

# **Testausdokumentti**

Kivireki

Helsinki 17.12.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (6 ov)

**Projektiryhmä**

Anu Kontio  
Ilmari Helen  
Olli Juvonen  
Joonas Murtola  
Teppo Niinimäki

**Asiakas**

Timo Aalto

**Ohjaaja**

Jari Suominen

**Kurssin vastuhenkilö**

Kimmo Simola

**Kotisivu**

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

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
0.1	10.10.2007	Pohja luotu
0.2	4.11.2007	Ensimmäinen versio
1.0	16.12.2007	Julkaistava versio

## **Sisältö**

<b>1 Johdanto</b>	<b>1</b>
<b>2 Sanasto</b>	<b>2</b>
<b>3 Yksikkötestaus</b>	<b>3</b>
<b>4 Integroititestausta</b>	<b>4</b>
<b>5 Järjestelmätestaus</b>	<b>5</b>
<b>6 Testausaikataulu</b>	<b>6</b>

## **Liitteet**

### **1 Järjestelmätestauksen testetapaukset**

# 1 Johdanto

Kurssikirjatietojen hallintajärjestelmä on syksyn 2007 Ohjelmistotuotantoprojekti-kurssin ryhmän Kivireki aikaansaannos. Aihe projektille syntyi Kumpulan kirjaston tarpeesta saada järjestelmä, jolla kirjaston henkilökunta pystyisi keskitetysti hoitamaan kurssikirjojen hankintaan ja yleiseen hallintaan liittyviä toimenpiteitä. Järjestelmän piiriin kuuluu myös Tietojenkäsittelytieteen laitoksen luennoitsijoille ja kurssihallinnolle toteuttava käyttöliittymä. Kurssihallinto kirjaa järjestelmään luennoitavat kurssikerrat sekä kurssien arvioidut osallistujamäärät. Luennoitsija voi tämän jälkeen ilmoittaa luennoimiensa kurssien kurssikirjatiedot.

Testaussuunnitelmasta käy ilmi, miten kurssikirjatietojen hallintajärjestelmä testaus suoritetaan. Testauksen tarkoituksena on selvittää, vastaako toteutettu ohjelma vaatimusmäärittelyä, tekeekö ohjelma sen mitä sen pitää tehdä ja löytää ohjelmistosta virheitä, joiden johdosta ohjelmisto ei toimi tai toimii väärin.

Testaus jakautuu kolmeen vaiheeseen: yksikkötestaukseen, integrointitestaukseen sekä järjestelmätestaukseen.

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

Yksikkötestauksen apuvälineenä käytetään JUnit-testikehystä. Testauksessa pyritään luomaan mahdollisimman kattava testitapausjoukko, jolla virheet toimintalogiikassa ja koodissa pystytään tehokkaasti löytämään. Testitapausien odotettuja tuloksia verrataan saattuihin tuloksiin.

## 2 Sanasto

**Arvoalueanalyysi** – Tapa valita testattavat syötteet siten, että kaikkien testattavan kohteen syötteiden joukko ositetaan tunnetuilla rajaehdoilla osa-arvoalueiksi.

**JUnit** – Ilmainen testikehys Javalle

**Kattavuuskriteeri** – Pienin luku, joka kattavuus saa olla, että testit voidaan vielä hyväksyä

**Lausekattavuus** – Testissä käytyjen yksikön lauseiden lukumäärän suhde kaikkien yksikön lauseiden lukumäärään

**Haaraumakattavuus** – Kuinka suuressa osassa testattavan yksikön lauseita on käyty

**Lasilaatikkotestaus** – (White-box testing) Testausmenetelmä, jossa testattavan komponentin rakenne tunnetaan ennalta. Tämä mahdollistaa sen, että virhealttiita paikkoja voidaan arvioida tarkastelemalla koodia ja testausta keskittää erityisesti näihin kohtiin. Testien tuloksia verrataan ja analysoidaan suhteessa odotettuihin tuloksiin.

**Mustalaatikkotestaus** – (Black-box testing) Testausmenetelmä, jossa testataan testattavan komponentin toiminnallisuutta. Komponentin sisäisestä toteutuksesta ei olla kiinnostuneita, vaan testaus perustuu syötteiden ja saatujen tulosten vastaavuuksien arviointiin.

**Bottom-up strategia** – Bottom-up strategiassa valmiiksi yksikkötestattuja komponentteja integroidaan toisiinsa yksi kerrallaan ja näin saatuja osakokonaisuuksia testataan yksitellen. Integrointia ja testaamista jatketaan kunnes ollaan saatu valmis järjestelmä.

### 3 Yksikkötestaus

Yksikkötestauksessa testataan järjestelmän luokat ja JSP-sivut käyttäen lasilaatikkoperiaatetta. Yksikkötestaus jatkuu koko toteutusvaiheen ajan. Testauksessa pyritään testaamaan komponenttien pienimmät loogiset osat, ja testauksen suorittaa ko. komponentin ohjelmoija ohjelmointityön yhteydessä. Ohjelmoitu funktio testataan kattavalla joukulla syötteitä (Arvoalueet käsitellään eri tavoin riippuen siitä, onko kyseessä luku vai merkkijono):

- Arvoalueen ääriarvokohdat
- Nollakohta
- NULL-arvoilla, tyhjillä ??-stringeillä
- Laillinen, oikea arvo
- Laillinen, väärä arvo
- Laiton arvo

Jos funktion toimintaa ei pystytä eristämään muista funktioista, pyritään löytämään minimijoukko funktioita joita testitapaus vaatii. Yksikkötestaus katsotaan onnistuneeksi, kun määritellyt testitapaukset on suoritettu onnistuneesti. Jokaisen komponentin kohdalla pyritään vähintään 80% haaraumakattavuuteen. Yksikkötesteihin käytettävät testiluokat ovat kansiossa src/test.

## 4 Integroititestausta

Integroititestausta testataan moduulien väliset yhteydet ja rajapinnat Bottom-up periaatteella. Top-down -strategiasta poiketen yksikkötestattuja osia integroidaan toisiinsa yksi kerrallaan ja näin saatuja osakokonaisuuksia testataan yksitellen. Integroititestausta on valmis, kun kaikki yksiköt on integroitu yhteen. Testauksen eteneminen vaiheittain:

- Selvitetään, mitä rajapintojen palveluja integroidut tarjoavat toisilleen.
- Tehdään palveluille arvoalueanalyysi ja valitaan sen perusteella testisyöteet.
- Käytetään rajapintaa annetuilla testisyöteillä kutsujan kautta.

Käytännössä integroititestausta tehdään seuraavassa järjestyksessä:

- Testataan Strutsin kartoitus Action-luokkien kanssa.
- Testataan Persistenssiluokat Hibernaten kartoituksen sekä tietokannan kanssa.
- Testataan Strutsin ja Hibernaten yhteistoiminta.
- Testataan Strutsin ja Hibernaten toiminta JSP-sivujen kanssa.
- Testataan rajapinnat Action-luokkien ja Business-luokkien välillä.
- Testataan rajapinnat Action-luokkien ja JSP-sivujen välillä.

Integroititestausta katsotaan onnistuneeksi, kun kaikki komponentit on integroitu ja yhteyksiä sekä rajapintoja testaavat testitapaukset on onnistuneesti ajettu.

## 5 Järjestelmätestaus

Järjestelmätestauksessa testataan integroitu valmis järjestelmä Blackbox-periaatteella. Tavoitteena on olennaisesti testata toimiiko järjestelmä aina oikein, eli määritellyllä tavalla. Järjestelmätestauksen testitapaukset luodaan suoraan käyttäjä -ja järjestelmävaatimusten pohjalta. Järjestelmätestaus katsotaan onnistuneeksi, kun kaikki määrittelyvaiheen käyttötapaukset ja toteutetut toiminnalliset järjestelmävaatimukset sekä kaikki ei-toiminnalliset järjestelmävaatimukset on testattu. Järjestelmätestauksen testitapaukset liitteenä.



## **6 Testausaikataulu**

Yksikkö & Integrointitestausta jatketaan läpi toteutusvaiheen (takarajana järjestelmätestauksen aloituspäivä 2.12).

## **Liite 1. Järjestelmätestauksen testetapaukset**