

Testaussuunnitelma

Kohahdus

Helsinki 13.12.2006

Ohjelmistotuotantoprojekti
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Taro Morimoto, Projektipäällikkö
Tuomas Palmanto, Vaatimusmäärittelyvastaava
Mikko Kinnunen, Suunnitteluvastaava
Markus Kivilä, Koodivastaava
Jari Inkinen, Testausvastaava
Paula Kuosmanen, Dokumenttivastaava

Asiakas

Teemu Kerola

Johtoryhmä

Sanna Keskiöja

Kotisivu

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

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	11.10.2006	Ensimmäinen versio
0.2	15.10.2006	Lisätty puuttuvat käyttötapaustestit
0.3	20.10.2006	Korjattu virheitä
0.4	01.11.2006	Korjattu käyttötapaustestejä
0.5	20.11.2006	Lisätty 2. syklin tapaukset
0.6	13.12.2006	Tarkistettu versio

Sisältö

1. Johdanto.....	1
1.1 Tavoitteet.....	1
1.2 Dokumentin rakenne.....	1
2. Sanasto.....	2
3. Testausstrategia.....	3
3.1 Yksikkötestaus.....	3
3.2 Integroititestaus.....	4
3.3 Järjestelmätestaus.....	5
3.4 Muu testaus.....	5
4. Testaukseen käytettävät välineet.....	6
4.1 Testauskirjastojen lisääminen Eclipseen.....	6
4.2 JUnit-testien teko Eclipseessä.....	6
4.3 JUnit-testin rakenne.....	6
4.4 JUnit-testien ajo eclipseessä.....	7
4.5 HttpUnit-testit.....	7
4.6 JSPUnit-testit.....	7
5 Suoritettavat testitapaukset.....	9
5.1 Koodin testaus.....	9
5.2 Käyttötapaukset.....	9
5.2.1 Opettajan käyttötapaukset.....	9
5.2.1 Opiskelijan käyttötapaukset.....	18
5.3 Testien raportointi.....	25
6 Testausaikataulu.....	26

1. Johdanto

Kohahdus on järjestelmä automaattisesti tarkastettavien TTK-91-konekielen harjoitustehtävien luomiseen ja ratkaisemiseen. Järjestelmä on tarkoitettu käytettäväksi opetuksen tukena, opettaessa Tietokoneen toiminta -kurssia. Tietojenkäsittelytieteen opettajat voivat tehdä järjestelmään uusia tehtäviä ja määrittellä kuinka ne tarkastetaan automaattisesti. Tietokoneen toiminta -kurssin opiskelijat ja kurssin tehtävistä kiinnostuneet itseopiskelijat voivat ratkaista tehtäviä ja saada palautetta niiden onnistumisesta.

1.1 Tavoitteet

Tämän testausdokumentin tavoitteena on opastaa testausvaihe, siten että Kohahdus- järjestelmä toimii virheettömästi ja toteuttaa vaatimusdokumentin vaatimukset.

Kaikkia virhetilanteita ei välttämättä löydetä testauksen aikana, mutta pyritään siihen että ohjelma suoriutuu annetusta tehtävästä virheettömästi ja mahdollisimman nopeasti. Tässä testaussuunnitelmassa kuvataan, kuinka Kohahdus testataan, jotta päästäisiin yllä kuvattuihin tavoitteisiin. Testauksen suunnitteludokumentissa on käytetty mallina muiden ohjelmistotuotantoprojektien dokumentteja.

1.2 Dokumentin rakenne

Luvussa 2 sanasto.

Luvussa 3 kuvataan testausstrategia: yksikkötestaus, integrointitestausta, järjestelmätestaus.

Luvussa 4 kerrotaan testaukseen käytettävistä välineistä.

Luvussa 5 kerrotaan, mitä kaikkea halutaan testata.

Luvussa 6 kerrotaan testausaikataulusta.

2. Sanasto

TTK91=Auvo Häkkisen kehittämä ohjelmointikieli, joka läheisesti muistuttaa symbolista konekieltä.

KOKSI=Auvo Häkkisen kirjoittama konekielisimulaattori, joka toteuttaa TTK-91-kielen.

Järjestelmä=Projektimme tuotos, Kohahdus

Ohjelma=Opiskelijan kirjoittama TTK91-ohjelma, eli vastaus johonkin tehtävään

eAssari=Tietokantapohjainen ympäristö ohjelmallisesti tarkastettavien harjoitus- ja koetehtävien suorittamiseen

Titokone=Koski-nimisen Ohjelmistotuotantoprojektiryhmän vuonna 2004 rakentama järjestelmä konekielisten ohjelmien kääntämiseen ja suorittamiseen.

Koski=Vuoden 2004 Ohjelmistotuotantoprojekti joka rakensi konekielen simulaattorin ja debug-ympäristön, eli Titokoneen

Koskelo=Vuoden 2004 Ohjelmistotuotantoprojekti, joka integroi Titokoneen ja eAssarikehyksen yhteen. Ratkaisusta ei tullut kuitenkaan käyttökelpoista, eikä sitä ole otettu käyttöön.

Kriteeri=Sääntö jonkamukaan tehtävän oikeellisuus tarkistetaan. Kriteereitä voi olla monta yhdelle tehtävälle.

Aihepiiri=Tehtävälle täytyy määritellä aihepiiri, johon tehtävä kuuluu.

JUnit=Testaustyökalu javakielelle

HttpUnit=Testaustyökalu verkkosivuille (kuten servletit). Perustuu Junittiin.

JSPUnit=Työkalu erityisesti JSP-sivujen testaamiseen. Perustuu Junittiin.

JSP=JavaServer Pages. Dynaamisten sivujen tekoon kehitetty ohjelmointikieli.

TitoTrainer=Kohahduksen tuotoksen nimi

3. Testausstrategia

Ohjelman testauksessa käytetään testauksen V-mallia. Tässä mallissa ideana on aloittaa yksikkötestauksella pienistä ohjelman osista, kuten metodeista ja luokista. Tämän jälkeen siirrytään testaamaan komponenttien yhteistoimintaa pienissä osissa, jotta saadaan selville missä virhe tapahtuu. Kolmannessa vaiheessa testataan osajärjestelmiä kokonaisuuksina, ja lopulta testataan koko järjestelmän toimivuutta.

Yksikkö- ja integrointitestaus pyritään tekemään testiluokilla mahdollisimman vähällä työllä. Järjestelmätestauksessa testataan ohjelmaa suoraan käyttöliittymien kautta, testaten kaikki palvelut, käyttötapaukset ja toiminnot. Projektin pääpainon ollessa käyttöliittymän selkeydessä, voidaan käyttää tämän asian testaukseen Tietokoneen toiminta- kurssin oppilaita. Kun ohjelmaa testaa joku muu kuin projektin jäsen, on paikalla kirjaamassa testaajan kommentteja joku projektilainen.

Koska Kohahdus-projektin järjestelmä joudutaan siirtämään projektin loppuasiakkaan toimesta varsinaiseen asiakasympäristöön, on tätä varten tehtävä asennusohjeet. Tämän testaus suoritetaan nykyisessä järjestelmässä poistamalla tietokannan taulut ja ohjelmisto, jonka jälkeen luodaan taulut sekä asennetaan ohjelmisto asiakasympäristöön.

3.1 Yksikkötestaus

Yksikkötestauksessa testataan jokainen luokka ja jsp-sivu. Painopiste on toteutuksen testaamisessa. Luokat testaavat luokan tekijät. Testit kirjoitetaan koodin jälkeen, koska testien kirjoitus sekä testien toteuttavan koodin tekeminen on työlästä eikä aikataulu anna sijaa tälle ratkaisulle.

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 tietenkin vain yksi.
- Kutakin lausetta kuvaa solmu ja siirtymistä lauseesta toiseen kuvaa verkon särmä.
- Ehtolauseen tapauksessa siitä lähtee kaksi särmää.
- Case-lauseessa niin monta särmää kuin case-tapauksia, sekä yksi särmä, joihin mikään vaihtoehto ei sovi.
- 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, kuten vaikka yritys hakea tietokannasta henkilön tiedot, jota siellä ei ole. Mahdolliset virhetilanteet luodaan ja tarkistetaan kuinka metodi selviää niistä. Rajapintojen toteutusta testattaessa tarkistetaan, että olion metodit ja attribuutit toimivat yhdessä, jotta ne voivat tuottaa luokan rajapinnan palvelut.

Lausekattavuus voidaan laskea seuraavalla kaavalla:

$$LK=TL/AL, \text{ missä}$$

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. Yleisesti tulisi pyrkiä mahdollisimman lähelle 100%:ia, mutta Kohahdus voinee tyytyä 70%:in. Hyväksymiskriteerinä myös että kaikki yksikön toiminnot ja mahdolliset tilat on testattu. Samoin kaikki poikkeustilanteet. Testaamattomia ohjelman osia ei tule liittää testaukseen vaan tulee luoda tynkiä (tai testata ensin ne osat joiden suorittamiseen ei tarvita muita osia).

Yksikkötestaus suoritetaan luokan ohjelmoijan laitteistolla. Laitteistoissa ei ole paljoa eroja, mutta ainakin testausvastaavan koneella voidaan testata, miten ohjelma suoriutuu huonolla prosessoriteholla.

3.2 Integroititestausta

Integroititestausta tullaan tekemään ns. Bottom-up -strategialla. Tällöin ei rakenneta runkoa kuten Top-down -strategiassa, vaan yksikkötestattuja osia integroidaan toisiinsa yksi kerrallaan ja näin saadaan hiukan suurempia yksiköitä. Jatketaan integroimista, kunnes lopulta saadaan valmis tuote.

Testauksessa testataan Kohahdus-komponenttien toimintaa keskenään, ja lisäksi testataan niiden toimintaa Eassarin ja Titokoneen kanssa. Toisaalta, koska Kohahdus on oma järjestelmänsä, testaus keskittyy vain Eassarin tietokantaan. Titokoneesta halutaan testata tarkasti tehtävien ajaminen ja lopputilojen haku ja oikeellisuus.

Integroititestauksessa tarkoituksena on kuitenkin testata nimenomaan integroitujen yksiköiden palveluiden yhteistyötä, eli rajapintoja. Testausprosessi menee seuraavasti:

1. Selvitetään, mitä rajapintojen palveluja integroidut osat vaativat toisiltaan ja tarjoavat toisilleen: Siis mistä kohtaa osat liittyvät toisiinsa.
2. Tehdään jokaiselle palvelulle arvoanalyysi ja valitaan sen perusteella testisyötteet.
3. Käytetään rajapintaa annetuilla testisyötteillä kutsujan kautta.

Integroititestauksessa ei pitäisi tulla ilmi muuta kuin rajapintaongelmia, sillä kukin yksikkö on jo testattu erikseen ja siten varmistettu että ne toimivat oikein. Vaikka kaikki yksiköt toimisivat oikein voi tulla ongelmia niiden yhteistyössä.

Integroititestauksen ongelmia on esimerkiksi se, että kutsuja ymmärtää rajapinnan väärin, tai kutsuttava palauttaa väärin tulkitun arvon. Rajapintaa saatetaan myös käyttää väärällä tavalla. Kutsuja voi odottaa palvelulta sivuvaikutuksia, jotka eivät toteudu, tai kutsuttava aiheuttaa sivuvaikutuksia, joita kutsuja ei odottanut. Kutsuja voi myös aiheuttaa poikkeustilanteet, johon ei oltu varauduttu. Kutsuja ja kutsuttava voivat myös ymmärtää palvelun syötteiden arvoalueet eri tavoin.

Integroititestauksessa voidaan vaikkapa testata parametrien arvoalueiden ääriarvoilla olevia

testiarvoja tai antaa osoitinparametreille voidaan antaa null-osoitin. Viestinvälitysrajapinnoille voidaan tehdä rasiustestaus. Proseduraaliselle rajapinnalle voidaan tehdä poikkeuksellisia kutsumisjärjestyksiä, kuten vaikka tiedoston luku ennen sen avaamista.

Kahden yksikön integrointitestausta on valmis, kun kaikki yksiköiden välinen yhteistyö on testattu, kaikki poikkeukset on testattu, ja kaikki mahdolliset kutsuttavan aiheuttamat sivuvaikutukset järjestelmään on testattu. Integrointitestausta on valmis, kun kaikki yksiköt integroitu yhteen.

3.3 Järjestelmätestaus

Järjestelmätestaus (System testing) tehdään integrointitestauksen jälkeen. Järjestelmää testataan kokonaisuutena, johon kuuluvat ohjelmiston lisäksi laitteisto ja järjestelmän kanssa yhteistyössä toimivat ulkoiset ohjelmat.

Kohahdusta testataan suoraan käyttöliittymän kautta. Luodaan tunnus, luodaan tehtävä, ratkaistaan tehtävä ym. Toteutustapa eli oliopohjaisuus ei ole enää näkyvillä. Ohjelma testataan sekä Windowsissa että Linuxissa selaimina ainakin Explorer ja Firefox.

Järjestelmän testauksessa pyritään kustakin vaatimuksesta kirjaamaan täyttykö vaatimus Kohahduksessa. Jos vaatimus ei täyty, täytyy kirjata miten vaatimuksen täyttymättömyys käy ilmi. Kirjataan myös mikäli jotain vaatimusta ei voida havaita tai testata.

3.4 Muu testaus

Käyttöliittymää on tarkoitus testata projektin ulkopuolisilla henkilöillä, jotta saadaan selville kuinka selkeä käytettävyys ohjelmalla on. Mikäli selkeitä epäselvyyksiä ilmenee, on syytä muokata käyttöliittymää selkeämpään muotoon.

Lopullisen testauksen suorittaa asiakas. Erimielisyyksistä pyritään keskustelemaan asiakkaan ja ryhmän kesken, mutta pienet toiminnalliset virheet pyritään lähes varmasti korjaamaan. Projektin ollessa kahdessa iteraatiossa voidaan ottaa asiakkaan kommentit ensimmäisessä demossa, ja korjata mahdollisesti puutteet toisessa iteraatiossa. Lopullisen testauksen voidaan todeta olevan valmis kun kaikki virheet on joko korjattu, tai niiden korjaamatta jättäminen on sovittu asiakkaan kanssa.

4. Testaukseen käytettävät välineet

Testaukseen voidaan käyttää JUnit, HttpUnit ja JSPUnit -testaustyökaluja. Jälkimmäiset perustuvat JUnitiin, ja ne on tehty testaamaan verkkosivuja.

4.1 Testauskirjastojen lisääminen Eclipseen

Luodussa projektissa mennään **properties** – sivulle. Valitaan sieltä **Java Build Path** ja **libraries** – välilehti. Painetaan **Add External JARs**:ia ja haetaan tarvittavat kirjastot (.jar) tiedostoista. Lopuksi painetaan **ok**, jonka jälkeen testejä voidaan alkaa suorittamaan.

4.2 JUnit-testien teko Eclipseessä

Valitaan projektista testattava tiedosto, oikealla napilla **new, JUnit testcase**. Eclipse antaa testiluokalle valmiin nimen, ja voidaan valita tekekö Eclipse testille valmiiksi tyngät metodeista. Testien tekoon löytyy apua osoitteessa www.junit.org.

4.3 JUnit-testin rakenne

Alkuun täytyy importoida testikirjasto(t), kuten `import junit.framework.*`; JUnit-testit laajentavat yläluokkaa `TestCase`, joten ne määritellään muodossa:

```
public luokannimi extends TestCase {...}
```

Luokan nimelle syytä antaa jokin Test-päätteinen nimi, kuten vaikka `DBHandlerTest`, jotta ne erottaa normaalista luokista. Luokassa käytettävät yhteiset muuttujat määritellään kuten muissakin luokissa metodien ulkopuolella. Yhteisiin muuttujiin voi vaikkapa sijoittaa testattavan luokan oliot, tosin välttämättä muuttujia ei tarvitse olla yhtään.

`setUp()`- metodissa määritellään ennen kutakin testimetodia suoritettavat alustustoimet. `setUp` ajetaan automaattisesti ennen kutakin testimetodia. Jos testit esimerkiksi käsittelevät listarannetta ja muokkavat sitä, mutta halutaan, että testien alussa on kaikilla sama lähtötilanne, kannattaa alustus suorittaa `setUp`-metodissa. Esimerkki `setUp` metodista:

```
protected void setUp() {  
    f12CHF= new Money(12, "CHF");  
    f14CHF= new Money(14, "CHF");  
}
```

`tearDown()`-metodissa määritetään kunkin testimetodin jälkeen suoritettavat toimenpiteet. Tässä metodissa vapautetaan pysyvät resurssit, jotka `setUp`-metodissa annettiin. Esimerkiksi suljetaan avatut verkkoyhteydet. `tearDown()`-metodia ei yleensä tarvita.

Testimetodit määritellään nimeämällä julkinen testimetodi muodossa `testTestinNimi`. Yhdessä testimetodissa kannattaa testata vain yhtä toimintoa eli käytännössä metodia. Testi kirjoitetaan käyttäen JUnitin `assert`-lauseita ja kirjoittamalla tarvittava määrä muuta koodia. Testien kirjoittajan on syytä varmistaa, että testi kattaa metodin testauksen tarpeeksi hyvin. Esimerkki testimetodista

(käyttää ylläolevaa setUp()-metodia):

```
public void testSimpleAdd() {
    Money expected= new Money(26, "CHF");
    Money result= f12CHF.add(f14CHF);
    assertTrue(expected.equals(result));
}
```

assert-lauseita on erilaisia, kuten *assertTrue*, *assertEquals*, *assertNotNull*. Lisää tietoa asserteista löytyy Assertin API:ssa. <http://junit.sourceforge.net/javadoc/junit/framework/Assert.html>.

simerkit on otettu seuraavasta artikkelista: <http://junit.sourceforge.net/doc/testinfected/testing.htm>.

4.4 Junit-testien ajo eclipsessä

Testimetodeja ei tarvitse kutsua missään luokassa, vaan ajetaan luokka suoraan testinä. Valitaan **Run as**-valikosta JUnit. Testin jälkeen nähdään tiedot ajetusta testistä. *Runs* kertoo ajettujen testimetodien lukumäärän. *Errors* ilmoittaa monessako tapauksessa ohjelman suoritus on keskeytynyt johonkin virheeseen. *Failures* kertoo kuinka monta *assert*-lausetta on mennyt pieleen; tosin ohjelman osa on tällöin suoritettu loppuun. Klikkaamalla virheen antanutta testimetodia nähdään *Failure Trace* – ikkunassa tarkemmin mitä on mennyt pieleen.

4.5 HttpUnit-testit

HttpUnit-testit toimii kuten JUnit-testit eli ne laajentaa *TestCase*:n, niissä on setUp() ja tearDown() ja niissä käytetään myös assert-lauseita. HttpUnittia voidaan käyttää HttpServlet-luokkien testaamiseen. Tosin Kohahdus ei tule käyttämään niitä paljoa. Hyvä opastus aiheesta löytyy seuraavasta linkistä: <http://httpunit.sourceforge.net/doc/tutorial/task1.html>.

HttpUnit on kirjastot jotka täytyy importoida testiluokan alkuun. Testeissä luodaan olio ServletRunner, jolla luodaan client, ServletUnitClient. Tälle voidaan suorittaa operaatioita. Seuraavassa esimerkkejä:

```
ServletRunner sr = new ServletRunner( "web.xml" );
ServletUnitClient client = sr.newClient();
client.setAuthorization( "aUser", "pool-admin" );
WebResponse response = client.getResponse( "http://localhost/PoolEditor" );
WebForm form = response.getFormWithID( "pool" );
```

Asserttia käytetään siis jälleen testauksen oikeellisuuden tarkistamiseen, esimerkiksi vaikka:

```
assertEquals( "Away team 0", "", form.getParameterValue( "away0" ) );
```

HttpUnittia voidaan käyttää myös JSP-sivujen testaukseen, mutta vaikeasti saatavien ohjeiden vuoksi on parempi käyttää testityökalua JSPUnit, joka luotu juuri niitä sivuja varten.

4.6 JSPUnit-testit

JSPUnit on HttpUnitin tavoin lisäys JUnitiin. HttpUnit on monipuolisempi ohjelma, mutta samalla vaikeampi käyttää, joten JSPUnit soveltuu projektille hyvin. Luokat löytää sivulta:

<http://www.dallaway.com/jsptest/>.

Testiluokat laajentavat taas TestCase:a. Alkuun on hyvä määritellä luokalle String-tyyppinen muuttuja, jossa on osoite testattavan tai testattavien JSP-sivujen sijainnille. Testimeteodeissa luodaan Session-olioita, johon voidaan lisätä Cookie-olioita, tai käyttää luokkaa Response, jolla voi hakea Sessionista tietoja. Assert-lauseilla testataan jälleen kerran sivujen oikeellisuus.

Hyvä esimerkkitesti löytyy linkistä <http://www.dallaway.com/jsptest/JSPTTestTest.txt>.

5 Suoritettavat testitapaukset

5.1 Koodin testaus

Ensimmäisessä iteraatiossa tullaan toteuttamaan tai muokkaamaan luokkia User.java, Task.java, Course.java, DBHandler.java ja Criterion.java. Testauksessa näistä luokista täytyy luoda Junit-testiluokka, ja kirjoittaa kattavat testimetodit, jotka käyvät läpi koko koodin. Samoin JSP-sivujen koodi testataan, mahdollisesti käyttäen JSPUnittia.

5.2 Käyttötapaukset

Kohahduksen vaatimukset koostuvat paljolti käyttötapauksista, joten testaus on hyvä tehdä siltä pohjalta.

HYV=hyväksytään kyseinen syöte.

5.2.1 Opettajan käyttötapaukset

Kt1. järjestelmään kirjautuminen

Sivu: Aloitussivu (login.jsp)

Testattavat käyttöliittymäkomponentit:

- tekstikentät Username, Password
- dropdown Language
- painike Sign in

Tekstikentät: Username ja Password

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: molemmat tyhjiä</i>	<i>'User name or password incorrect'</i>
<i>Toinen tyhjä</i>	<i>'User name or password incorrect'</i>
<i>Väärä syöte: Teme xxyzz</i>	<i>"User name or password incorrect"</i>
<i>Väärä salasana: Teemu xxyzz</i>	<i>"User name or password incorrect"</i>
<i>Oikea syöte: Teemu xxyzz</i>	<i>HYV</i>

Painike: Sign in

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Järjestelmä siirtää käyttäjän tehtävien selaussivulle</i>

Kt2. Tehtävien selaaminen

Sivu: Kurssi ja tehtälistaus (teacherTaskList.jsp)

Testattavat käyttöliittymäkomponentit:

- Tekstikenttä Create new course
- painikkeet new course, remove course, course statistics(**kt7**), search users(**kt7**), new task, edit task, edit as new, remove task(**kt6**), log off(**kt8**),

Tekstikenttä: New course

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Välilyönti</i>	<i>'Virheellinen kurssin nimi'</i>
<i>Html-koodia "Tito 2006 Linkki"</i>	<i>'Virheellinen kurssin nimi'</i>
<i>Yli 40 merkkiä: 'sdfd...sdf'</i>	<i>'New course name must be 1-40 characters long'</i>
<i>Oikea "Titou syksy 2006"</i>	<i>HYV</i>

Painike: New course

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Luo kurssin tekstikentässä määritellyssä nimessä, näkyy heti sivulla</i>

Painike: Remove Course

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Kysyy varmistuksen poistetaanko kurssi. Mikäli valitaan kyllä kyseinen kurssi poistuu järjestelmästä.</i>

Painike: Course Statistics

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Siirtää käyttäjän valitun kurssin статистиikkasivulle (showstatistics.jsp), jossa nähdään k kurssin tilastot</i>

Painike: Search users

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Siirtää käyttäjän oppilaiden tietojen hakusivulle (searchusers.jsp)</i>

Painike: New task

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Siirtää käyttäjän tehtävänluontisivulle</i>

Painike: Remove Task

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Kysyy varmistuksen poistetaanko tehtävä. Mikäli valitaan kyllä kyseinen kurssi poistuu järjestelmästä</i>

Painike: Edit task

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Siirtää käyttäjän tehtävänluontisivulle, jonka kenttien oletusarvona kyseisen tehtävän arvot</i>

Painike: Edit as new

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Siirtää käyttäjän tehtävänluontisivulle, jonka kenttien oletusarvona kyseisen tehtävän arvot</i>

Painike: Log off

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Käyttäjä kirjautuu ulos järjestelmästä, siirtyy kirjautumissivulle</i>

Kt3. Tehtävän lisääminen

Sivu: Tehtävänluontisivu (composer.jsp)

Testattavat käyttöliittymäkomponentit:

- painikkeet Add variable, Save, ja kielletyille/vaadituille käskyille olevat napit, Show printable

view, hide printable view (**kt5**)

- Radiopainikkeet Fill-In/Programming, UseModel/Criteria
- tekstikentät Task name, Task Description, Inputs, Secret Inputs, Register values (7kpl), Register values for secret input (7kpl), Variable names, Variable values, Variable secret values, Outputs, Secret Outputs, Code size, Code size quality, data area size, data area size quality, stack max size, stack max size quality, executed commands, executed commands quality, memory references, memory references quality, max commands allowed, Palautekentät kaikille kriteereille, Final feedback success, Final feedback failed.
- Dropdownit, jossa vertailut muuttujien ja rekistereiden arvoille.

Painike: Add Variable

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Lisää uuden muuttujan listaan, jolle kaikki kentät määriteltä</i>

Painike: Save

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Tallentaa tehtävän tietokantaan mikäli kaikki kentät kunnossa, siirtää käyttäjän takaisin tehtälistä sivulle, jossa näkyy lisätty tehtävä</i>

Painikkeet: kielletyt/vaaditut käskyt

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Valitaan yksi vaadituksi</i>	<i>HYV</i>
<i>Valitaan yksi vaadituksi, yksi kielletyksi</i>	<i>HYV</i>
<i>Kaikki kielletty/vaadittu</i>	<i>HYV</i>
<i>Yksi kielletty</i>	<i>HYV</i>
<i>Ei yksikään kielletty tai sallittu</i>	<i>HYV</i>

Radiopainikkeet: Fill-In/Programming

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Fill-In</i>	<i>Kaksi ylimääräistä tekstikenttää alku- ja jälkikoodille</i>
<i>Programming</i>	<i>Ei ylimääräisiä kenttiä</i>

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Fill-In -> Programming -> Fill-In, Fill-In kentissä tekstiä</i>	<i>Teksti säilyy muutoksen mukana</i>

Radiopainikkeet: Use model/Criteria

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Use model</i>	<i>Tekstikenttä mallikoodille</i>
<i>Criteria</i>	<i>Mallikoodin tekstikenttä häviää</i>
<i>Use model -> Criteria -> Use model, mallikoodissa tekstiä</i>	<i>Mallikoodin koodi tulee takaisin näkyviin</i>

Tekstikenttä: Task name

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo, tyhjä</i>	<i>Virheellinen nimi</i>
<i>Välilyönti</i>	<i>Virheellinen nimi</i>
<i>Html-tekstiä: harjoitus 1</i>	<i>Virheellinen nimi</i>
<i>Yksi merkki</i>	<i>HYV</i>
<i>Yli 40 merkkiä: 'sdfg...fdg'</i>	<i>'Task name must be 1-40 characters long'</i>
<i>Oikea syöte: Harjoitus 1</i>	<i>HYV</i>

Tekstikenttä: Task description

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo, tyhjä</i>	<i>Virheellinen kuvaus</i>
<i>Välilyönti</i>	<i>Virheellinen kuvaus</i>
<i>Html-tekstiä: Laske sitä...</i>	<i>HYV</i>
<i>Yksi merkki</i>	<i>HYV</i>
<i>Oikea syöte: Laske sitä...</i>	<i>HYV</i>

Tekstikentät: Inputs ja Secret inputs

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo, tyhjä</i>	<i>HYV, ei syötteitä</i>

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>välilyönti</i>	<i>HYV, ei syötteitä</i>
<i>Yksi syöte: 3</i>	<i>HYV</i>
<i>Negatiivinen -1</i>	<i>HYV</i>
<i>Monta, mutta väärin, 4.4.3 3 5</i>	<i>Virheellinen syöte</i>
<i>Muu kuin numero: E</i>	<i>Virheellinen syöte</i>
<i>Välit eri lailla: 4,5, -4, 3, -2</i>	<i>HYV</i>
<i>Oikea: 3, -5, 7</i>	<i>HYV</i>
<i>Eri määrä syötteitä Inputs ja Secret Inputs</i>	<i>HYV</i>

Tekstikentät: Register values and Register secret values

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>tyhjä</i>	<i>HYV, ei luoda kriteeriä</i>
<i>välilyönti</i>	<i>HYV, ei luoda kriteeriä</i>
<i>Muu kuin numero: r, &</i>	<i>Virheellinen rekisterin arvo</i>
<i>Oikea: 4</i>	<i>HYV</i>
<i>Numeron jälkeen jotain: 4TT</i>	<i>Virheellinen rekisterin arvo</i>
<i>Negatiivinen: -4</i>	<i>HYV</i>
<i>Nolla: 0</i>	<i>HYV</i>

Tekstikentät: Variable names

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: tyhjä tai välilyönti</i>	<i>HYV, ei luoda kriteeriä</i>
<i>Numero: -4, 6</i>	<i>Virheellinen muuttuja</i>
<i>Pitkä nimi: Muuttis</i>	<i>HYV</i>
<i>Ääkkösiä: å</i>	<i>HYV</i>
<i>Muita merkkejä: e-w</i>	<i>Virheellinen muuttuja</i>
<i>Oikea: x</i>	<i>HYV</i>
<i>Numero perässä: X3</i>	<i>HYV</i>
<i>Numero alussa tai välissä 1234r23</i>	<i>HYV</i>

Tekstikentät: Variable and secret variable values

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo:tyhjä</i>	<i>HYV, ei määr itellä muuttujaa</i>
<i>Muu kuin numero, -, r, &</i>	<i>Virheellinen muuttujan arvo</i>
<i>Oikea: 4</i>	<i>HYV</i>
<i>Numeron jälkeen jotain: 4TT</i>	<i>Virheellinen muuttujan arvo</i>
<i>Negatiivinen: -4</i>	<i>HYV</i>
<i>Nolla: 0</i>	<i>HYV</i>

Tekstikentät: Outputs and secret outputs

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo, tyhjä</i>	<i>HYV, ei tulosteita</i>
<i>välilyönti</i>	<i>HYV, ei tulosteita</i>
<i>Yksi syöte: 3</i>	<i>HYV</i>
<i>Negatiivinen -1</i>	<i>HYV</i>
<i>Monta, mutta väär in, 4.4.3 3 5</i>	<i>Virheellinen tuloste</i>
<i>Muu kuin numero: E</i>	<i>Virheellinen tuloste</i>
<i>Välit eri lailla: 4,5, 4, 3, 2</i>	<i>HYV</i>
<i>Oikea: 3, 5, 7</i>	<i>HYV</i>
<i>Eri määr ä syötteitä Outputs ja Secret Outputs</i>	<i>HYV</i>

Tekstikentät: Laadulliset kriteerit: Code size, Code size quality, data area size, data area size quality, stack max size, stack max size quality, executed commands, executed commands quality, memory references, memory references quality

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo:tyhjä</i>	<i>HYV, ei määr itellä arvoa</i>
<i>Muu kuin numero, -, r, &</i>	<i>Virheellinen kriteerin arvo</i>
<i>Oikea: 43</i>	<i>HYV</i>
<i>Numeron jälkeen jotain: 4TT</i>	<i>Virheellinen kriteerin arvo</i>
<i>Negatiivinen: -4</i>	<i>Virheellinen kriteerin arvo</i>
<i>Nolla: 0</i>	<i>HYV</i>

Tekstikenttä: Max commands allowed

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: 10 000</i>	<i>HYV</i>
<i>Muu kuin numero: :, r, &</i>	<i>Virheellinen arvo</i>
<i>Oikea: 43</i>	<i>HYV</i>
<i>Numeron jälkeen jotain: 4TT</i>	<i>Virheellinen arvo</i>
<i>Negatiivinen: -4</i>	<i>Virheellinen arvo</i>
<i>Nolla: 0</i>	<i>Virheellinen arvo</i>
<i>100 000 tai yli</i>	<i>Varoitus, HYV</i>

Tekstikentät: Palautekentät kriteereille

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: oletuspalaute</i>	<i>HYV</i>
<i>Tyhjä</i>	<i>HYV, antaa oletuspalauteen</i>
<i>Yksi merkki</i>	<i>HYV</i>
<i>Palaute mutta esim rekisterille ei ole annettu arvoa</i>	<i>HYV, ei tallenneta kriteeriä</i>
<i>Välilyönti</i>	<i>HYV, antaa oletuspalauteen</i>

Tekstikentät: Final feedback success/Final feedback failed

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: oletuspalaute</i>	<i>HYV</i>
<i>Tyhjä</i>	<i>HYV, antaa oletuspalauteen</i>
<i>Välilyönti</i>	<i>HYV, antaa oletuspalauteen</i>
<i>Yksi merkki</i>	<i>HYV</i>

Dropdownit: Vertailut muuttujien ja rekisterien arvoille

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo =</i>	<i>HYV</i>
<i>Muut, <, <=, >, >=, !=</i>	<i>HYV</i>

Painikkeet: Show printable view, hide printable view

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus show</i>	<i>Poistaa palautteissa muokkausmahdollisuuden ja laatikon</i>
<i>Painallus hide</i>	<i>Palauttaa laatikot</i>

Kt4. Tehtävän muokkaaminen

Sivu: Tehtävän luontisivu (composer.jsp)

Testattavat käyttöliittymäkomponentit:

- Samat mitä tehtävän luonnissa, ei kuitenkaan syytä testata niitä uudestaan. Täytyy kuitenkin katsoa että muutokset tulevat voimaan.

Kt7. Raporttien katsominen

Sivu: Tilastosivut (statistics.jsp ja searchusers.jsp)

Testattavat käyttöliittymäkomponentit:

- tekstikenttä Search users
- painike Search, Remove user

Tekstikenttä: Search users

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Tyhjä syöte</i>	<i>HYV, ei listaa yhtään nimeä</i>
<i>välilyöntejä</i>	<i>HYV, ei listaa yhtään nimeä</i>
<i>1 merkki, esim. p</i>	<i>HYV, Listaa kaikki käyttäjät, joissa esiintyy kyseinen merkki</i>
<i>Oikea syöte: iina</i>	<i>HYV, Listaa kaikki käyttäjät joissa esiintyy kyseinen merkkisarja. Esim. Löytää Tiina Testi</i>

Painike: Search

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Näyttää listan käyttäjistä hakuarvojen mukaan</i>

Painike: Remove user

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Kysyy varmistuksen poistetaanko käyttäjä.</i>

<i>Syöte</i>	<i>Toivottu tulos</i>
	<i>Mikäli valitaan kyllä kyseinen käyttäjä häv iää järjestelmästä.</i>

5.2.1 Opiskelijan käyttötapaukset

Kt9. järjestelmään rekisteröityminen

Sivu: Rekisteröintisivu (signup.jsp)

Testattavat käyttöliittymäkomponentit:

- tekstikentät First name, Last name, Student number, Social security number, E-mail, User name, Password, Password again.
- painike Sign up

Tekstikenttä: First name

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: tyhjä</i>	<i>'Please enter your first name'</i>
<i>Välilyönti tai useita välejä</i>	<i>'Please enter your first name'</i>
<i>Yksi merkki '3'</i>	<i>HYV</i>
<i>Yli 40 merkkiä'fdgd...dfgd'</i>	<i>'First name must be 1-40 characters long'</i>
<i>Html-koodia <l>Taro</l></i>	<i>'Field may not contain characters “ <, >, &” ’</i>
<i>Heittomerkkejä “Tarou”</i>	<i>'Field may not contain characters “ <, >, &” ’</i>
<i>Et-merkki 'Taro&friends'</i>	<i>'Field may not contain characters “ <, >, &” ’</i>
<i>Oikea syöte: Taro</i>	<i>HYV</i>

Tekstikenttä: Last name

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: tyhjä</i>	<i>'Please enter your last name'</i>
<i>Välilyönti tai useita välejä</i>	<i>'Please enter your last name'</i>
<i>Yksi merkki '3'</i>	<i>HYV</i>
<i>Yli 40 merkkiä'fdgd...dfgd'</i>	<i>'Last name must be 1-40 characters long'</i>
<i>Html-koodia <l>Morimoto</l></i>	<i>'Field may not contain characters “ <, >, &” ’</i>
<i>Heittomerkkejä “Morimoto”</i>	<i>'Field may not contain characters “ <, >, &” ’</i>

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Et-merkki: 'Mori&Moto'</i>	<i>'Field may not contain characters ; " <, >, &" '</i>
<i>Oikea syöte: Morimoto</i>	<i>HYV</i>

Tekstikenttä: Student number

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: tyhjä</i>	<i>'Please enter either social security number or student number'</i>
<i>Tyhjä mutta social security number hyväksytty</i>	<i>HYV</i>
<i>Muita kun numeroita: 129as½3u7</i>	<i>'invalid student number'</i>
<i>Liian lyhyt: 123</i>	<i>'student number must be 9 digits'</i>
<i>Liian pitkä: 999999999999999999</i>	<i>'student number must be 9 digits'</i>
<i>Oikea: 047839222</i>	<i>HYV</i>

Tekstikenttä: Social security number

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: tyhjä</i>	<i>'Please enter either social security number or student number'</i>
<i>Tyhjä mutta student number hyväksytty</i>	<i>HYV</i>
<i>Virheellinen muoto 2343-gg-333A</i>	<i>'Social security number must be of Finnish format'</i>
<i>Oikea muoto: 170287-123C</i>	<i>'HYV'</i>

Tekstikenttä: E-mail

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: tyhjä</i>	<i>'Please enter your E-mail address'</i>
<i>välilyönti</i>	<i>'email-address not correct format'</i>
<i>Yksi merkki '9'</i>	<i>'email-address not correct format'</i>
<i>html-koodia: taro.morimoto@cs.helsinki.fi</i>	<i>'Field may not contain characters ; " <, >, &" '</i>
<i>Muita html-erikoismerkkejä: "taro".morimoto@helsinki.fi</i>	<i>'Field may not contain characters ; " <, >, &" '</i>

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Taro.mori&moto@helsinki.fi</i>	<i>'Field may not contain characters ; , < , > , & ' '</i>
<i>Oikea syöte: taro.morimoto@cs.helsinki.fi</i>	<i>HYV</i>
<i>Virheellinen syöte: taro@</i>	<i>'E-mail address not correct format'</i>
<i>Virheellinen syöte: taro@morimoto@helsinki.fi</i>	<i>'E-mail address not correct format'</i>
<i>Virheellinen syöte: taro.morimoto@hel.sinki.fi</i>	<i>'E-mail address not correct format'</i>

Tekstikenttä: User name

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: tyhjä</i>	<i>'Please enter your user name'</i>
<i>Välilyönti tai useita välejä</i>	<i>'Please enter your user name'</i>
<i>Nimi joka jo kannassa</i>	<i>'Username already in use, please choose another'</i>
<i>Alta 3 merkkiä: tm</i>	<i>'Username must be 3-20 characters long'</i>
<i>2 välilyöntiä, 1 merkki: ' t'</i>	<i>'Username must be 3-20 characters long'</i>
<i>Yli 20 merkkiä</i>	<i>'Username must be 3-20 characters long'</i>
<i>Välilyöntejä alussa tai lopussa, merkkejä alle 3-20</i>	<i>HYV</i>
<i>Välilyönti nimen keskellä</i>	<i>'Invalid user name'</i>
<i>Oikea:ninja</i>	<i>HYV</i>

Tekstikentät: Password ja Password again

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Password tyhjä</i>	<i>'Please enter your password'</i>
<i>Password again tyhjä</i>	<i>'Please repeat your password'</i>
<i>Välilyönti salasanan keskellä</i>	<i>'invalid password'</i>
<i>Password ja password again eivät samat</i>	<i>'invalid password'</i>
<i>Password ja password again täsmäävät hyväksytyllä salasanalla</i>	<i>HYV</i>
<i>Alle 6 merkkiä: passu</i>	<i>'Password must be 6-12 characters long'</i>
<i>Yli 12 merkkiä: password12345</i>	<i>'Password must be 6-12 characters long'</i>

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Välilyöntejä alussa tai lopussa</i>	<i>'invalid password'</i>
<i>Välilyönti tai useita välilyöntejä</i>	<i>'invalid password'</i>
<i>Html-koodia '<p>ss</p>'</i>	<i>HYV</i>
<i>Oikea syöte: passunl</i>	<i>HYV</i>

Painike: Sign up

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Järjestelmä siirtää käyttäjän kirjautumissivulle</i>

Kt10. järjestelmään kirjautuminen

Sivu: Aloitussivu (login.jsp)

Testattavat käyttöliittymäkomponentit:

- tekstikentät Username, Password
- dropdown valikko Course, Language
- painike Sign in, Sign up

Tekstikentät: Username ja Password

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: molemmat tyhjiä</i>	<i>'User name or password incorrect'</i>
<i>Toinen tyhjä</i>	<i>'User name or password incorrect'</i>
<i>Väärä syöte: Tarou xxyzz</i>	<i>"User name or password incorrect"</i>
<i>Väärä salasana: Taro xxyzz</i>	<i>"User name or password incorrect"</i>
<i>Oikea syöte: Taro xxyzz</i>	<i>HYV</i>

Dropdown: Course

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Selaaminen</i>	<i>Näyttää kaikki järjestelmän kurssit</i>

Painike: Sign in

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Järjestelmä siirtää käyttäjän dropdownissa</i>

<i>Syöte</i>	<i>Toivottu tulos</i>
	<i>valitun kurssin tehtävien selaussivulle</i>

Painike: Sign up

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Siirtää käyttäjän tunnuksenluontisivulle</i>

Kt11. Tehtävien katselu

Sivu: Tehtävisivu (studentTaskList.jsp)

Testattavat käyttöliittymäkomponentit:

- painikkeet Log off (**kt13**), Do task

Painike: Log off

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Käyttäjä kirjautuu ulos järjestelmästä, siirtyy kirjautumissivulle</i>

Painike: Do task

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>painallus</i>	<i>Käyttäjä siirtyy valitun tehtävän tehtävienratkaisusivulle</i>

Kt12. Tehtävien ratkaiseminen

Sivu: Tehtävisivu (answer_task.jsp)

Testattavat käyttöliittymäkomponentit:

- tekstikentät Simulator input, Program code
- Painikkeet Execute, Titokone report

Lisäksi testataan tehtävien ratkaisemista erilaisilla tehtävillä, ja katsotaan tarkastaako ohjelma tehtävän oikein. Testatut tehtävät kirjataan testausraporttiin.

Painike: Simulator input

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletusarvo: tyhjä tai valmis syöte</i>	<i>HYV, mikäli kelpaa vastaukseksi</i>
<i>välilyönti</i>	<i>HYV, sama mitä tyhjä</i>
<i>Liian vähän syötteitä: esim. tehtävässä 3</i>	<i>Virheellinen syöte</i>

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>syötettä, vain 2 annetaan.</i>	
<i>Liian paljon syötteitä</i>	<i>HYV, otetaan vain ensimmäiset</i>
<i>Monta, mutta väärin, 4.4.3 3 5</i>	<i>Virheellinen syöte</i>
<i>Muu kuin numero: E</i>	<i>Virheellinen syöte</i>
<i>Välit eri lailla: 4,5, -4, 3, -2</i>	<i>HYV</i>
<i>Oikea: 3, -5, 7</i>	<i>HYV</i>

Tekstikenttä: Program code

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oikeellista koodia</i>	<i>HYV</i>
<i>Koodissa virhe</i>	<i>Virheellinen koodi</i>

Tekstikenttä: Program code

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oikeaa koodia, suorittaa tehtävän oikein</i>	<i>HYV</i>
<i>Oikeaa koodia, ei suorita tehtävää</i>	<i>Ei toteuta kriteerejä</i>
<i>Koodissa virhe</i>	<i>Virheellinen koodi</i>

Painike: Execute

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Ohjelma antaa palautteen kriteereistä ja koodin oikeellisuudesta</i>

Painike: Titokone Report

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Avaa ja sulkee titokoneen tiedot tehtävästä</i>

Kt14. Omien tietojen muokkaus

Sivu: Aloitussivu (edit_profile.jsp)

Testattavat käyttöliittymäkomponentit:

- tekstikentät First name, Last name, Student number, Social security number, E-mail, Old password, New password, New password again.
- Painike Save

Passwordeja lukuunottamatta testaus samalla periaatteella, kuten rekisteröitymisessä. Oletusarvoina vanhat tiedot.

Tekstikentät: Old Password

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Oletus: tyhjä</i>	<i>'Please enter you old password '</i>
<i>välilyönti</i>	<i>'Please enter you old password '</i>
<i>Html-koodia '<p>passun1</p>'</i>	<i>'virheellinen salasana'</i>
<i>Väärä syöte: @\$}</i>	<i>'virheellinen salasana'</i>
<i>Oikea syöte: passun1, eli se joka oli jo aiemmin määritelty</i>	<i>HYV</i>

Tekstikentät: New password ja New password again

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Molemmat tyhjiä, ei muuteta</i>	<i>HYV</i>
<i>New password tyhjä, new password again ei tyhjä</i>	<i>'Please enter new password'</i>
<i>New password again tyhjä, New password ei tyhjä</i>	<i>'Please repeat your new password'</i>
<i>New password ja New password again eivät samat</i>	<i>'invalid password'</i>
<i>New password ja New password again täsmäivät hyväksytyllä salasanalla</i>	<i>HYV</i>

Painike: Save

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus</i>	<i>Tallentaa tiedot kantaan, Siirtää käyttäjän takaisin studentTaskList.jsp</i>

V7. Salasanan lähetys

Sivu: Salasanan lähetyssivu (send_email.jsp)

Testattavat käyttöliittymäkomponentit:

- tekstikenttä Username
- Painike Send

Tekstikenttä: Username

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>tyhjä</i>	<i>Käyttäjätunnusta ei löydy</i>
<i>välilyönti</i>	<i>Käyttäjätunnusta ei löydy</i>
<i>Oikea: morimoto</i>	<i>HYV</i>
<i>Väärä: morimot</i>	<i>Käyttäjätunnusta ei löydy</i>

Painike: Send

<i>Syöte</i>	<i>Toivottu tulos</i>
<i>Painallus, cs.helsinki.fi – muotoinen sähköposti</i>	<i>Läheittää sähköpostin, jossa 6-12 merkkiä pitkä uusi salasana.</i>
<i>Painallus, salasana muunlainen</i>	<i>Kerrotaan käyttäjälle, että ottaa yhteyttä opettajaan.</i>

5.3 Testien raportointi

Java-luokkien testeistä kirjoitetaan raportti, joista täytyy käydä ilmi vähintään seuraavat asiat:

DBHandler.java

Testaaja: Taro

Päivämäärä: 18.10.2006

Testin kuvaus: Testattiin JUnit-testiluokalla testDBHandler

Lausekattavuus: noin 80%.

Odotetut tulokset: Tietokantaoperaatiot onnistuu

Havaitut virheet: Ei ongelmia.

Tulokset: Testi suoritettu onnistuneesti.

Testausraportit liitetään Testausraportti-dokumenttiin.

6 Testausaikataulu

Yksikkötestauksessa ei tarvita aikataulua, koska se katsotaan osana koodausta ja suoritetaan sen yhteydessä. Integrointi – ja järjestelmätestaus pyritään toteuttamaan GANT-kaavion mukaan.