksen tarkoituksena on osoittaa järjestelmän kehittäjille sekä asiakkaalle, miten ohjelmisto täyttää sille asetetut vaatimukset ja löytää ohjelmistosta virheitä sekä puutteita, joiden johdosta ohjelmisto ei toimi, toimii väärin tai ei vastaa sille vaatimusmäärittelyssä asetettuja määrittelyjä.

Testaus jakautuu kolmeen vaiheeseen: yksikkötestaukseen, integrointitestaukseen sekä järjestelmätestaukseen.

- Yksikkötestauksessa testataan komponentit ja niiden toimivuus
- Integrointitestauksessa testataan yksikkötestauksen läpäisseiden komponenttien rajapinnat
- Järjestelmätestauksessa testataan järjestelmän toimivuus loppukäyttäjän näkökulmasta.

Tämä dokumentti sisältää myös ns. Testausraportin. Testausraportti on tämän dokumentin luku numero 9

## 2 Sanasto

**Luokkatestausta:** katso yksikkötestaus

**Arvoalueanalyysi:** löydetään kaikkien testattavien syötteiden joukosta osa-arvoalueita.

**EUCT:** Järjestelmätestaus toteutetaan Extended Use Case-testinä , jossa käytetään niin kutsuttuja laajennettuja käyttötapauksia. Yksi laajennettu käyttötapaus määrittelee joukon skenaarioita, jotka kertovat kuinka järjestelmä toimii tietyillä syötteillä, eli mitä tuloksia se palauttaa

## 3 Yksikkötestaus

Yksikkötestaus (myös ”luokkatestausta”) on matalimman tason testausta. Sitä tehdään ohjelmiston

jakamattomille osille ja sen tarkoituksena on testata komponenttien, luokkien ja luokkaryppäiden toimivuus ja virheettömyys. Yksikkötestaus tapahtuu yksinkertaisesti metodiensa kautta, kutsumalla niitä erilaisin syöttein. Mikäli metodeilla on keskinäisiä riippuvuuksia, on nekin testattava. Yksikkötestauksen suorittaa pääsääntöisesti ko. metodin ohjelmoija ohjelmointityön yhteydessä.

### **3.1 Testausstrategia**

eMo-projektissa yksikkötestaus suoritetaan siten, että kaikki ei-triviaalit metodit pyritään suorittamaan sekä kelpoisilla että kelpaamattomilla syötteillä. Ei-triviaalilla metodilla tarkoitetaan sellaista metodia, joka tekee muutakin kuin asettaa paikallisen arvon tai palauttaa sen.

Parametrisoitavat metodit täytyy testata. Ne testataan kutsumalla niitä erilaisilla parametreilla. Lisäksi metodien käyttäytyminen voi muuttua riippuen sen tilasta, joten metodien testaus tulisi toistaa myös eri tiloissa.

Sopivien syötteiden löytämiseen käytetään apuna arvoalueanalyysia:

Metodille annettavat testiparametrit muodostavat eräänlaisia luokkia: saman luokan sisällä olevien parametrien oletetaan tuottavan samankaltaisen käyttäytymisen metodissa. Testiparametreiksi valitaan metodin ulkoisten tai sisällä olevien rajoitusten lähellä olevat arvot (raja itse ja raja +-1), sekä jokin arvo luokan arvoalueen ”keskeltä”. Esimerkiksi if-lauseet ja silmukat tuottavat helposti tällaisia rajoja. Kelpaamattomilla syötteillä varmistetaan, että metodi toimii niiden kanssa tarkoituksenmukaisella tavalla.

### **3.2 Testauksen kattavuus**

Yksikkötestaus pyritään suorittamaan lausekattavasti, mikä tarkoittaa, että testattavasta kohteesta käydään muodostetusta suunnatusta verkosta käydään kaikki solmut läpi. Esimerkiksi metodista suunnattu verkko muodostetaan seuraavalla tavalla:

- Metodiin tulo on lähtösolmu ja siitä poistuminen maalisolmu. Molempia on vain yksi.
- Kutakin lausetta kuvaa solmu ja siirtymistä lauseesta toiseen kuvaa verkon särmä.
- Ehtolauseen tapauksessa siitä lähtee kaksi särmää.
- Jos lause on return, lähtee siitä särmä maalisolmuun.

Luokasta testataan metodit, paikalliset tietorakenteet ja rajapintojen toteutus. Metodeista testataan koodi, silmukat ja sisäiset tietorakenteet. Lisäksi suoritetaan virhetilanteiden testaus. Mahdolliset virhetilanteet luodaan ja tarkistetaan kuinka metodi selviää niistä. Rajapintojen toteutusta testattaessa tarkistetaan, että metodit ja attribuutit toimivat yhdessä, jotta ne voivat tuottaa luokan rajapinnan palvelut.

Lausekattavuus voidaan laskea seuraavalla kaavalla:

$LK=TL/AL$ , missä LK on lausekattavuus, TL on testeissä käytetyn yksikön lauseiden lukumäärä AL on kaikkien yksikön lauseiden lukumäärä.

Toisin sanoen lausekattavuus kertoo, kuinka suuressa osassa testattavan yksikön lauseita on käyty, kun testit on suoritettu. Testauksessa tulisi pyrkiä mahdollisimman lähelle 100%:n lausekattavuutta mutta projektissa voidaan tyytyä noin 80%:n lausekattavuuteen. Hyväksymiskriteerinä myös että kaikki yksikön toiminnot ja mahdolliset tilat testattu. Samoin kaikki poikkeustilanteet täytyy testata.

### **3.3 Toiminta virhetilanteessa**

Virheen ilmetessä tulee testattava luokka poistaa testauksesta, virhe korjata ja tämän jälkeen luokan/metodin testaus aloitetaan alusta.

## **4 Integrointitestaus**

Integrointitestauksen tarkoituksena on testata luokkien väliset rajapinnat ja niiden toimivuus. Erityisesti eMo-projektissa korostuu eMo-järjestelmän ja Moodlen välinen plugin-rajapinta ja sen käyttäminen eMo-järjestelmän toiminnallisuuden Moodleen liittämiseen.

Luokat etenevät integrointitestaukseen niiden läpäistyä yksikkötestauksen, eli on todettu että ne toimivat oikein yksinään. Integrointitestauksen perusajatuksena on varmistaa komponenttien toiminta yhteistyössä toistensa kanssa ja paikantaa mahdollisia virheitä komponenttien välisistä rajapinnoista.

Integrointitestauksessa testataan kaikki komponenttien väliset rajapinnat. Käytännössä tämä toteutetaan siten, että komponentit liitetään ja testataan yksi kerrallaan, kunnes kaikki liitettävät komponentit on liitetty ja testattu. Tämän jälkeen testataan vielä, toimivatko näin saadut suuremmat

kokonaisuudet yhdessä oikein.

## **4.1 Testausstrategia**

Integroititestausta uusille komponenteille pyritään tekemään heti kun uusi komponentti on valmis, eli se on ohjelmoitu, dokumentoitu ja yksikkötestattu. Uuden luokan valmistuessa se integroidaan sopivilta osin jo testattuun ryppäeseen. Tällöin testataan rajapinnat vain uuden komponentin osalta, jo tehtyjä testejä ei siis tarvitse suorittaa uudelleen uusien komponenttien valmistuessa ellei komponentti itsessään vaikuta olennaisilta osin muun ryppään komponenttien toimintaan.

Tavoitteellisesti integroititestausta pyritään tekemään alhaalta ylös (bottom up) -strategialla. Näin pyritään aloittamaan testaus mahdollisimman alhaisen tason komponenteista ja edetä integroinnissa korkeamman tason komponentteihin. Jotta strategia olisi mahdollinen tulee aikataulutusta projektisuunnitelmassa hoitaa siten että komponenttien toteutus aloitetaan mahdollisimman alhaisen tason komponenteista edeten korkeamman tason komponentteihin (mahdollisuuksien mukaan).

Kahden yksikön osalta integroititestausta on valmis kun yksiköiden välinen yhteistyö on testattu, poikkeukset on testattu, ja kaikki mahdolliset kutsuttavan aiheuttamat sivuvaikutukset järjestelmään on testattu.

Järjestelmä ei ole valmis järjestelmätestaukseen integroititestausta osalta ennen kuin integroititestausta on valmis koko järjestelmän osalta. Integroititestauksessa ei pyritä tekemään järjestelmätestauksen piiriin kuuluvia testejä (käyttötapausten testaus) vaan pitäydytään rajapintojen testauksessa.

## **4.2 Toiminta virhetilanteissa**

Mikäli integroititestauksessa huomataan testaajan korjattavissa oleva virhe, se korjataan välittömästi. Jos virhe on vakavuudeltaan merkittävä ja integroititestausta tekevä ryhmäläinen ei ole vastuussa testattavan luokan toteuttamisesta, eikä osaa korjata virhettä omin neuvoin, otetaan yhteyttä luokan toteuttajaan (yhteydenotto tapa valitaan kiireestä riippuen, mielellään kuitenkin henkilökohtaista sananvaihtoa) ja sovitaan luokan korjaamiseen liittyvistä seikoista.



# 5 Järjestelmätestaus

Järjestelmätestaus voidaan aloittaa kun integrointitestaus on suoritettu loppuun.

Järjestelmätestauksessa ei oteta enää kantaa ohjelmiston toteutustapaan, vaan ohjelmistoa käytetään järjestelmätestauksessa vain sen tarjoaman käyttöliittymän kautta.

Järjestelmätestaus on käytännössä vaatimusmäärittelyssä määriteltyjen vaatimusten, sekä toiminnallisten että ei-toiminnallisten, toteutumisen tarkistamista. Toki järjestelmätestaus voi sisältää myös laadun arvioimista muista näkökulmista. Testaus tapahtuu kohdeympäristössä, jotta voidaan olla varmoja vaatimusten toteutumisesta asiakkaan ympäristössä. Ohjelmisto läpäisee järjestelmätestauksen jos se toteuttaa kaikki vaatimusmäärittelyssä määritellyt vaatimukset.

## 5.1 Testausstrategia

Järjestelmätestaus toteutetaan Extended Use Case-testinä (EUCT), jossa käytetään niin kutsuttuja laajennettuja käyttötapauksia (extended use cases). Yksi laajennettu käyttötapaus määrittelee joukon skenaarioita, jotka kertovat kuinka järjestelmä toimii tietyillä syötteillä, eli mitä tuloksia se palauttaa. Syötteiden ja tulosteiden yhdistelmistä rakennetaan päätöstaulu. Jokainen päätöstaulun rivi kertoo yhden toimintatavan syötteet ja tulokset. Jokainen sarake kertoo käyttötapauksen yhden syötteen tai tuloksen nimen ja tyypin.

## 5.2 Järjestelmätestauksen toteutus

Emoa testataan suoraan sen käyttöliittymän kautta.

Testaus toteutetaan käymällä läpi jokainen testattava EUCT-käyttötapaus. Testitulokset sekä havaittavat ongelmat kirjataan ylös. Lisäksi lopullista järjestelmää verrataan vaatimusmäärittelydokumentissa esitettyihin vaatimuksiin ja varmistetaan, että kaikki vaaditut vaatimukset toteutuvat.

## 5.3 Laajennetut (EUCT) käyttötapaukset

Esimerkki käsittelee järjestelmään kirjautumista salasanalla.

N	Salasana	Salasana syötetty	Järjestelmän kuittaus	Tila	Vastausviesti	Järjestelmän toiminta
---	----------	-------------------	--------------------------	------	---------------	-----------------------

1	1234			Anna salasana	Uusi sisäänkirjautuminen
2	1234	1333	closed	Salasana väärin	Uusi sisäänkirjautuminen
3	1234	1234OK	open	Salasana oikein	Siirry seuraavalle sivulle
4	1234	1234	closed	Ei yhteyttä tietokantaan	Kirjautuminen estyy

Vaatusmäärittelyssä määriteltyjen käyttötapausten EUCT-taulukot on toimitettu liitteenä.

## 5.4 Toiminta virhetilanteissa

Mikäli EUCT-testissä havaitaan virhe, pyritään se korjaamaan mahdollisuuksien rajoissa. Jos projektin aikaraja tulee vastaan, kirjataan virhe ohjelmiston ylläpidodokumenttiin ja dokumentoidaan keinot sen sietämiseen.

# 6 Hyväksymistestaus

Kun järjestelmätestaus on suoritettu luovutetaan ohjelmisto asiakkaalle hyväksymistestaukseen. Asiakkaan on tarkoitus testata tuotetta ja arvioida sitä. Testauksessa huomattuja puutteita voidaan korjata aikataulun, vaatimusmäärittelyn ja asiakkaan kanssa tehtyjen sopimusten puitteissa.

# 7 Työskentelytavat

Testaus tulisi pyrkiä tekemään siten, että se voidaan halutessa toteuttaa sellaisenaan uudelleen. Mielessä tulisi pitää, että tuotteen on luvattu toimivan laitoksen Moodle-ympäristössä. Tämä aiheuttaa sen, että käytössä on vanhempi versio PHP-ohjelmointikielestä. Vaatusmäärittelyssä on myös korostettu projektin jatkokehittävyyttä, tämä seikka tulisi myös muistaa testauksessa. Mitä paremmin testaus on dokumentoitu, erityisesti virheiden osalta, sitä helpompaa projektin jatkokehitys on tuleville ohjelmistotuotantoprojektiryhmille.

## 7.1 Testauksen dokumentointi

Testauksessa mahdollisesti löytyvät virheet ja puutteet, joita ei kyetä korjaamaan projektin kuluessa dokumentoidaan kattavasti ylläpidodokumenttiin. Tähän dokumentointiin kuuluu ainakin virhetilanteen kuvaus, virheen vakavuuden arviointi sekä ohjeet virhetilanteen välttämiseen ja siitä toipumiseen.

Testeistä kirjoitetaan raportti josta käy ilmi seuraavat asiat: testaaja, testattu toiminta, syötteet,

odotettu tulos ja tulos sekä mahdolliset havaitut virheet. Nämä raportit koostetaan tämän dokumentin luvuksi numero 9.

Esimerkki raportoidusta testistä:

<b>Testaaja:</b>	Teemu
<b>Testin kohde:</b>	Tehtävätyypin lisäys ja tietokantamuutokset
<b>Miten testattiin:</b>	Lisättiin tehtävätyyppi
<b>Odotetut tulokset:</b>	Tehtävätyyppi näkyy moodlessa, tietokanta päivittyy oikein
<b>Tulos:</b>	Tehtävätyyppi näkyy, tietokantamuutokset ok.
<b>Havaitut virheet:</b>	--
<b>Testin tulos:</b>	Testi on onnistunut.

## 7.2 Työkaluista

Moodle sisältää yksikkötestausalustan. Lisätietoa tästä alustasta sekä käyttöohjeet löytyvät seuraavasta osoitteesta:

[http://docs.moodle.org/en/Development:Unit\\_tests](http://docs.moodle.org/en/Development:Unit_tests)

Testauksen kannalta on kuitenkin tärkeintä, että jokainen yksikkö on testattu kattavasti ja aukottomasti. Jos testauksen pystyy omasta mielestään tekemään tehokkaasti jollain muulla tyylillä kuin esimerkiksi tällä Moodlen alustalla, saa ryhmäläinen käyttää tätä tyyliä yksikkötestauksessa. Tärkeintä on siis testauksen kattavuus ja tarkka dokumentointi.

# 8 Testausaikataulu

Projektisuunnitelmassa testaukseen on varattu aikaa viikon 44 alusta viikon 50 alkuun. Se tekee noin 6½ viikkoa. Testausta tehdään lomittain toteutuksen kanssa.

Testaus on aloitettu laatimalla testaussuunnitelma.

Yksikkötestausta tehdään käytännössä koko toteutuksen ajan.

Integrointitestaus aloitetaan kun yksikkötestauksessa on hyväksytty integroitavia yksiköitä.

Järjestelmätestaus voidaan aloittaa integrointitestauksen valmistuttua, käytännössä järjestelmätestauksesta saatetaan joutua palaamaan vielä integrointitestaukseen.

Hyväksymistestauksen suorittaa asiakas valmiille ohjelmistolle.

	<b>44</b>	<b>45</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>50</b>
<b>Testaussuunnitelma</b>	xx						
<b>Yksikkötestaus</b>		xx	xx	xx	xx	xx	xx
<b>Integrointitestaus</b>				x	xx	xx	xx
<b>Järjestelmätestaus</b>						xx	xx
<b>Hyväksymistestaus</b>							xx

# 9 Testausraportti

## 9.1 Yleistä

Tässä testausdokumentin osassa kuvataan Ohjelmistotuotantoprojekti-kurssilla suoritettun ohjelmistotuotantoprojekti eMon tuotteen testausta. Dokumentissa raportoidaan testauksen tuloksista sekä verrataan testausta suunnitelmassa määriteltyihin tavoitteisiin. Suoritetuista testeistä on pyritty kirjoittamaan pienet raportit sekä listattu testeissä ilmenneet virheet. Myöskin vinkkejä näiden virheiden sietämiseen annetaan.

Testausta suoritettiin sekä toteutuksen kanssa rinnakkain, että toteutuksen jälkeen. Kuitenkin testauksessa tuli kiire, vaikka sille olikin näennäisesti varattu ruhtinaallisesti aikaa. Suunnittelun ja toteutuksen venyminen söivät aikaa itse testaukselta.

## 9.2 Testiympäristö

Testaus on suoritettu eMo-ryhmän omalla Moodlella. Sitä on ajettu palvelimella db.cs.helsinki.fi. Käytössä on ollut vaatimusmäärittelydokumentissa määritellyt ympäristöt.

## 9.3 Yksikkötestaus ja integrointitestaus

Järjestelmää toteutettaessa huomattiin, että monet toteutettavaksi ajatellut toiminnot ja komponentit olivat jo tarjottuja Moodle-ympäristön puolesta. Näiden komponenttien toimintaa ei katsottu projektiryhmän testattavaksi. Yksikkötestaus ja integrointitestaus painottuivatkin erityisesti näiden komponenttien käytön testaamiseen tuotteessa sekä erityisesti lisäys ja poisto-operaatioissa algoritmien toiminnan testaamiseen.

Testauksen kattavuus huomattiin vaikeasti todettavaksi, sillä mitään järkevää mittausta tälle ei ympäristö huomioon ottaen löydetty. Testaussuunnitelmassa esitettyyn kattavuustavoitteeseen pääsyä ei voida todentaa, mutta käytännössä tätä tavoitetta tuskin saavutettiin. Yksikkötestaus ja integrointitestaus suoritettiin koodauksen lomassa improvisoiden, luokan tai toiminnan koodaaja testasi itse omat koodinsa ja totesi sen toimivan, tai jos se ei toiminut niin tämän jälkeen korjasi työnsä ja testasi sen uudelleen.

Tässä luvussa järjestelmä on jaettu ”osajärjestelmiin”, pienempiin kokonaisuuksiin, joiden voidaan

katsoa liittyvän toisiinsa. Yhdistettynä nämä kokonaisuudet muodostavat yhdessä toimivan kokonaisuuden – eMo järjestelmän.

### 9.3.1 Tehtävätyyppi (OKREGEXP)

Koodi:Okregexp 1  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävään vastaus  
Miten testattiin: Annetaan palautettavaan vastauskenttään tyhjä merkkijono ja palautetaan se  
Odotetut tulokset: Virhe ilmoitus vastauksen väärästä syntaksista  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi:Okregexp 2  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävään vastaus  
Miten testattiin: Annetaan palautettavaan vastauskenttään -1 ja palautetaan se  
Odotetut tulokset: Virhe ilmoitus vastauksen väärästä syntaksista  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi:Okregexp 3  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävään vastaus  
Miten testattiin: Annetaan palautettavaan vastauskenttään /ooo ja palautetaan se  
Odotetut tulokset: Virhe ilmoitus vastauksen väärästä syntaksista  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi:Okregexp 4  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävään vastaus  
Miten testattiin: Annetaan palautettavaan vastauskenttään ooo/ ja palautetaan se  
Odotetut tulokset: Virhe ilmoitus vastauksen väärästä syntaksista  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi:Okregexp 5  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävään vastaus  
Miten testattiin: Annetaan palautettavaan vastauskenttään /// ja palautetaan se  
Odotetut tulokset: Virhe ilmoitus vastauksen väärästä syntaksista  
Tulos: Ei-odotettu tulos  
Havaitut virheet: Warning: preg\_match() [function.preg-match]: Unknown modifier '/'  
Testin tulos: Testissä löytyi virhe.  
Virheenaiheuttaja: Palautettavassa merkkijonossa ei saa olla kauttaviivaa, jollei se ole ensimmäinen, viimeinen tai toiseksi viimeinen merkki, jolloin viimeisen merkin tulee olla i.  
Virhe havaittiin projektin viimeisessä vaiheessa eikä sitä päätetty korjata vaan se on dokumentoitu.

Koodi:Okregexp 6  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävän tietojen tallettaminen tietokantaan  
Miten testattiin: Luotiin uusi kysymys arvoilla:  
Question name Tietokantakokeilu  
Question <span lang="fi">Kysymys suomeksi</span>  
<span lang="en">Question in english</span>  
Feedback: <span lang="fi">Palaute</span>  
<span lang="en">Feedback</span>  
Correct clauses Ensimmäinen oikea  
Toinen oikea  
Kolmas oikea  
Neljäs oikea  
Viides oikea  
Wrong clauses Ensimmäinen väärä  
Toinen väärä  
Kolmas väärä  
Neljäs väärä  
Viides väärä

Odotetut tulokset: Tiedot tallentuvat tietokantaan  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi:Okregexp 7  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävän lisäys Tenttiin (Quiz)  
Miten testattiin: Luotiin uusi tentti nimeltä Testi-tentti ja lisättiin testissä Okregexp 6 määritelty kysymys siihen.  
Odotetut tulokset: Uusi tentti, jonka ainoana kysymyksenä on testissä Okregexp 6 määritelty kysymys.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi:Okregexp 8  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävän esitys opiskelijalle eli tietojen hakeminen tietokannasta  
Miten testattiin: Käytettiin testissä Okregexp 6 määriteltyä kysymystä ja testissä Okregexp 7 määriteltyä tenttiä, johon opiskelija pyrkii vastaamaan  
Odotetut tulokset: Opiskelijalle näytetään testissä Okregexp 6 määritellyn kysymyksen tiedot display.html:ssä määritellyllä tavalla.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi:Okregexp 9  
Testaaja: Ossi  
Testin kohde: Ok Regular Expression-tehtävätyypin tehtävään vastaus tentissä.  
Miten testattiin: Käytettiin testissä Okregexp 7 saatua lopputulosta. Annettiin palauttettavaan vastauskenttään merkkijono /car/ ja palautettiin se  
Odotetut tulokset: Opiskelijalle ilmoitetaan hänen antamansa vastauksen perusteella tunnistaako lause oikeita tai vääriä lauseita ja sen perusteella opiskelija saa arvosanan  
Tulos: Odotettu tulos  
Havaitut virheet: --

Testin tulos: Testi on onnistunut.

Koodi:Okregexp 10

Testaaja: Ossi

Testin kohde: Ok Regular Expression-tehtävätyypin tehtävän poistaminen kun tehtävä kuuluu johonkin tenttiin

Miten testattiin: Käytettiin testissä Okregexp 7 saatua lopputulosta ja pyrittiin poistamaan kysymys.

Odotetut tulokset: Varoitus siitä että kysymys kuuluu tenttiin.

Tulos: Odotettu tulos

Havaitut virheet: --

Testin tulos: Testi on onnistunut.

Koodi:Okregexp 11

Testaaja: Ossi

Testin kohde: Ok Regular Expression-tehtävätyypin tehtävän poistaminen kun tehtävä ei kuulu tenttiin

Miten testattiin: Käytettiin testissä Okregexp 7 saatua lopputulosta ja pyrittiin poistamaan kysymys.

Odotetut tulokset: Varmistus kysymyksen poistosta ja vastaamalla Kyllä (Yes) tehtävä poistuu (myös tietokannasta)

Tulos: Odotettu tulos

Havaitut virheet: --

Testin tulos: Testi on onnistunut.

Koodi:Okregexp 12

Testaaja: Ossi

Testin kohde: Ok Regular Expression-tehtävätyypin tehtävän muokkaaminen

Miten testattiin: Käytettiin testissä Okregexp 6 luotua tehtävää ja muokattiin sen tietoja.

Odotetut tulokset: Kysymyksen tiedot muuttuvat ja päivittyvät tietokantaan.

Tulos: Odotettu tulos

Havaitut virheet: --

Testin tulos: Testi on onnistunut.

### 9.3.2 Tehtävätyypin lisääminen

Koodi: LIS1

Testaaja: Juho

Testin kohde: chmod\_recursive() -funktio

Miten testattiin: Ohjelma ajettiin testihakemistorakenteeseen, joka sisälsi hakemistoja ja tiedostoja täysillä oikeuksilla(777), kutsumalla seuraavasti: chmod\_recursive('755', '644', 'moditesti', 1);

Odotetut tulokset: Annettu hakemisto (moditesti) ja sen sisältämät hakemistot saavat chmod 755 (drwxr-xr-x) -oikeudet. Hakemiston sisältämät tiedostot saavat chmod 644 (-rw-r--r--) -oikeudet. Funktio palauttaa numeron 1, mikäli funktion kutsussa annettu hakemistonimi ei olekaan hakemisto. Muuten funktio ei palauta arvoa.

Tulos: Odotettu tulos.

Havaitut virheet: --

Testin tulos: Testi on onnistunut

Muuta: Testiä ei ole tarpeen suorittaa muilla oikeuksilla, koska tämä on ainoa eMo-järjestelmässä vastaantuleva tapaus. Funktio käyttää linuxin chmod-komentoa, joten se ei mitään ilmeisimmin toimi muissa käyttöjärjestelmissä

Koodi: LIS2

Testaaja: Ville-Pekka

Testin kohde: Sivun pohjan toiminta

Miten testattiin: Katsottiin, näkyykö sivu web-selaimessa

Odotetut tulokset: Sivun näkyminen web-selaimessa sellaisenaan, eikä esim mitään



debug-ilmoituksia tai virheilmoituksia näy.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: LIS3  
Testaaja: Ville-Pekka  
Testin kohde: Step 1  
Miten testattiin: Yritettiin ladata pakettia antamatta mitään tietoja, tai antamalla sopimattomia tietoja, esim. tyyppin ID:ksi merkkejä, jotka eivät sovi väleille [a-z][0-9]  
Odotetut tulokset: Näytetään virheilmoitus, jossa kerrotaan mitä on muutettava, jotta template-paketin voi ladata.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: LIS4  
Testaaja: Ville-Pekka  
Testin kohde: Step 1  
Miten testattiin: Annettiin kunnolliset tiedot ja katsottiin, saatiinko niitä vastaava paketti  
Odotetut tulokset: Saadaan latauksena zip-paketti, joka sisältää templatien, jossa on lomakkeessa annetut tiedot ja kaikki tarvittavat tiedostot.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: LIS5  
Testaaja: Ville-Pekka  
Testin kohde: Step 2  
Miten testattiin: Annettiin useita virheellisiä zip-tiedostoja, joista puuttui joko valinnaisia tai pakollisia tiedostoja  
Odotetut tulokset: Virheilmoitus. Jos vain valinnaisia tiedostoja puuttuu, tehtävätyyppi lisätään. Jos pakollisia tiedostoja puuttuu, tehtävätyyppejä ei lisätä.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: LIS6  
Testaaja: Ville-Pekka  
Testin kohde: Step 2  
Miten testattiin: Annettiin zip-tiedosto, jonka sisältämä tehtävätyyppi on jo järjestelmässä  
Odotetut tulokset: Virheilmoitus, tehtävätyyppejä ei lisätä.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: LIS7  
Testaaja: Juho  
Testin kohde: Step 2  
Miten testattiin: Annettiin zip-tiedosto, jonka sisältämä tehtävätyyppi sisältää oleelliset tiedostot ja lang- ja help-tiedostot suomeksi ja englanniksi.  
Odotetut tulokset: Tyyppi asennetaan, lang- ja help-tiedostot kopioituvat oikeille paikoilleen ja kaikkien tiedostojen käyttäjäoikeudet asetetaan oikein. Tehtävätyyppejä kokeiltaessa suomeksi ja englanniksi, kielet toimivat.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: LIS8  
Testaaja: Teemu  
Testin kohde: Lang-tiedostot väärään paikkaan  
Miten testattiin: Lisättiin tehtävätyyppi, jonka pakettiin määriteltiin lang-tiedostoja jotka yritettiin lisätä väärään paikkaan hakemistopuussa (ts. lang-tiedostot eivät sijaitse lang/ -kansiossa).  
Odotetut tulokset: Tehtävätyyppi lisääntyy, mutta lang-tiedostot eivät, sillä langtiedostot menevät väärään paikkaan. Tätä hakemistoa ei luoda.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut

### 9.3.3 Tehtävätyypin poistaminen

#### 9.3.3.1 *showremovetype.php*

Showremovetype.php huolehtii poistokäyttöliittymän toteuttamisesta. Se piirtää listan poistettavista ja ei-poistettavista tehtävätyypeistä sekä tarjoaa mahdollisuuden poistaa näitä.

showremovetype.php:n testit

Koodi: SR1  
Testaaja: Teemu  
Testin kohde: Sivun pohjan toiminta  
Miten testattiin: Katsottiin, näkyykö sivu web-selaimessa  
Odotetut tulokset: Sivun näkyy web-selaimessa sellaisenaan, eikä esim mitään debug-ilmoituksia tai virheilmoituksia näy.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: SR2  
Testaaja: Teemu  
Testin kohde: Listan generointi 1  
Miten testattiin: Haettiin listalle tietoja, normaali määrä tietueita (muutama)  
Odotetut tulokset: Listaa tietueet checkbox-listaan  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: SR3  
Testaaja: Teemu  
Testin kohde: Listan generointi 2  
Miten testattiin: Haettiin listalle tietoja, ei yhtään tietuetta haettavissa  
Odotetut tulokset: Ei listaa mitään.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: SR4  
Testaaja: Teemu  
Testin kohde: Listan generointi 3  
Miten testattiin: Hakemistonimessä skandeja  
Odotetut tulokset: Tulostaa hakemistot ok  
Tulos: skandimerkit tulostuu huonosti  
Havaitut virheet: Skandimerkit ei tulostu kunnolla  
Testin tulos: Testi ei onnistunut.  
Virheen sietäminen: Tehtävätyypin lisäämiseen määritellään, ettei skandimerkkejä

saa syöttää.

Koodi: SR5

Testaaja: Teemu

Testin kohde: Tietokantatesti

Miten testattiin:

Odotetut tulokset: Poistolistaan tulee ainoastaan valideja poistettavia, mistään ei ole kysymyksiä käytössä. Sellaiset tyytit, joista on kysymyksiä, tulostetaan mutta niitä ei voida poistaa

Tulos: Odotettu tulos

Havaitut virheet: --

Testin tulos: Onnistunut

Koodi: SR6

Testaaja: Teemu

Testin kohde: Listan generointi 3

Miten testattiin: Haettiin listalle tietoja, todella monia tietueita, kaikkien tehtävätyyppien pitää tulla listaan.

Odotetut tulokset: Listaa kaikki tiedot listaan, kaikki tehtävätyypit

Tulos: Odotettu tulos

Havaitut virheet: --

Testin tulos: Testi on onnistunut.

Koodi: SR7

Testaaja: Teemu

Testin kohde: Reset-buttonin toiminta

Miten testattiin: Klikkailtiin eri määrä checkboxeja aktiiviseksi, (0, muutama, kaikki), tämän jälkeen painettiin "Reset"-nappia

Odotetut tulokset: Poista ruksit listasta, jos ei ole yhtäkään valittu: älä tee mitään

Tulos: Odotettu tulos, kaikissa tapauksissa

Havaitut virheet: --

Testin tulos: Testi on onnistunut.

Koodi: SR8

Testaaja: Teemu

Testin kohde: Submit-buttonin toiminta

Miten testattiin: Klikkailtiin eri määrä checkboxeja aktiiviseksi, (0, muutama, kaikki), tämän jälkeen painettiin "Submit"-nappia

Odotetut tulokset: Käynnistä toiminnallisuus, välitä lomakkeen tiedot käsittelijälle joka hoitaa jatkotoimenpiteet

Tulos: Odotettu tulos

Havaitut virheet: --

Testin tulos: Testi on onnistunut.

Koodi: SR9

Testaaja: Teemu

Testin kohde: Kieliriippumattomuus

Miten testattiin: Vaihdettiin kieltä, ensin englanninkielinen: tekstit näkyivät englanniksi. Sitten suomenkielinen, tekstit suomeksi.

Odotetut tulokset: Kieli vaihtuu, haetaan tekstit eri lang-filuista.

Tulos: odotettu tulos.

Havaitut virheet: --

Testin tulos: onnistunut

Koodi: SR10

Testaaja: Teemu

Testin kohde: Avustustoiminnot

Miten testattiin: Napauteltiin sivujen '?' ikoneita ja katsottiin mitä tapahtuu.

Odotetut tulokset: Avustusikkuna näytetään, ja jos siihen on koodattu käytössä olevalla kielellä avustustoimintoja, ne näytetään.

Tulos: odotettu tulos.  
Havaitut virheet: --  
Testin tulos: onnistunut

Koodi: SR11  
Testaaja: Teemu  
Testin kohde: Listan muodostaminen  
Miten testattiin: ladattiin sivu. Tulostaako pelkästään tyyppit, joista EI OLE kysymyksiä, testihetkellä ainoa eMo-tyyppi josta ei ole kysymyksiä on tyyppi 'hapokasta'.  
Odotetut tulokset: ainoastaan 'hapokasta' tulostuu checkboxilla varustettuna, muista tehtävätyypeistä tulostetaan tunnukset ja ilmoitus ettei näitä voida poistaa.  
Tulos: odotettu tulos  
Havaitut virheet: --  
Testin tulos: Onnistunut

### **9.3.3.2 processremovetype.php**

Processremovetype.php toteuttaa poiston toiminnallisen osan. Se saa showremovetype.php:lta poistettavaksi tarkoitetut tyyppit ja poistaa ne sekä niihin liittyvät lang- sekä help-tiedostot

processremovetype.php:n testit:

Koodi: PR1  
Testaaja: Teemu  
Testin kohde: Poisto 1  
Miten testattiin: Asetetaan poistettavaksi 0 kpl tyyppejä  
Odotetut tulokset: Älä poista mitään, palaa poistosivulle, kaikki näkyvät edelleen  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: PR2  
Testaaja: Teemu  
Testin kohde: Poisto 2  
Miten testattiin: Asetetaan poistettavaksi kaikki listan tyyppit (kaikissa sisältöä)  
Odotetut tulokset: Poista kaikki, (poista tietokantaviitteet) , palaa poistosivulle, ei poistetut tyyppit näkyvät edelleen listassa.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: PR3  
Testaaja: Teemu  
Testin kohde: Poisto 3  
Miten testattiin: Asetetaan poistettavaksi muutamia tyyppejä  
Odotetut tulokset: Poista kyseiset tyyppit, palaa poistosivulle, ei poistetut näkyvät edelleen.  
Tulos: Odotettu tulos  
Havaitut virheet: --  
Testin tulos: Testi on onnistunut.

Koodi: PR4  
Testaaja: Teemu  
Testin kohde: Poisto 4  
Miten testattiin: Poistetaan tyyppi jonka id-nimessä on skandeja  
Odotetut tulokset: Poista kyseiset tyypit, palaa poistosivulle  
Tulos: ei poista mitään  
Havaitut virheet: Ei poista  
Testin tulos: Testi ei onnistunut.  
Virheen sietäminen: Skandit eivät sallittuja tyypissä, ei voida lisätä skandimerkkejä sisältäviä kysymyksen id-nimiä.

Koodi: PR5  
Testaaja: Teemu  
Testin kohde: lang-tiedostojen poisto  
Miten testattiin: poistettiin tyyppettä ja tarkastettiin poistuuko niiden lang-tiedostot myöskin, jos ei lang tiedostoja niin se ei vaikuta poistoon: tiedostot vaan jäävät poistamatta  
Odotettu tulos: Lang tiedostot poistuvat jos niitä on, muuten normi poisto  
Tulos: Lang-tiedostot poistuivat jos niitä oli  
Havaitut virheet: --  
Testin tulos: Onnistunut  
Huomio: Jos kysymystyyppillä ei ole lang-tiedostoja, menetellään kuin aikaisemmissa Poisto-testeissä.

Koodi: PR6  
Testaaja: Teemu  
Testin kohde: Helppitiedostojen poisto  
Miten testattiin: Kuten yllä, mutta lang- korvataan helpillä  
Odotetut tulokset: Kuten yllä  
Tulos: odotettu tulos  
Havaitut virheet: --  
Testin tulos: Onnistunut

Huomio: Jos kysymystyyppillä ei ole help-tiedostoja, menetellään kuin aikaisemmissa Poisto-testeissä.

### **9.3.4 Käyttöoikeudet**

Tässä luvussa testataan, että ainoastaan 'laatija' (eMo creator) on kykeneväinen lisäämään ja poistamaan eMo-tehtävätyyppejä.

**Testaaja ja aika:** Teemu ja Ossi 12.12.2006 13:44

#### **9.3.4.1 Laatija**

**Tehtävätyypin lisääminen:** Katso UCT5 testitulokset

**Tehtävätyypin poistaminen:** Katso UCT6 testitulokset.

**Testin tulos:** onnistunut

### **9.3.4.2 Opettaja**

**Tehtävätyypin lisääminen ja poistaminen:** Opettaja-käyttäjryhmällä ei ole oikeuksia mennä eMo-kysymystyyppin poisto ja lisäysvalikoihin. Yritys mennä sorkkimaan näitä päättyy virheilmoitukseen, jossa kerrotaan ettei ole oikeuksia mennä tälle alueelle.

**Testin tulos:** onnistunut

### **9.3.4.3 Opiskelija**

**Tehtävätyypin lisääminen ja poistaminen:** Opiskelija-käyttäjryhmällä ei ole oikeuksia mennä eMo-kysymystyyppin poisto ja lisäysvalikoihin. Yritys mennä sorkkimaan näitä päättyy virheilmoitukseen, jossa kerrotaan ettei ole oikeuksia mennä tälle alueelle.

**Testin tulos:** onnistunut

## **9.4 Järjestelmätestaus**

Tässä luvussa on testattu tuotetta sen käyttöliittymää käyttäen. Testit ovat perustuneet käyttötapuksiin ja niiden pohjana on käytetty jo mainittuja EUCT-käyttötapauksia, jotka on toimitettu liitteenä. Järjestelmätestauksessa ei ole enää puututtu teknisiin asioihin, vaan järjestelmää on testattu samalla tavalla kuin järjestelmän käyttäjä voisi sitä käyttää.

### **9.4.1 Opiskelijan käyttötapaukset**

#### **9.4.1.1 UCS1: Tehtäviin tutustuminen**

**Yleiskuvaus:** Opiskelija katselee järjestelmään tallennettuja tehtäviä. Tehtävien täytyy olla kysymyssarjassa jotta ne näkyvät.

**Tapahtumien normaali kulku:** Opiskelija valitsee tehtävälisteristä tehtävän, jonka tiedot (kysymyksen ja vastausmahdollisuudet) järjestelmä näyttää ruudulla. Arvotut muuttujat arvottaa tässä vaiheessa.

**Odotettu tulos:** Tehtävä(t) näkyy näytetään ja siihen on mahdollista vastata.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta: --**

**Testin tulos:** Testi on onnistunut

**Testaaja ja aika:** Teemu ja Ossi 12.12.2006 12:08

#### **9.4.1.2 UCS2: Tehtävään vastaaminen**

**Yleiskuvaus:** Opiskelija vastaa tehtävään ja saa siitä arvostelun. Käytetään okregexp-tehtävää.

**Tapahtumien normaali kulku:** Opiskelija valitsee tehtäviin vastaamisen toiminnon. Järjestelmä näyttää yhden tehtävän kerrallaan, antaa mahdollisesti pistemäärän opiskelijalle ja kertoo, oliko vastaus oikein vai väärin. Opiskelijan tekemät tehtävät ja niistä saadut pisteet tallennetaan. Mikäli kyseessä on ollut tehtävä, johon kuuluu arvottavia osia, arvotut osat tallennetaan myös, jotta mahdollisella uudella yrityskerralla opiskelijalle näytetään täysin sama versio kysymyksestä.

**Odotettu tulos:** Opiskelijan vastaus käsitellään ja opiskelija saa siitä palautteen. Vastaukset tallennetaan.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta:** Valittaa säännöllisen lausekkeen syntaksista virheilmoituksella, jos syntaksi väärin, mutta käsittelee vastauksen.

**Testin tulos:** Testi on onnistunut.

**Testaaja ja aika:** Teemu ja Ossi 12.12.2006 12:10

#### **9.4.2 Opettajan käyttötapaukset**

Opettajaksi kutsutaan henkilöä, joka pitää kurssia Moodlessa ja haluaa tehdä kurssilleen uusia tehtäviä sekä muokata jo olemassa olevia versioita.

##### **9.4.2.1 UCT1: Tehtävän lisääminen**

**Yleiskuvaus:** Opettaja lisää tehtävän järjestelmään.

**Tapahtumien normaali kulku:** Opettaja käynnistää tehtävän lisäämisen. Hän määrittelee tehtävänannon, oikean vastauksen, opiskelijalle annettavan pistemäärän ja palautteen ja tallentaa tehtävän järjestelmään. Jos jokin määriteltävistä kohdista jää määrittelemättä, annetaan virheilmoitus.

**Odotettu tulos:** Tehtävä näkyy questions-valikossa ja sitä voidaan katsoa ja täyttää. Jos virhetilanne, niin annetaan virheilmoitus.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta: --**

**Testin tulos:** Testi on onnistunut.

**Testaaja ja aika:** Teemu ja Ossi, 12.12.2006 11:50

#### **9.4.2.2 UCT2: Tehtävän muokkaaminen**

**Yleiskuvaus:** Opettaja muokkaa olemassa olevaa tehtävää.

**Tapahtumien normaali kulku:** Opettaja valitsee muokattavan tehtävän ja muokkaa jotain tehtävän osaa, kuten tehtävänantoa, oikeaa vastausta, pistemäärää tai palautetta. Jos joku on jo vastannut tehtävään, opettajalta kysytään poistetaanko olemassa olevat vastaukset vai jätetäänkö ne talteen. Näin siksi, että jos tehtävästä on korjattu vain vähäinen kirjoitusvirhe, se ei vaikuta annettuihin arvosanoihin. Lopuksi opettaja tallentaa muokatun tehtävän järjestelmään.

**Odotettu tulos:** Jos on vastauksia, kysytään että tallennetaanko vastaukset vai ei? Tämän jälkeen tallennetaan muutokset. Jos ei ole vastauksia, tallennetaan muutokset suoraan.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta:** Ei esitä kysymyksiä, vaikka joku olisikin jo vastannut kysymykseen. Muuten muokkaaminen ok.

**Testin tulos:** Onnistunut poikkeuksin.

**Testaaja ja aika:** Teemu ja Ossi 12.12.2006 12:20

#### **9.4.2.3 UCT3: Tehtävän poistaminen**

**Yleiskuvaus:** Opettaja poistaa olemassa olevan tehtävän järjestelmästä.

**Tapahtumien normaali kulku:** Opettaja valitsee poistettavan tehtävän ja, napauttaa poistonappia ja hyväksyy poiston. Jos joku on jo vastannut tehtävään, opettajaa varoitetaan siitä että jos tehtävän poistaa niin myös suoritukset poistetaan. Mikäli opettaja hyväksyy tämän, tehtävä poistetaan järjestelmästä. Muuten palataan edelliseen tilanteeseen.

**Odotettu tulos:** Jos on vastauksia, varoitusilmoitus ja varmistuskysymys. Muutoin varmistuskysymys. Jos opettaja varmistaa poiston, poistetaan tehtävä.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta:** Jos kysymyksestä on



vastauksia, kysymys voidaan poistaa kysymyslistasta, mutta kysymys jää vielä quiziin. Muutoin poisto ok.

**Testin tulos:** Onnistunut poikkeuksin.

**Testaaja ja aika:** Ossi ja Teemu 12.12.2006 12:30

#### **9.4.2.4 UCT4: Tehtävien selailu**

**Yleiskuvaus:** Opettaja katselee järjestelmään tallennettuja tehtäviä.

**Tapahtumien normaali kulku:** Opettaja valitsee tehtävällystasta tehtävän, jonka tiedot järjestelmä näyttää ruudulla.

**Odotettu tulos:** Valitun tehtävät tiedot näkyvät ruudulla.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta: --**

**Testin tulos:** Testi on onnistunut

**Testaaja ja aika:** Teemu 12.12.2006 1:40

### **9.4.3 Laatijan käyttötapaukset**

Laatija on henkilö, joka laatii järjestelmään uusia tehtävätyyppejä PHP-kielillä. Laatija voi olla opettaja tai muu toimihenkilö. Laatijalla on myös oikeudet poistaa eMo-tehtävätyyppejä järjestelmästä.

#### **9.4.3.1 UCT5: Uuden tehtävätyypin lisääminen**

**Yleiskuvaus:** Laatija lisää järjestelmään uuden tehtävätyypin, joka on ohjelmoitu PHP-kielillä eMo-järjestelmän asettamien vaatimusten mukaisesti.

**Toiminnan kulku:** Täytetään lomake, imuroidaan koodipaketti, ohjelmoidaan tehtävätyyppi, lähetetään toteutettu tehtävätyyppi moodleen. Tämän jälkeen tehtävätyyppi on lisätty. Tässä testissä käytetään projektissa toteutettua okregexp-tehtävätyyppeä.

**Odotettu tulos:** Tehtävätyyppi näkyy tehtävätyyppi-listassa.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta: --**

**Testin tulos:** Testi on onnistunut.

**Testaaja ja aika:** Teemu, 12.12.2006 0:52

#### **9.4.3.2 UCT6: Tehtävätyypin muokkaaminen**

**Yleiskuvaus:** Laatija muokkaa järjestelmässä olevaa tehtävätyyppejä.

**Toiminnan kulku:** PHP-koodia ja tietokannan määrittelyä voi muuttaa vain, jos tällaisia tehtäviä ei ole, jos halutaan muuttaa tehtävätyyppejä josta on jo kysymyksiä, on kysymykset ensin poistettava. Laatija valitsee muokattavan tehtävätyypin, tekee muutokset ja tallentaa ne php-koodiinsa. Vain tehtävätyypin nimeä ja vastaavia kenttiä voi muuttaa, jos tehtävätyypin perusteella on tehty olemassa olevia tehtäviä.

**Odotettu tulos:** Jos on tehtäviä, vain tehtävätyypin nimeä ja vastaavia kenttiä voidaan muuttaa. Jos tehtäviä ei ole, voidaan koko tehtävätyyppejä muuttaa ohjelmoimalla php-koodia uudelleen.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta: --**

**Testin tulos:** Testi on onnistunut.

**Testaaja ja aika:** Teemu, 12.12.2006 1:00

#### **9.4.3.3 UCT7: Tehtävätyypin poistaminen**

**Yleiskuvaus:** Laatija poistaa järjestelmässä olevan tehtävätyypin.

**Tapahtumien normaali kulku:** Laatija valitsee poistettavan tehtävätyypin. Tehtävätyyppejä poistetaan järjestelmästä. Ainoastaan sellaisia tehtävätyyppejä voidaan poistaa, mitkä eivät ole käytössä millään kurssilla ts. sellaiset tehtävätyypit voidaan poistaa, joista ei ole kysymyksiä.

**Odotettu tulos:** Jos tehtävätyypistä ei ole tehtäviä, tarjotaan se poistettavaksi poistomenussa ja se poistuu tehtävätyyppejä valitsemalla ja napauttamalla nappulaa. Jos tehtävätyypistä on olemassa tehtäviä, se tulostuu siten, ettei sitä voida yrittääkään poistaa. Valitut tehtävätyypit poistuvat kun napautetaan nappia.

**Havaitut virheet, puutteet tai poikkeamat odotetusta toiminnasta: --**

**Testin tulos:** Testi on onnistunut.

**Testaaja ja aika:** Teemu, 12.12.2006 1:06

## **9.5 Muita järjestelmätestejä**

Tässä osiossa käydään läpi testausraportteja sellaisista järjestelmätesteistä, joiden käyttötapauksia ei oltu määritelty vaatimusmäärittelydokumentissa, vaan ne keksittiin järjestelmätestausvaiheessa.

Koodi: JÄRJ1

Mitä testataan? Tehtävätyypin lisäys, lang-tiedostojen lisäys, help-tiedostojen lisäys.

Skenaario: Laatija kirjautuu järjestelmään ja lisää tehtävätyypin, jossa on kahdenkielisiä lang-tiedostoja (englanninkielinen ja suomenkielinen) sekä englanninkielinen help-tiedosto. Laatija imuroi koodipohjapaketin ja koodaa tehtävätyypin sekä lisää em. tiedostot ja lähettää tehtävätyypinsä järjestelmään.

Odotettu tulos: Tehtävätyyppi näkyy Moodlessa, lang-tiedostot ovat menneet omiin kansioihinsa, samoin help-tiedosto.

Havaitut virheet: --

Testin tulos: Onnistunut

Testaaja ja aika: Teemu 14.12.2006 14:52

Koodi: JÄRJ2

Mitä testataan? Tehtävätyypin poisto, lang-tiedostojen poisto, help-tiedostojen poisto.

Skenaario: Jatkoa edelliseseen tehtävään. Edellisessä testissä luotu tehtävätyyppi on nyt saavuttanut elinkaarensa pään eikä siitä ole enää mitään tehtäviä. Se pitää poistaa. Laatija poistaa sen järjestelmästä valitsemalla tehtävätyypin poistolista ja napauttamalla "Remove" nappia poistaa sen. Samalla poistuvat myös kielimäärittelyt sekä helpimäärittelyt hakemistoistaan.

Odotettu tulos: Tehtävätyyppi poistuu, lang-tiedostot poistuvat, help-tiedosto poistuu.

Havaitut virheet: --

Testin tulos: Onnistunut

Testaaja ja aika: Teemu 14.12.2006 14:58

Koodi: JÄRJ3

Mitä testataan? Monikielisyys

Skenaario: Vaihdetaan kieli Suomenkieliseksi ja suoritetaan testit JÄRJ1 sekä JÄRJ2

Odotettu tulos: Muuten sama kuin testeissä JÄRJ1, JÄRJ2, mutta kieli on englannin sijaan selvää suomea.

Havaitut virheet: Kielioppivirhe poistolistan tulostamisessa

Toimenpiteet: Kielioppivirheen korjaaminen

Testin tulos: Onnistunut korjauksin

Testaaja ja aika: Teemu 14.12.2006 15:10

Koodi: JÄRJ4

Mitä testataan: Asennus ja käyttöönotto

Skenaario: Asennettiin uusi moodle johon tämän jälkeen asennettiin eMo sekä laatija-käyttöoikeudet

Odotettu tulos: Moodle toimii ja eMo toimii

Havaitut virheet: qtype\_emo.php -tiedosto ei ollu paikallaan, jolloin kysymystyyppi-listassa luki [[eMo]] pelkän 'eMo' tekstin sijaan. Muutoin odotettu toiminta.

Toimenpiteet: Korjattiin

Testin tulos: Onnistunut korjauksin

Testaaja ja aika: Ville-Pekka 15.12.2006 15:42

## 9.6 Vaatimusten toteutuminen

Tässä luvussa tarkastetaan, ovatko asiakkaan ja projektiryhmän vaatimusmäärittelyssä määrittelemät vaatimukset järjestelmälle toteutuneet. Vaatimusten toteutuminen on huomattu testauksen yhteydessä. Alla olevista taulukoista löytyvistä tiedoista käy ilmi, mitä osa-alueita testaamalla mikäkin vaatimus on saatu määriteltyä toteutetuksi sekä testatuksi.

### 9.6.1 Järjestelmävaatimukset

<i>Id</i>	<i>Vaatus</i>	<i>Validointitapa</i>	<i>Tulos</i>
J1	pisteytys	UCS2, UCT4	OK
J2	arvotut tehtävän muuttajat	Suunnitteludokumentti	OK (ei toteutusta)
J3	palautteen antaminen	UCS2, UCT4	OK
J4	tehtävien automaattinen tarkistus	UCS2, UCT4	OK
J5	Javascript-tuki	Koodirungot	OK
J6	avustustoiminnot	Yksikkötestaus	OK
J7	kieliriippumattomuus	Yksikkötestaus	OK
J8	yksi tietokantarakenne	UCT5, UCT6	OK
J9	geneerinen tehtävätyyppi	UCT5, UCT6, koodipohjat	OK
J10	tehtävätyyppikohtainen tyylimäärittely	UCT5, UCT6	OK
J11	vastauksen arvostelun tallennus	UCS2, UCS4	OK
J12	arvottavien tehtävämuuttujien tallennus	Suunnitteludokumentti	OK
J13	tehtävän näyttäminen	UCS1	OK

## 9.6.2 Käyttäjävaatimukset

<i>Id</i>	<i>Vaatus</i>	<i>Validointitapa</i>	<i>Tulos</i>
K1	uuden tehtävätyypin lisäys	UCT5, okregexp-esimerkki, koodipohjat	OK
K2	uuden tehtävätyypin luominen	UCT5, okregexp-esimerkki, koodipohjat	OK
K3	selkeä käyttöliittymä lisäykselle	UCT5	OK (makuasioita)
K4	hyvin kommentoitu koodi	Ohjelmakoodit	OK
K5	selkeät rajapinnat	Moodlen rajapinnat ja funktiot käytössä	OK
K6	eri käyttäjäryhmät	Yksikkö ja integrointitestaus, laatija-käyttäjäryhmä, laatijatestaus	OK
K7	tehtävätyypin erikoistaminen	UCT5, okregexp-esimerkki	OK
K8	käyttöohje	Käyttöohjedokumentti	OK
K9	tehtävään vastaaminen	UCS2	OK
K10	omien tulosten selaaminen	UCS1	OK

## 9.7 Hyväksymistestaus

Tuote luovutetaan asiakkaalle 15.12.2006, asiakas suorittaa hyväksymistestauksen tämän jälkeen. Asiakas on kertonut kiinnittävänsä erityistä huomiota dokumentaatioon ja tehtävätyypin lisäämiseen sekä sen helppouteen ja ohjeistukseen.

## 9.8 Testauksen yhteenveto

Testaus sujui muuten aikataulussa, mutta projektin loppupuolelle mentäessä aika alkoi loppua kesken, ja järjestelmätestauksessa tuli hieman kiire. Töitä jouduttiin tekemään yötä myöten. Muuten testaus sujui jouhevasti. Suurimpia ongelmia aiheutti eksaktin testausympäristön puuttuminen. Moodlesta löydettyihin testaustyökaluihin ei kellään ollut halua perehtyä ja ne vaikuttivat muutenkin epäselviltä ja varsin heikosti ohjeistetuilta, myöskin toteutuksen aikataulun kireys vaikutti osaltaan työkaluihin tutustumiseen – toteutuksessa oli usein niin kiire, ettei tällaista tutustumista yksinkertaisesti ehtinyt suorittaa siinä mittakaavassa, että työkaluihin olisi syventynyt niin paljon että niistä olisi ollut jotain erityistä hyötyä.

Testauksen perusteella on syytä olettaa järjestelmän vastaavan sille määriteltyjä vaatimuksia sekä sen toimivan virheettömästi ja jouhevasti. Mitään dramaattisia virheitä, jotka olisivat johtaneet

ohjelmiston suorituksen loppumiseen, ei löydetty. Testauksen perusteella voidaan myös todeta, että järjestelmä selviytyy sille vaatimusmäärittelydokumentissa määritellyistä käyttötapauksista. Ohjelmisto luovutetaan asiakkaalle projektin päätöspäivänä 15.12.2006, jonka jälkeen asiakas suorittaa sille hyväksymistestauksensa. Näin hyväksymistestaus suoritetaan projektin ulkopuolella.