

TIETOKANTA MERIKOTKIEN SEURANTAAN
Testaussuunnitelma

Versiohistoria:

Versio	Päivämäärä	Kuvaus	Tekijä
1.0	14.12.2007	Virallinen versio	Janne Piippo

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos
Ohjelmistotuotantoprojekti Merikotka

Projektiryhmä:
Janne Piippo
Juha Hiekkämäki
Pekka Maksimainen
Petri Setälä
Teemu Pulkkinen
Tuire Huhtamäki

Ohjaaja:
Sanna Keskiöja

Sisällysluettelo

1 Johdanto	1
1.1 Dokumentin tarkoitus	1
1.2 Kuvaus tuotteesta	2
1.3 Käytetyt ohjelmointikielet ja niiden testauspiirteet.....	2
2 Sanasto	2
3 Yksikkötestaus	5
3.1 Lähestymistapa.....	5
3.2 Testattavat kohdat	6
3.3 Hyväksymiskriteerit.....	7
3.4 Yksikkötestausohje yksikön toteuttajille.....	7
4 Integroititestaus	8
4.1 Lähestymistapa.....	8
4.2 Testattavat kohdat ja testauksen toteutus	9
4.3 Testauksen rajoitteita ja ongelmakohtia.....	9
4.4 Integroititestauksen ajankohta	9
4.5 Integroititestauksen hyväksymiskriteerit	9
5 Järjestelmättestaus	10
5.1 Järjestelmättestauksen tavoite	10
5.2 Testattavat kohdat	10
5.3 Järjestelmättestauksen suoritus.....	15
6 Hyväksymistestaus.....	15
7 Testausaikataulu	15
7.1 Määräajat.....	16
7.2 Testivaiheiden tekijät	16
7.3 Testauksen dokumentointi.....	16
8 Testausraportit	16
8.1 Yksikkö- ja integroititestaus	16
8.2 Järjestelmättestaus.....	25
8.3 Hyväksymistestaus	32

1 Johdanto

Testaussuunnitelma sisältää tiedot siitä, miten toteutettavan järjestelmän testaus tehdään.

Tämä suunnitelma perustuu yleisesti ohjelmistotuotannossa käytettyyn V-malliin, jossa testaus jakautuu eri vaiheisiin, jotka voidaan rinnastaa ohjelmistoprojektin kulun kanssa. V-mallin mukaisesti testaussuunnitelma on jaettu kolmeen vaiheeseen: yksikkötestaukseen, integrointitestaukseen sekä järjestelmätestaukseen.

- Yksikkötestaus suoritetaan koodauksen aikana ja siinä testataan komponentit ja niiden toimivuus.
- Integrointitestauksessa testataan yksikkötestauksen läpäisseiden komponenttien rajapinnat.
- Järjestelmätestauksessa testataan järjestelmän toimivuus loppukäyttäjän näkökulmasta.

Yksikkötestauksessa testataan järjestelmän pienimmät loogiset osat, kuten luokkia ja metodeita. Integrointitestaus on rajapintatestausta, jossa testataan integroitujen yksiköiden palveluiden yhteistyötä. Järjestelmätestauksessa testataan järjestelmän toimivuutta kokonaisuutena.

Testaussuunnitelmassa selvitetään kustakin testausvaiheesta lähestymistapa eli miten testataan, testattavat kohteet eli mitä testataan ja hyväksymiskriteerit eli milloin on testattu tarpeeksi.

Toteutus ja testaus limittyvät suurelta osin yhteen. Testausta tulee suorittaa toteutuksen rinnalla. Tällöin on muistettava kirjata osa toteutukseen käytetyistä tunneista myös testaukseen. Jos jossain testausvaiheessa järjestelmästä löytyy virheitä, ne pyritään korjaamaan, jos se on aikataulun puitteissa mahdollista. Jos virheitä ei korjata, ne dokumentoidaan.

Testaussuunnitelman pohjana on käytetty Kimmo Simolan ohjeistusta syksyn 2007 ohjelmistotuotantoprojektin testaussuunnitelman laadintaa ja testausta varten.

1.1 Dokumentin tarkoitus

Tämä dokumentti on testaussuunnitelma Merikotka-työryhmän Helsingin yliopiston tietojenkäsittelytieteen laitoksen syksyn 2007 ohjelmistotuotantoprojektin aikana tuottaman järjestelmän testausta varten. Dokumentti määrittelee projektissa käytettävän testausprosessin, käytettävät menetelmät, testauksen kattavuuden ja testauksen raportoinnin.

Testaussuunnitelman tavoitteena on toimia testausvaiheessa testauksen ohjeistuksena sekä ohjelman mahdollisen jatkokehityksen aikana testauksen toistettavuuden ja suunnittelun apuna. Tavoitteena on, että Haliaaetus-järjestelmä toimii projektin päätyttyä virheettömästi ja toteuttaa vaatimusdokumentissa määritellyt vaatimukset.

1.2 Kuvaus tuotteesta

Projektin tarkoituksena on olemassa olevan Haliaeetus-järjestelmän toiminnallisuuden kehittäminen ja parantaminen sekä uusien toiminnallisuuden toteuttaminen. Haliaeetus-järjestelmä, johon jatkossa viitataan lyhyesti järjestelmänä, on merikotkien pesimätietojen seurantaan ja ylläpitoon kehitetty ohjelmisto, jonka käyttäjinä toimivat WWF:n merikotkatyöryhmän valtuuttamat henkilöt. Järjestelmä on kuvattu tarkemmin suunnitteludokumentissa. Projekti on jatkoa aikaisempien työryhmien ohjelmistotuotantoprojekteille, erityisesti ryhmän Hali2 keväällä 2004 toteuttamalle projektille.

Järjestelmä toteutetaan ja testataan sen toteutusympäristössä eli Tomcat servlet -ympäristössä tietojenkäsittelytieteen laitoksen koneella db.cs.helsinki.fi ja järjestelmä on tarkoitettu käytettäväksi Firefox-selaimen kautta (Firefox versio 2.0.0 tai uudempi).

1.3 Käytetyt ohjelmointikielät ja niiden testauspiirteet

Ohjelmointikielenä tuotantoprojektissa käytetään Java-ohjelmointikieltä. Muita toteutuksessa käytettyjä kieliä ovat HTML, CSS, FreeMarker Template Language, JavaScript ja Oraclen SQL-lauseet.

Oliokielenä Java tuo testaamiseen omat haasteensa, jotka liittyvät lähinnä periyymiseen sekä dynaamiseen sidontaan. Ominaisuuksien periytyminen ja polymorfismi voivat hankaloittaa suoraviivaista rakenteellista testaamista. Olioiden väliset yhteydet voivat tuottaa monimutkaisia rakenteita, joiden testaaminen on hankalaa.

HTML- ja CSS-koodin testauksen apuna on mahdollista käyttää yleisesti saatavilla olevia validaattoreita, jotka tarkistavat koodin syntaksin olevan spesifikaation mukainen.

2 Sanasto

Sanastoon on koottu Merikotka-ohjelmistotuotantoprojektissa toteutettavaan ohjelmistoon ja sen testaamiseen liittyviä termejä.

Ajuri Testiajurit simuloivat yksiköitä, jotka kutsuvat tai käyttävät testattavan ohjelmistokomponentin palveluita. Ajuri vastaanottaa ja muokkaa testiaineiston testattavan komponentin vaatimaan muotoon, kutsuu komponenttia ja vastaanottaa tuloksen.

Arvoanalyysi Testitapausjoukon rajaaminen järkeviksi osa-arvoalueiksi, joiden reunoilta valitaan testauksessa käytettävät arvot.

Bottom-up -strategia Integrointitestauksen strategia jossa integrointi aloitetaan yksiköistä, joita integroidaan yhteen kunnes koko järjestelmä on koottu.

CSS (Cascading Style Sheets) Tyylimäärittelyt määrittelevät kuinka dokumentti esitetään ruudulla ja tulosteissa. Mahdollistavat värien, fonttien, asemoinnin jne. lisäämisen HTML-dokumentteihin. CSS-määrittelyt voidaan lisätä suoraan dokumenttiin tai ne voidaan määrittää css-tyylitiedostossa.

EUCT (Extended Use Case Test), laajennettu käyttötapaustesti. Järjestelmätestaus toteutetaan Extended Use Case-testinä, jossa käytetään niin kutsuttuja laajennettuja käyttötappauksia. Yksi laajennettu käyttötappaus määrittelee joukon skenaarioita, jotka kertovat kuinka järjestelmä toimii tietyillä syötteillä, eli mitä tuloksia se palauttaa

FTL ks. Freemarker-template.

Freemarker-template FTL (FreeMarker Template Language) -kielellä luotuja HTML-templaatteja.

Haaraumakattavuus Testausmenetelmä, jolla pyritään käymään läpi testiyksikön jokainen haaraumakohta (esim. if-lauseiden vaihtoehdot).

Haliaeetus järjestelmä, järjestelmä Toteutettava järjestelmä, joka sisältää käyttöliittymän, tietokannan ja näiden välillä olevat toiminnallisuudet.

HTML (HyperText Markup Language) Rakenteellinen merkkaukieli, jolla webin sivut kirjoitetaan. JavaScript on sulautettu HTML-sivujen sisälle.

Integrointitestaus Testausvaihe, jossa toimivia yksiköitä liitetään toisiinsa. Rajapintojen testausta.

Java Projektissa käytettävä ohjelmointikieli.

JavaScript Selaimessa suoritettava skriptauskieli, jolla voidaan toteuttaa yksinkertaisia dynaamisia elementtejä HTML-dokumentissa.

Java-servlet Java-kielinen ohjelma web-palvelimella, jolla tuotetaan dynaamisia web-sivuja.

Java-luokka Java-luokka kuvaa olion rakenteet (attribuutit) ja käyttäytymisen (metodit). Luokat ovat yksikkötestauksen kohteina.

JDBC (Java Database Connectivity) Sunin kehittämä ohjelmointirajapinta (API) Java-sovellusten ja tietokantojen välillä.

JUnit Testikehys Java-luokkien yksikkö- ja integrointitestaukseen.

Järjestelmä ks. Haliaeetus järjestelmä

Järjestelmätestaus Ohjelmistotuotteen (järjestelmän) testaus kokonaisuutena.

Kattavuus Luku, joka kertoo kuinka hyvin suoritettut testit ovat testanneet testatun yksikön rakennetta.

Kattavuuskriteeri Testauksen kattavuuden minimiarvo.

Käyttötapaus Käyttötapauksia käytetään toimijan ja järjestelmän välisen vuorovaikutuksen kuvaamiseen. Toimija voi olla henkilö, toinen tietojärjestelmä jne. Käyttötapauksessa kuvataan toimijan tavoite jonkin päämäärän saavuttamiseksi, ja mahdollisimman yksityiskohtaiset tiedot tilanteen taustoista.

Laajennettu käyttötapaus ks. EUCT

Lasilaatikkotestaus (White-box testing) Lasilaatikkotestausta kutsutaan rakenteelliseksi testausmenetelmäksi. Se perustuu testattavan kohteen rakenteen tuntemiseen. Mahdollisia virhealttiita paikkoja voidaan arvioida tarkastelemalla koodia ja keskittää testaus näihin kohtiin. Testien tuloksia verrataan ja analysoidaan suhteessa odotettuihin tuloksiin.

Lausekattavuus Lausekattavuudella tarkoitetaan testeissä läpikäytyjen yksikön lauseiden lukumäärä / yksikön kaikkien lauseiden lukumäärä. Lausekattavuutta käytetään yksikkötestauksessa kertomaan paljonko yksikön lauseista on testattu.

Luokkatestaus katso yksikkötestaus

Metodi Java-luokan sisällä oleva aliohjelma, jota voidaan kutsua itse luokasta tai toisesta Java-luokasta.

Mustalaatikkotestaus (Black-box testing) Mustalaatikkotestauksessa testataan testattavan komponentin toiminnallisuutta. Siinä komponentin sisäinen rakenne (esimerkiksi koodi ja tietorakenteet) ei ole näkyvissä. Testaus perustuu syötteiden ja sovelluksen antamien tulosten analysointiin. Saatuja tuloksia verrataan odotettuihin, jolloin voidaan päätellä sovelluksen toimivuus testitapauksessa.

Osa-arvoalue ks. Arvoalueanalyysi.

Servlet ks. Java-servlet.

SQL Monien tietokantajärjestelmien käyttämä kieli.

Tynkä Yksikkö- ja integraatiotestauksessa yksikön tarvitseman testaamattoman komponentin palvelua korvaava palvelu. Esimerkiksi metodia testattaessa tyngät korvaavat kutsut muihin metodeihin ja olioihin.

V-malli Kun toteutusta kuvaavaan vesiputousmalliin yhdistetään joka vaihetta vastaava testaus (esim. vaatimusmäärittelystä hyväksymistestaus), aikajanelle syntyy V-kirjaimen muotoinen malli, jonka vasen sakara kuvastaa prosessin vaiheita ja oikea kunkin testaus-ta.

Yksikkö Koodin selkeästi rajautuva kompakti osio, esim. luokka tai metodi. Myös tietokannan luonnissa käytetyt SQL-lauseet muodostavat yksiköitä.

Yksikkötestaus Yksiköiden testaamista (ks. yksikkö).

3 Yksikkötestaus

Yksikkötestauksessa testauksen kohteena ovat pienimmät loogiset ohjelmistonosat. Projektissa tällaisia ovat esimerkiksi Java-luokat, metodit ja muut toiminnallisuutta sisältävät yksiköt, kuten templatet. Myös pieniä vahvasti toisiinsa sitoutuneita luokkia voidaan testata yksikkötestauksen menetelmin.

Tavoitteena on se, että kaikki yksiköt tulee testattua riittävän kattavasti ja todennettua, että ne toteuttavat niiltä vaaditut tehtävät.

Yksikkötestauksessa on tyypillisesti lasilaatikkotestausta (white box testing), joka perustuu testattavan kohteen rakenteen tuntemiseen.

Testauksen automatisointiin on kehitetty useita työkaluja, kuten Java-kielisten luokkien yksikkötestauksen apuvälineenä käytettävä JUnit-testikehys, joka voidaan sisällyttää Eclipse Java -kehitysympäristöön. Parhaimmillaan testauksen automatisointi voikin vähentää testauskustannuksia jopa 80 %. Automatisointi ei kuitenkaan ole ideaali kertaluonteisia testaustapahtumia varten, sillä automatisoinnin tuottama säästö muodostuu pääosin testauksen toistoista.

Automatisoinnin toteuttaminen vie tavallisesti aikaa n. 3 - 10 kertaa enemmän kuin manuaalisen testin kertasuoritus. Tämän vuoksi ohjelmistotuotantoprojektin luonteisessa kertaluonteisessa kehitystyössä ei välttämättä ole järkevää sijoittaa niukkoja ajallisia resursseja testauksen automatisointiin. Käytettävissä olevien testaustyökalujen kartoitukseen ja niihin tutustumiseen kuluu helposti paljon aikaa, jollei ryhmässä ole kokeneita testauksen ammattilaisia. Perinteinen manuaalinen testaus on tällöin käytännöllisempää, ja ennen kaikkea vähemmän resursseja vievää.

Merikotka-projektissa yksikkötestaus toteutetaan yksikön toteuttajan valinnan mukaisesti manuaalisesti tai ohjelmallisesti, yksikön toteuttajan toimiessa testaajana.

3.1 Lähestymistapa

Yksikkötestauksessa testaustapaukset perustuvat ohjelman koodiin ja sen rakenteisiin. Tämän vuoksi yksikkötestauksen luontevin suorittaja pienimuotoisessa projektissa on yksikön toteuttaja.

Koska työn kohteena on jo olemassa olevan järjestelmän kehittäminen, voidaan resurssien rajallisuuden vuoksi olettaa, että projektin puitteissa toteutettujen yksikköjen tai yksikköjen osien hyödyntäessä järjestelmän olemassa olevia komponentteja niiden olevan riittävän kattavasti testattuja aiempien toteuttajien toimesta. Näin ollen ei näissä tapauksissa tarvitse tehdä testiajureita tai -tynkiä.

Java-koodi testataan yksikkövaiheessa 90 -100% lausekattavuudella, ja mahdollisuuksien ja aikataulun puitteissa myös mahdollisimman haaraumakattavasti.

HTML-, CSS-, FTL- ja SQL-koodi käydään läpi 100% lausekattavuudella.

Tietokanta-ajoissa käytetyt scriptit toteutetaan Javalla, ja testataan kuten muukin Java-koodi. Lisäksi scripteihin sisältyvät SQL-lauseet käydään läpi 100% lausekattavasti.

HTML-koodin sisällä käytetty JavaScript käydään läpi 100% lausekattavuudella.

3.2 Testattavat kohdat

Kaikki dataa muokkaavat metodit testataan sekä kelvollisilla että kelpaamattomilla syötteillä. Mikäli metodin käyttäytyminen voi muuttua riippuen sen tilasta, tulee testaus toistaa eri tiloissa.

Syötteiden optimoimiseen käytetään arvoalueanalyysin periaatteita, jossa testiparametreiksi valitaan metodin ulkoisten tai sisällä olevien rajoitusten lähellä olevat arvot, sekä jokin arvo luokan arvoalueen "keskeltä". Lisäksi kelpaamattomilla syötteillä testataan, että metodi osaa käsitellä niitä tarkoituksenmukaisella tavalla ja että poikkeusten käsittely on kunnossa.

Arvoalueanalyysissä testin syötearvoalue ositetaan osa-arvoalueiksi seuraavasti:

Lukuarvoista testataan osa-arvoalueet:

- NULL
- minimiarvo-1
- täsmälleen minimiarvo
- minimiarvon ja maksimiarvon väliltä
- täsmälleen maksimiarvo
- maksimiarvo+1

Taulukoista testataan osa-arvoalueet:

- indeksi negatiivinen, nolla, maksimi, maksimi+1, tyhjä taulukko

Merkkijonoista testataan osa-arvoalueet:

- tyhjä merkkijono, merkkijonon pituus 0.
- merkkijonon pituus 1.
- arvo merkkijonon keskivaiheilta
- merkkijonon pituus maksimi.
- merkkijonon pituus ylittää maksimin.
- erikoismerkit merkkijonoissa.

Lisäksi tulisi varmistaa, että kaikkiin koodiriveihin päästään jostain käsiksi ja poikkeuksien käsittely on kunnossa.

Testausvastaava raportoi testaukseen käytettäviä resursseja ja aikatauluja. Saamiensa tietojen perusteella hän kirjoittaa testiraportin.

3.3 Hyväksymiskriteerit

Testattu yksikkö hyväksytään yksikkötestausvaiheen lopussa, kun seuraavat vähimmäisvaatimukset on saavutettu:

- Kaikki sen palvelut on onnistuneesti testattu määrätyillä osa-arvoaluilla
- Kaikki tilat on testattu
- Kaikki määritellyt poikkeustilanteet on testattu
- Lausekattavuus on vähintään 90%, yksinkertaisimmissa yksiköissä 100%

Koodin läpikäynnin kattavuus voidaan määritellä useilla tavoilla. Yksinkertaisimmillaan tutkitaan kuinka monessa testattavan yksikön lauseessa on käyty, kun testitapaukset on suoritettu. Tätä kutsutaan lausekattavuudeksi.

Kaikki testattavat ehdot eivät käy ilmi vain lauseita tarkastelemalla. Tällöin haaraumakattavuus on usein lausekattavuutta käytännöllisempi mittari. Se lasketaan kaavalla

$HK = TS/AS$, missä

- HK = haaraumakattavuus
- TS = testeissä käytyjen siirtymien lukumäärä testattavassa yksikössä
- AS = testattavan yksikön kaikkien siirtymien lukumäärä.
- Siirtymä = yhteys kahden lauseen välillä.

Esimerkiksi if-lauseesta on kaksi siirtymää: 1) ehto on tosi ja 2) ehto on epätosi, myös silloin kun erillistä else-haaraa ei ole. Siksi projektissa pyritään resurssien puitteissa suorittamaan myös haaraumakattavuustestausta, tavoitteena n. 80% kattavuus, mutta sitä ei määritellä hyväksymiskriteeriksi.

3.4 Yksikkötestausohje yksikön toteuttajille

Aluksi:

- Selvitä mitä toteutettavan yksikön halutaan tekevän, mitä parametreja se käsittelee ja palauttaa.
- Käytä testauslomakkeita ja palauta ne testausvastaavalle täytettynä testauksen päätyttyä.

Testaustekniikoita:

1. Toiminnalliset testit (lähinnä arvoalueanalyysit)
2. Staattinen testaus, koodin läpikäynti (esim. kynällä ja paperilla, kylmällä logiikalla)
3. Lasilaatikkotestaus (mahdollisia virhealttiita paikkoja arvioidaan tarkastelemalla koodia, keskitetään testaus niihin)
4. Yksikön tilan mahdollisten tilan muutoksien havaitseminen (jos yksikkö voi olla eri tiloissa, uusi arvoalueanalyysi voi olla hyödyksi)
5. Ei-toiminnalliset testit (tarvittaessa, esim. tietoturvallisuus, käytettävyys, tehokkuus)

Testauksen tulokset:

1. Testatut yksiköt

2. Testidataa (vähintään kuvaus siitä, mitä on tehty, millä syötteillä testattu, mitä arvoja saatu, esiintykö poikkeuksia ja miten ne käsiteltiin, lause- ja haaraumakattavuus).

3. Palautettu testituloslomake.

4 Integroititestausta

Integroititestauksessa testataan yksiköiden (Java-luokat, Freemarker-templatet) väliset rajapinnat. Integroititestaukseen luokat etenevät kun ne ovat läpäisseet luokkatestauksen, eli on todettu, että ne toimivat oikein yksinään. Integroititestauksen tarkoitus on varmistaa komponenttien toiminta yhteistyössä toistensa kanssa ja paikantaa mahdollisia virheitä komponenttien välisistä rajapinnoista. Integroititestausta on tyypillisesti black box -testausta, joka ei perustu komponentin sisäiseen rakenteeseen, vaan rajapintojen kautta välittyvien syötteiden ja tulosten analysointiin.

4.1 Lähestymistapa

Integroititestausta tullaan tekemään ns. bottom-up -strategialla jossa yksikkötestattuja osia integroidaan toisiinsa yksi kerrallaan kunnes kaikki yksiköt on integroitu järjestelmään.

Integroititestauksessa testataan järjestelmän komponenttien toimintaa keskenään, sekä niiden toimintaa tietokannan kanssa. Testauksessa keskitytään yksiköiden rajapintoihin. Testausprosessi etenee seuraavasti:

1. Selvitetään, mitä rajapintojen palveluja integroidut osat vaativat toisiltaan ja tarjoavat toisilleen.

2. Tehdään jokaiselle palvelulle arvoalueanalyysi 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.

Integroititestauksessa mahdollisesti ilmeneviä ongelmia voivat olla muun muassa:

- kutsuja ymmärtää rajapinnan väärin
- kutsuttava palauttaa väärin tulkitun arvon
- rajapintaa käytetään väärällä tavalla
- kutsuja voi odottaa palvelulta sivuvaikutuksia, jotka eivät toteudu, tai kutsuttava aiheuttaa sivuvaikutuksia, joita kutsuja ei odottanut
- kutsuja voi aiheuttaa poikkeustilanteen, johon ei oltu varauduttu.
- kutsuja ja kutsuttava voivat ymmärtää palvelun syötteiden arvoalueet eri tavoin.

Kahden yksikön integroititestausta on valmis, kun kaikki yksiköiden välinen yhteistyö on testattu, mukaan lukien virheiden ja poikkeusten testausta. Integroititestausta vaihe päättyy, kun kaikki yksiköt on testatusti integroitu yhteen.

4.2 Testattavat kohdat ja testauksen toteutus

Integroitintestaus uusille komponenteille pyritään tekemään heti kun uusi komponentti on valmis, eli se on ohjelmoitu, dokumentoitu ja yksikkötestattu.

Testausta varten on määriteltävä mihin kaikkiin järjestelmän yksiköihin testattava yksikkö on suoraan tai välillisesti yhteydessä ja mitä rajapintojen palveluja integroitavat palvelut pyytävät toisiltaan ja tarjoavat toisilleen.

Testisyötteillä käytetään testattavaa rajapintaa kutsujan kautta. Kun osien liittyminen toisiinsa rajapintojen tarjoamien palveluiden kautta on selvitetty, valitaan sopivat testisyötteet ja rajapinnan toimivuus testataan.

Projektin ja järjestelmän luonteesta johtuen kokonaan uusia Java-luokkia tulevat olemaan vain kuvatoimintoihin liittyvä operate- validate ja serach-luokat sekä muutama report-luokka. Kuitenkin useita olemassa olevia java-luokkia täytyy muokata vastaamaan uusien toteutusten vaatimuksia. Olemassa oleviin luokkiin lisätyt toiminnallisuudet tulee myös integroitintestata.

Suuri osa toteutusta tulee olemaan Template-tiedostojen toteuttaminen, joiden yhteydet muuhun järjestelmään tulee myös integroitintestata.

Järjestelmän tietokantayhteydenottojen kattava integroitintestaaminen voi olla hankalaa, joten tietokantaan lähetettävien hakujen validointiin tulee kiinnittää erityistä huomiota, pääsääntöisesti hyödyntämällä Validate-pakkauksen luokkia sekä varautumalla poikkeustilanteiden käsittelyyn.

4.3 Testauksen rajoitteita ja ongelmakohtia

- Oliopohjaisissa järjestelmissä teoreettisia yhteyksiä voi olla n^2-n kappaletta, n on luokkien lukumäärä. Kaikkien eri yhteyksien kattava läpikäynti on usein hyvin vaikea toteuttaa. Tärkeimmät ja käytetyimmät yhteydet ovat testauksessa etusijalla.
- Perintä, monimuotoisuus ja ylikuormittaminen aiheuttavat ongelmia
- Rajapinnan väärinymmärrys (kutsuja / kutsuttava).
- Rajapinnan käyttö ei-toivotulla tavalla.
- Sivuvaikutukset (kutsujan odottamat, kutsuttavan aiheuttamat, poikkeustilanteet).
- Arvoalueiden väärinymmärrykset ja arvoalueiden rajat

4.4 Integroitintestauksen ajankohta

Integroitintestaus voidaan aloittaa heti kun integroitavaksi tarkoitetuille yksiköille on suoritettu yksikkötestaus siten, että ne täyttävät vaaditut hyväksymiskriteerit.

4.5 Integroitintestauksen hyväksymiskriteerit

Kun yksiköiden välinen toiminta (niiden rajapinnat) on osoitettu toimiviksi ja kaikki komponentit on integroitu yhteen on integroitintestaus suoritettu. Tällöin voidaan ryhtyä testaamaan koko järjestelmää.

Käyttöliittymä testataan järjestelmätestauksen yhteydessä.

5 Järjestelmätestaus

Järjestelmätestaus tehdään integrointitestauksen jälkeen. Järjestelmä testataan kokonaisuutena, johon kuuluvat ohjelmiston lisäksi tietokanta, laitteisto ja järjestelmän kanssa yhteistyössä toimivat ulkoiset ohjelmat kuten selain. Järjestelmä testataan käyttöliittymän kautta.

Järjestelmän testaus ja testitapausten valinta perustuu käyttötapauksiin ja käyttäjävaatimuksiin. Kustakin vaatimuksesta kirjataan täyttyykö vaatimus. Jos vaatimus ei täyty, täytyy kirjata miten vaatimuksen täyttymättömyys käy ilmi. Kirjataan myös mikäli jotain vaatimusta ei voida havaita tai testata.

Testitapaukset kirjoitetaan siten, että niiden avulla voi yksiselitteisesti määrittellä täyttääkö järjestelmä annetun vaatimuksen vai ei, eli voiko määritelty operaatio tehdä järjestelmällä vai ei. Testitapauksessa käsitellään yleensä vain yksi tapa suorittaa operaatio. Jos saman operaation voi tehdä useammalla tavalla, on vaihtoehtoiset tavat testattu ainoastaan, mikäli niistä on testausdokumentissa maininta.

5.1 Järjestelmätestauksen tavoite

Järjestelmätestauksen tavoite on selvittää:

- Tekeekö järjestelmä ne toiminnot, jotka on vaadittu?
- Onko järjestelmä helppokäyttöinen?
- Vastaako käyttöohjeistus järjestelmää?
- Toimiiko järjestelmä kuormitettuna?
- Toimiiko järjestelmä suunnitellussa ympäristössä?
- Pysyykö järjestelmä pystyssä ja onko se vikasietoinen?
- Onnistuuko asennus ohjeiden mukaan?

5.2 Testattavat kohdat

Seuraavassa testitapaukset johdettu vaatimusdokumentin käyttötapauksista ja numeroitu vastaavasti, Esim. käyttötapaus "T1 "Pesätarkastus"-tietojen syöttö järjestelmään" on pohjana testitapaukselle "KT1 "Pesätarkastus"-tietojen syöttö järjestelmään".

KT1 "Pesätarkastus"-tietojen syöttö järjestelmään

Syöte	Toivottu tulos
ks. kohta KT1.syöte	ks. kohta KT1.tulos

KT1.syöte: Koska Pesätarkastus-lomakkeessa on yli 100 syötekenttää, joista osa pakollisia ja joista taas joidenkin sallittu arvo riippuu toisen kentän arvosta, on kaikki tapaukset kattavaa syötejoukkoa hankala määrittellä. Testaus toteutetaan testaamalla:

1. hyväksytyillä syötteillä
2. hyväksytyillä syötteillä, mutta jättämällä pois pakollisia kenttiä

3. syötteillä, jotka ovat muuten hyväksytyillä mutta joissa rikotaan kenttien välisiä riippuvuusvaatimuksia
4. vaillinaisilla ja virheellisillä syötteillä

Testausraportista on ilmettävä käytetyt syötteet ja perustelut niiden valitsemiselle.

Huom. Koska tietokantaan tallennettavat tiedot voivat vaikuttaa todellisiin tietoihin (esim. reviirien kuntaan kuulumiseen ja vastaaviin) pitää olla varovainen syötteitä antaessaan, vaikka tämä tarkoittaisikin joiden syötekombinaatioiden pois jättämistä.

KT1.tulos:

Toivottu tulos:

1. oikeelliset tiedot tallentuvat tietokantaan
2. virheellisistä syötetiedoista tulee oikeaa ja täsmällistä informaatiota sisältävä virheilmoitus

Virheellinen (ei-toivottu) tulos:

1. oikeelliset tiedot eivät tallennu tietokantaan
2. virheelliset tiedot tallentuvat tietokantaan
3. virheelliset tiedot eivät tallennu tietokantaan, mutta virhesyötteistä ei tule ilmoitusta tai ilmoitus on väärä

KT2 "Uusi pesä" -tietojen syöttö järjestelmään

Testauksen toteutus ja tulosten arviointi kuten testaustapauksessa KT1.

KT3.1 Kuvan ja siihen liittyvien tietojen syöttö järjestelmään

SYÖTE:	TOIVOTTU TULOS:
Kelvolliset syötteet joka kohtaan	Kuva ja tiedot tallentuvat tietokantaan
Puuttuva pakollinen tieto (jokainen erikseen)	Puuttuva kenttä muuttuu punaiseksi, ei tallennusta
Virheellisiä syötetietoja (pesä id, kuvaaja id, pvm, kuvatyypin väärä; ei-numeerisia- ja erikoismerkkejä)	Virheilmoitus, ei tallennusta

KT3.2 Digikuvien haku

SYÖTE (reviiri_ID):	TOIVOTTU TULOS:
Olemassa oleva reviirinumero, reviiriin liittyy kuvia	Ilmoitus tallennuksesta levyille.
Olemassa oleva reviirinumero, reviiriin ei liity kuvia	Ilmoitus siitä, ettei reviirillä ole kuvia.
Olemassa olematon reviirinumero	Ilmoitus olemattomasta reviiristä (Nyt: Siirtää Haku-sivulle reviiri ID valmiiksi täytettynä reviiritietojen hakua varten)
Tyhjä kenttä	Tulostaa kaikki kuvat ja kuvatiedot hakemistoon. Onnistumisilmoitus.
Ei-numeerisia merkkejä (kirjain- ja erikoismerkkejä)	Ilmoitus olemattomasta reviiristä.

KT4 Raporttien tuottaminen

SYÖTE	TOIVOTTU TULOS
Raporttisivulla, ei raporttia valittuna ja kentät tyhjänä	virheilmoitus
Raporttisivulla ei raporttia valittuna, vuosikentät täytetty	virheilmoitus
Raporttisivulla useita raportteja valittuna, vuosikentät täytetty	virheilmoitus

R-raportti

SYÖTE	TOIVOTTU TULOS
raportti R valittu(kuten kaikissa seuraavissakin kohdissa), kentät tyhjänä	virheilmoitus
suuralue valittu: A, muut kentät tyhjänä	HaliOutput kansioon tulostuu valitun suuralueen pesien esitäytetyt pdf-lomakkeet vuodelle 2007
ymp.keskus valittu: LA, muut kentät tyhjänä	HaliOutput kansioon tulostuu valitun ympäristökeskuksen pesien esitäytetyt pdf-lomakkeet vuodelle 2007
kunta valittu: HAMMAR, muut kentät tyhjänä	HaliOutput kansioon tulostuu valitun kunnan pesien esitäytetyt pdf-lomakkeet vuodelle 2007
reviiri valittu: Finbo, muut kentät tyhjänä	HaliOutput kansioon tulostuu valitun reviirin pesien esitäytetyt pdf-lomakkeet vuodelle 2007
pesän nimi kirjoitettu virheellisesti, muut kentät tyhjinä	virheilmoitus
pesän nimi kirjoitettu oikein: Puosletti, muut kentät tyhjinä	HaliOutput kansioon tulostuu valitun pesän esitäytetty pdf-lomake vuodelle 2007
kentät täytetty satunnaisilla arvoilla siten etteivät osu yhteen: Suuralue A, Ymp.keskus ES, Kunta KUUSAM, reviiri Kaurissalo, Pesän nimi Puotletti	virheilmoitus

Raportit T, U, V, P sekä S raportti siten että vuosiväli on vaihdettu yhteen vuoteen

Syöte	Toivottu tulos
raportti valittu, kentät tyhjänä	virheilmoitus
raportti valittu, täytetty alkuvuosi: 2007	virheilmoitus
raportti valittu, täytetty loppuvuosi: 2007	virheilmoitus
raportti valittu, täytetty vuodet väärässä järjestyksessä alkuvuosi: 2006, loppuvuosi 2007	virheilmoitus
raportti valittu, täytetty vuodet joilta ei ole tietoja 1922-1925	virheilmoitus
raportti valittu, täytetty alkuvuosi 2007, loppuvuosi 2007	tiedostoon tulostuu vaatimusmäärittelyä vastaava raportti
raportti valittu, vuosiväli 1999-2004, suuralue valittu: A, muut kentät tyhjänä	tulostuu raportti vuosivälin ja suuralueen A osalta
raportti valittu, vuosiväli 1999-2004, ymp.keskus valittu: LA, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun ymp.keskuksen osalta.
raportti valittu, vuosiväli 1999-2004, kunta valittu: HAMMAR, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun kunnan osalta.
raportti valittu, vuosiväli 1999-2004, suuralue ja kunta valittu niin etteivät vastaa toisiaan; kunta: HOUTSK, suuralue: A , muut kentät tyhjänä	tulostuu raportti vuosivälin ja kunnan osalta.
raportti valittu, vuosiväli 1999-2004, reviiiri valittu: Finbo, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun reviiirin osalta
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu virheellisesti, muut kentät tyhjinä	virheilmoitus
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu oikein: Puosletti, muut kentät tyhjinä	tulostuu raportti vuosivälin ja valitun pesän osalta
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu oikein: Puosletti, muut kentät täytetty satunnaisilla arvoilla	tulostuu raportti vuosivälin ja valitun pesän osalta
raportti valittu, vuosiväli 1999-2004, kentät täytetty satunnaisilla arvoilla siten etteivät osu yhteen: Suuralue A, Ymp.keskus ES, Kunta KUUSAM, reviiiri Kaurissalo, Pesän nimi Puosletti	tulostuu raportti vuosivälin ja valitun pesän osalta

KT5 Pesätietojen muokkaus

Testataan kuten KT1, hyväksytyillä ja vaillinnaisilla syötteillä, testaus keskittyy vain Merikotkien lisäämiin attribuutteihin, joita ovat: Naturassa, Relaskooppimittausten pvm, Työpanos attribuutit, Poikasten mittaus: mittauksen pvm, mittaaja, toisen mittauksen arvot, munan pituus.

5.3 Järjestelmätestauksen suoritus

Järjestelmätestauksen suorittajana tulisi olla järjestelmän kehittämiseen osallistumaton taho. Käytännössä ohjelmistotuotantoprojektissa tähän ei ole resursseja, vaan järjestelmätestauksen testitapaukset ja kirjaaminen suoritetaan ryhmäläisten toimesta.

6 Hyväksymistestaus

Järjestelmäkokonaisuus on tarkoitus testauttaa asiakkaan käyttäjäkunnan edustajien toimesta ennen järjestelmän luovutusta. Käyttöliittymän kautta tapahtuva asiakkaan hyväksymistestaus on käytännössä järjestelmätestausta järjestelmän tyypillisimmät käyttötapaukset tietävän käyttäjän toimesta.

Etenkin monia (>100) erilaisia valintavaihtoehtoja käsittävien Uusi pesä, Pesätieto ja Pesätietojen muuttaminen testauksessa on olennaisen tärkeää, että järjestelmän vakituinen käyttäjä, joka tuntee lomakkeiden eri kombinaatioiden todellisuudessa toteutuvat tapaukset, koekäyttää järjestelmää ennen toteutuksen lopullista lukkoon lyömistä.

Asiakkaan suorittama järjestelmä/hyväksymistestaus tulee tapahtumaan lähellä projektin päättymistä, joten vaihtoehtoina löydettyjen puutteiden käsittelyyn on pikainen korjaus, mikäli mahdollista ja varmuus pikakorjauksen luotettavuuteen on riittävä, sekä löydetyn puutteen kirjaus dokumentaatioon jatkokehitystä ja käyttäjän valintoja varten.

Järjestelmä ja käyttöliittymän testauksen voidaan katsoa olevan valmis kun ilmenneet virheet on joko korjattu tai niiden korjaamatta jättäminen ja tarvittava dokumentointi on sovittu asiakkaan kanssa.

7 Testausaikataulu

Projektisuunnitelmassa testaukseen on varattu aikaa viikon 42 alusta viikon 49 loppuun. Testausta tehdään lomittain toteutuksen kanssa. Testaus on aloitettu laatimalla testausuunnitelma.

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.

	43	44	45	46	47	48	49	50	viikko
Testaussuunnitelma	x	xx	xx	xx					
Yksikkötestaus		x	xx	xx	xx	x			
Integrointitestaus				xx	xx	xx	xx		
Järjestelmätestaus						x	xx	x	
Hyväksymistestaus							xx		

7.1 Määräajat

Projektin aikataulun mukaan testausaikataululle on määritelty seuraavat määräajat:

- Viikko 42 toteutus ja yksikkötestaus alkaa
- Viikko 45 integrointi- ja järjestelmätestaus alkaa
- Viikko 48 toteutus ja yksikkötestaus tehty
- Viikko 50 integrointi- ja järjestelmätestaus tehty
- Viikko 49 hyväksymistestaus

7.2 Testivaiheiden tekijät

Toteutetun komponentin ja sen luokkien yksikkötestaaminen on kyseisen komponentin tekijän vastuulla. Järjestelmätestaamisessa on osallisena koko projektiryhmä. Hyväksymistestauksen suorittaa asiakas valmiille järjestelmälle.

7.3 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 8 (Testausraportit).

8 Testausraportit

8.1 Yksikkö- ja integrointitestaus

Yksikkö- ja integrointitestauksen erottaminen toisistaan täysin oli tämän projektin luonteen vuoksi pääsääntöisesti turhaa, koska projektin työn kohteena oli valmis järjestelmä, jossa lisättiin toiminnallisuuksia (luokkia, metodeja, templateja) jo olemassa olevan järjestelmän runkoon. Useimmissa tapauksissa tynkien teko olisi ollut käytännössä hyödytöntä, koska järjestelmän tarjoamat syötteet uusille yksiköille olivat jo sangen monilta osin ennalta rajoitut erilaisten jo olemassa olevien validointirutiinien toimesta. Tällöin käytännöllisin tapa testata yksikkö oli käydä koodi läpi kunkin yksikön toteuttajan oman harkinnan mukaan.

Usein yksikkötestauksen osuus oli sangen triviaali, huomion keskittyessä yksikön sovittamiseen muuhun järjestelmään eli integrointitestaukseen. Koska rajanveto näiden testausvaiheiden välillä oli pitkälti integrointitestaukseen painottuva, on testaustulosten osalta yhdistetty yksikkö- ja integrointitestaus saman otsikon alle. Testausraporttien sisällöstä pääsääntöisesti selviää testitoimien jakauma yksikkö- ja integrointitestauksen osalta.

Yleisesti voi todeta, että yksiköitä on sekä järjestelmä- että integrointitestattu erittäin kattavasti tekijöidensä toimesta, ja oheiset raportit ilmentävät vain kehitystyön lopputuloksia useiden testausiteraatioiden jäljiltä.

Yksikkötestauksen hyväksymisrajaksi määritetyn 90% lausekattavuuden saavuttaminen ei työn jatkoprojektiluonteen vuoksi noussut kynnyksikysymykseksi missään vaiheessa, ja sen saavuttaminen oli rivimäärillä laskettuna usein helpoimmasta päästä kehitystyötä. Varsinaiset ongelmat kohdattiin uusien ja/tai muutettujen toiminnallisuuksien integroimisessa nykyiseen järjestelmään, ja integroinnin "esivaatimuksena" oletetun nykyisen järjestelmän toimintaperiaatteiden ja perusarkkitehtuurin ymmärtäminen. Tämän ymmärryksen saavuttaminen oli monessa kohdin suuri osa toteutusta, erityisesti dokumentaation vastaisten tai kokonaan dokumentoimattomien ominaisuuksien liittyessä työryhmän kehitystyöhön. Samalla se vei suuren osan projektin aikaresursseista, koska järjestelmä on jo nykyisellään kohtalaisen kokoinen ja siellä on paljon dokumentaation vastaisia tai ulkopuolisia ominaisuuksia.

Alla yksiköiden toteuttajien kuvauksia toteuttamiensa toiminnallisuuksien testauksesta ja tarpeen vaatiessa erittely yksikkö- ja integrointitestauksen osuudesta. Usein molemmat kulkivat perustellusti ja hyväksymisvaatimukset täyttäen käsi kädessä.

Raportti P

Testattava kohde	reportP.java	Testauksen ajankohta	10.12.2007
Testin kuvaus	SQL-lause on käyty läpi staattisesti ja testattu suorakäytöllä tietokantaan. Yhteenvetolaskuja on testattu tynkinä, arvoilla 0 ja 10000 sekä yksittäisillä arvoilla siltä väliltä, yhteenvetolaskut eivät voi saada parametreikseen alle nollan tai null arvoja johtuen P-raportin attribuuttien alustuksista eikä tietokantahauista tule neg.tuloksia, mahdollinen nollan päätyminen jakajaksi on otettu huomioon. Csv-tiedostoa testattiin importtaamalla se exceliin.		
Testin tulos	P-raportin tulokset vastasivat tietokannan sisältöä, rauhoitustilanne otetaan valittua vuotta edeltävän vuoden tiedoista. Summat ja prosenttiosuuskalkulat toimivat, prosentteja ei pyöristetä vaan ne katkaistaan. Tulostettu tiedosto näyttää excelissä juuri siltä miltä pitääkin.		
Havaittujen virheiden kuvaus	Raportti ottaa rauhoitustilanteen edellisen vuoden tiedoista: TRUNC(TO_D VUOSI ALKU_PVM LOP- PU_PVM PESA_ID P -----		

	01.01.2006 2006 14.05.2005 01.01.2020 86 L 01.01.2006 2006 24.06.2006 01.01.2020 752 R 01.01.2006 2006 07.06.2004 05.06.2005 131 R 01.01.2006 2006 11.06.2007 01.01.2020 131 R 01.01.2006 2006 29.09.2004 13.05.2005 86 L 01.01.2006 2006 18.06.2006 10.06.2007 131 R 01.01.2006 2006 06.06.2005 17.06.2006 131 R Näistä tuloksista P-raportti laskee vuodelle 2006 mukaan vain pesän 86 ja 131, koska pesän 752 rahoitustilanne on muuttunut vasta 24.6.2006. Tämä on ehkä toivottukin toiminnallisuus?		
Testaaja	Tuire Huhtamäki		

Raportti R

Testattava kohde	RaportR.java	Testauksen ajankohta	19.11.2007
Testin kuvaus	1) Yksikkötestauksessa käyty läpi pdf-tulostukseen liittyvä vähäinen uusi koodi. 2) Testattu uutta PDF pohjaa vanhalla koodilla valitsemalla Raportit-ikkunassa R-raportti testaussuunnitelman määrittelemillä syötteillä (ks. Testaussuunnitelma, luku 5).		
Testin tulos	Uusi PDF pohja toimii yhteen vanhan koodin kanssa, raportti tulostuu vaatimusten mukaisesti.		
Havaittujen virheiden kuvaus	Ei havaittuja virheitä.		
Testaaja	Janne Piippo		

Raportti S

Testattava kohde	reportS.java	Testauksen ajankohta	13.12.2007
Testin kuvaus	SQL-lause on käyty läpi staattisesti ja testattu suorakäytöllä tietokantaan. Yhteenvetolaskuja on testattu tynkinä, arvoilla 0 ja 10000 sekä yksittäisillä arvoilla siltä väliltä. Yhteenvetolaskut eivät voi saada parametreikseen nollaa, alle nollan tai null arvoja, koska yhteenvetolaskuihin päätyy vain suuralueet joilta on löytynyt pesiä, näin nolla ei päädy myöskään jakajaksi. Csv-tiedostoa testattiin importtaamalla se exceliin.		
Testin tulos	S-raportin tulokset vastasivat tietokannan sisältöä. Summat ja prosenttiosuuslaskut toimivat, prosentteja ei pyöristetä vaan ne katkaistaan. Tulostettu tiedosto näyttää excelissä juuri siltä miltä pitääkin.		
Havaittujen virheiden kuvaus	Ei havaittuja virheitä.		
Testaaja	Tuire Huhtamäki		

Raportti T ja RaporttiV

Testattava kohde	ReportT.java, ReportV.java (Luokkien samankaltaisuuden vuoksi testaus oli molemmissa luokissa käytännössä identtistä.)	Testauksen ajankohta	6.12.2007
Testin kuvaus	Jokainen SQL-lause testattiin omana yksikkönään ensin suoraan tietokantaan. Molemmat luokat yksikkötestattiin erillään muusta järjestelmästä ja toisistaan tietokantahaku- ja raportintuotto-ominaisuuksien suhteen, syöteinä testaussuunnitelman mukaiset arvoalueanalyysiin pohjautuvat syötteet. Yhdistetyt SQL-lauseet toimivat sisäkkäin ajettuna oikein kaikilla testisyötteillä. Kattavaa testauksesta ei kyennyt tekemään. Raporttien vaatimuksena ollutta csv-ominaisuutta erottimella testattiin siirtämällä raportit Exceliin.		
Testin tulos	Luokka ReportT.java toteuttaa RaporttiT:n asiakkaan vaatimalla tavalla.		

	Luokka ReportV.java toteuttaa RaporttiV:n asiakkaan vaatimalla tavalla.		
Havaittujen virheiden kuvaus	Ei havaittuja virheitä.		
Testaaja	Petri Setälä		

Raportti U

Testattava kohde	reportU.java	Testauksen ajankohta	10.12.2007
Testin kuvaus	SQL kysely on käyty läpi staattisesti ja tuloksia on verrattu tietokantaan. Csv- tiedostoa arvioitiin importtaamalla se MS exceliin.		
Testin tulos	SQL-kysely antaa oikeanlaisia tuloksia. Csv-tiedosto näyttää excelissä siltä miltä pitääkin.		
Havaittujen virheiden kuvaus	Ei havaittuja virheitä.		
Testaaja	Tuire Huhtamäki		

KuvaOperation luokka, insertPicture() metodi

Testattava kohde	KuvaOperation luokka, insertPicture() metodi	Testauksen ajankohta	9.12.2007
Testin kuvaus	Kuvan ja siihen liittyvän tiedon lisäys tietokantaan. Syöte sisään tulee datamallista, joka generoitiin kuvan lisäyssivulta. Operaation tuloksena samojen arvojen tulisi löytyä tietokannasta. Metodista löytyy koodia, joka tarkistaa ettei osakentistä voi olla nulleja. Tämän testaus ei kuitenkaan ole oleellista, sillä se on ValidateKuva -luokan tehtävä.		
Testin tulos	Tiedot lisätään tietokantaan onnistuneesti. Tietoja ei lisätä, mikäli ValidateKuva -luokka palauttaa datamallille virheen. Tämä testattiin yrittämällä tallentaa tietokantaan tyhjää lomaketta. Näin ollen oleellinen toiminnallisuus on tullut testattua.		
Havaittujen virheiden kuvaus	Ei havaittuja virheitä.		
Testaaja	Juha Hiekkämäki		

ValidateKuva

Testattava kohde	ValidateKuva	Testauksen ajankohta	9.12.2007
Testin kuvaus	<p>ValidateKuva tarkistaa, että saadusta datamallista löytyy sekä kuvadata että tarpeelliset kentät oikein muotoiltuna. Pakollisia kenttiä ovat kuvadata, pesä id, kuvaaja id sekä päivämäärä. Lisäksi datan oikeellisuus (mikäli sitä löytyy) tarkistetaan kentistä aikuisten määrä, rohkeus sekä renkaiden id:t. Kentät ovat toisistaan riippumattomia, joten haaraumakattavuudesta ei tarvitse välittää.</p> <p>Testissä tarkistettiin, että validoinnista menee läpi vain sellaiset kuvat joiden pesä id sekä kuva id löytyy tietokannasta. Pakolliset kentät tarkistettiin yrittämällä lähettää tyhjää kuvalomake, tällöin pakolliset kentät värikoodataan virheellisiksi.</p> <p>Päivämäärästä tarkistettiin, että päivämäärän päivän tulee olla väliltä 1-31 ja kuukauden väliltä 1-12.</p> <p>Uusi mallipohja ei salli syöttää virheellistä dataa muihin kenttiin, joten tätä toiminnallisuutta ei voi mallipohja muuttamatta tarkistaa.</p>		
Testin tulos	Luokka toimii yllä kuvatulla tavalla.		
Havaittujen virheiden kuvaus	Ei havaittuja virheitä.		
Testaaja	Juha Hiekkamäki		

ValidatePesätiedot.coordinatesInMunicipality()

Testattava kohde	ValidatePesätiedot.coordinatesInMunicipality()	Testauksen ajankohta	pvm 20.11.2007
Testin kuvaus	<p>Metodia on testattu irrallisena järjestelmästä erilaisilla koordinaattiarvoilla, arvioitu tuloksia maalaisjärjellä ja verrattu yhteen verkosta löytyvään etäisyyslaskuriin (http://jan.ucc.nau.edu/~cvm/latlongdist.html).</p>		
Testin tulos	<p>Suurin testattu etäisyys Inari-Helsinki ja pienin Hakaniemen tori - Rautatientori. Testattiin myös nolla- ja negatiivisilla arvoilla sekä tilannetta jossa lähtö- ja päätepiste olivat samat. Tulokset vaikuttivat järkeviltä (esim: In-HKI 1089km, Htori-Rtori 0,535km) ja vastasivat läheisesti netistä löydettyä laskuria.</p>		

Havaittujen virheiden kuvaus	Metodissa käytetty kaava saattaa aiheuttaa ongelmia jos yrittää laskea täsmälleen vastakkaisilla puolilla maapalloa, saman isoympyrän säteellä olevien kahden pisteen välistä etäisyyttä.		
Testaaja	Tuire Huhtamäki		

Pesätarkastus

Testattava kohde	Pesätarkastus	Testauksen ajankohta	10.12.2007
Testin kuvaus	Testataan tarkastuksen lisäystä: <ol style="list-style-type: none"> 1. hyväksytyillä syötteillä 2. hyväksytyillä syötteillä, mutta jättämällä pois pakollisia kenttiä 3. syötteillä, jotka ovat muuten hyväksytyillä mutta joissa rikotaan kenttien välisiä riippuvuusvaatimuksia 4. vaillinaisilla ja virheellisillä syötteillä 		
Testin tulos	<ol style="list-style-type: none"> 1. OK 2. OK, leiska osaa kertoa mitkä tiedot puuttuvat 3. OK (joitain virheitä, alla), joskin tämä testi ei ole testaajan ammattitaidosta johtuen kovinkaan kattava. Lomake näyttäisi tosin osaavan ilmoittaa hyvin monesta riippuvuusongelmista (lähinnä pesimistuloksen eri vaihtoehdot). 4. OK (joitain virheitä, alla). Numerokenttiin johon voidaan lisätä omia syötteitä (eli ei pudotusvalikot) osaavat huomata kirjaimia ja negatiivisia lukuja. Kommentti/muut vapaamuotoiset kentät eivät luonnollisesti aiheuta mitään ongelmia syötteiden suhteen (ei tarkisteta). 		
Havaittujen virheiden kuvaus	<ol style="list-style-type: none"> 2. Lomake ei ilmoita virheestä jos lisätään poikasten tietoja vaikka poikasten lukumäärää ei ole ilmoitettu. Sama aikuisten kanssa. Oletan tämän olevan vanha toiminnallisuus joka sinänsä ei ole kriittinen virhe. 4. Numerokentät joihin tulee double-arvoja eivät osaa havaita negatiivisia 		

	<p>lukuja. Virheentarkistuksessa käytetään eri metodia int- ja doublemuuttujille, joten double-versiossa on luultavasti jokin virhe.</p> <p>Huom: tarkastuksen vuoden muuttaminen (käymättä 'Uusi tarkastus'-napin kautta) tarkastuksen lisäyksen jälkeen aiheuttaa suuria ongelmia, joten tätä kenttää ei voi tästä lähtien muokata (muuten kuin uuden tarkastuksen kautta).</p>		
Testaaja	Teemu Pulkkinen		

Pesätiedot

Testattava kohde	Pesätiedot	Testauksen ajankohta	10.12.2007
Testin kuvaus	<p>Annetaan virheellisiä syötteitä, ja tarkistetaan että järjestelmä osaa</p> <p>1) huomata että kenttä on virheellisesti syötetty</p> <p>2) huomauttaa käyttäjälle mikä syöte on väärä. Tekstikentissä värjäämällä kenttä punaiseksi, muissa tapauksissa vähintään tekstimuotoisella virhesanomalla sivun ylä laidassa.</p> <p>Testissä on keskitytty lähinnä uusien kenttien tarkistukseen.</p> <p>Riippuvuusongelmia ei ole (kattavasti) testattu niiden määrän ja asiantuntemusvaatimusten takia.</p> <p>Lomakkeiden kentät ovat jo itse käyttöliittymässä määritelty tietyn pituisiksi, joten liian suuria arvoja ei voida edes antaa. Tarkistuksessa on tämän takia käytetty tekstiä ja negatiivisia arvoja.</p>		
Testin tulos	<p>Numerokentät osaavat havaita virheet (tekstiä, negatiiviset numerot).</p> <p>Virheelliset kentät värjäntyvät punaisiksi, ja tietoja ei lähetetä tietokantaan. Ongelmat pudotusvalikoissa (esim. riippuvuusongelmat) huomataan sivun ylä laidassa olevasta tekstiselitteestä.</p>		
Havaittujen	Järjestelmä ei osaa hoitaa tekstisyötteitä päi-		

virheiden kuvaus	<p>vämääräkentissä. Oletettavasti tämä on kuitenkin harvinainen virhe.</p> <p>Järjestelmä ei osaa huomata tekstisyötettä koordinaattikentissä ellei virheellinen koordinaattikenttä ole valittuna (radionappi kentän yhteydessä).</p> <p>Aikuisille voi antaa rengastietoja vaikka "aikuisien lukumäärä"-kohdassa olisi annettu tyhjä tai 0-arvo.</p> <p>Nämä ongelmat ovat luultavasti jo vanhassa järjestelmässä.</p> <p>Tekstikenttiin ei kohdistu tarkistuksia, joten ne eivät osaa havaita "virheellisiä" kenttiä (esim. 'Pesän nimi' voi olla negatiivinen numero).</p>		
Testaaja	Teemu Pulkinen		

Uusi pesä

Testattava kohde	Uusi pesä	Testauksen ajankohta	10.12.2007
Testin kuvaus	<p>Testi alkaa yrittämällä lähettää tyhjä lomake. Tämän jälkeen järjestelmän löytämät puutteet ja virheet korjataan yksi kerrallaan kunnes lomake menee läpi. Näin tiedetään että ainakin minimitiedoilla saadaan 'uusi pesä' kirjattua. Lukuisten riippuvuuksien (ja puuttuvan asiantuntemuksen) takia täydellinen testaus ei ole tässä ajassa mahdollista. Kenttien yksittäiset testaukset (virheelliset syötet) tehdään pesän päivityslomakkeella testauksen helpottamiseksi, sillä onnistunut pesän lisäys vaikeuttaa seuraavia testejä (ei saa käyttää samaa reviiriä jne.)</p>		
Testin tulos	<p>Uuden pesän lisäys onnistui minimitiedoilla. Pesä löytyy kannasta kaikilla annetuilla tiedoilla, ja sen tietoja voidaan muokata 'pesätiedot'-näköymästä.</p>		
Havaittujen virheiden kuvaus	<p>Samoja virheitä kun tietojen päivityksessä (kenttiä).</p> <p>Lomake ei varoita että reviirissä on jo kyseiselle vuodelle kirjattu pesä ennenkuin pesää yrittää lisätä. Tämä ei aiheuta</p>		

	varsinaista ongelmaa (kuten virheellisten tietojen lisäystä), mutta voi olla rasittavaa saada tietää tämä vasta kun pesän kaikki tiedot on jo annettu.		
Testaaja	Teemu Pulkkinen		

Kuvahaku

(Tämä on klassikko, siksi en editoinut. Arvostan näkökulman rehellisyyttä ja paljastavuutta. Eihän testausta ole peruskursseilla missään opetettu. Tämän testiraportin kirjoittaja teki projektissa erittäin laadukasta työtä. T: Testivastaava Petri Setälä)

"Testauksesta:

Sori, yksikkötestaukseen ei riitä rahkeet millään. Järjestelmätestausta kai tein kuvahaulle: Testasin nollalla, -1:llä, -9999999:llä, 1:llä, 374:llä (maksimi reviirien määrä ilmeisesti), 9999999:llä eikä tule virhettä niillä ja toiminta on odotettua.

Kun aluksi aloitin testauksen, niin noita 374:ää suuremmat luvut aiheutti vääränlaisen virheilmoituksen, mutta nyt tulee oikeaoppisesti "reviiriä ei löydy".

En nyt taas muista / ymmärrä pitääkö tuosta joku testausrapotti jonnekin pistää tms. vai häh. Tiedän, että koodiin jäi virhe joka tuottaisi "valkoisen sivun", mutta nyt leiskan kautta ei voi syöttää 9999999:ä suurempaa lukua siihen, joten se virhe ei esiinny. Ja jos nyt testauksesta mitään olen ymmärtänyt, niin mun olisi pitänyt tehdä se perl-scripti tai joku muu härpäke, jolla ne arvot voi syöttää tonne hakuun aina samalla tavalla (eikä siis mitään ohjetta, että "näppäile 99 siihen yhteen kenttän ja klikkaa hae")... noh, onneksi tää ei ole mitään oikea projekti "

8.2 Järjestelmätestaus

Järjestelmätestaus on ollut ryhmän jäsenten tekemää, mahdollisimman yleisiin käyttötapauksiin perustuvaa valmiin järjestelmän testausta käytännössä käyttöliittymäsyötteiden avulla. Syötteet on pyritty valitsemaan yleisen terveen järjen sekä tämän dokumentin suosittelien periaatteiden pohjalta. Kuitenkin esimerkiksi käyttöliittymänäkymien Uusi pesä, Pesätieto ja Pesätietojen muuttaminen kattava järjestelmätestaus on projektin puitteissa mahdotonta kyseisten näkymien sisältämien valikkomäärien vuoksi. Ryhmän jäsenet ovat toteuttaneet oman parhaan ymmärryksensä mukaiset testitapaukset, mutta paljon jää asiakkaan edustajien hyväksymistestauksen (luku 8.3) varaan ennen kuin koko järjestelmän toimivuutta voi riittävin perustein arvioida.

Raportit - yleistä

SYÖTE:	TOIVOTTU TULOS:	TULOS:
Raporttisivulla, ei raporttia valittuna ja kentät tyhjänä	virheilmoitus	Ei toimintoja. OK
Raporttisivulla ei raporttia valittuna, vuosikentät täytetty	virheilmoitus	Ei toimintoja. OK
Raporttisivulla useita raportteja valittuna, vuosikentät täytetty	virheilmoitus	Aakkosjärjestyksessä ensimmäinen raportti toteutetaan, muita ei, ei ilmoitusta. OK

Raportti P

SYÖTE:	TOIVOTTU TULOS:	TULOS:
Ei hakuehtoja	virheilmoitus	Vuosilukukentät muuttuvat punaiseksi. OK
Vuosilukuhaku 1900-2020 (hakuikkunan mahdollistamat rajat)	virheilmoitus	Raportti v. 1900-2020, jossa nollakentät niillä vuosilla, joilta ei tietoa saatavilla. HYVÄKSYTTÄVISSÄ
Vuosilukurajat + vain yksi muu hakuehto (kaikki kokeiltu)	raportti hakuehtojen mukaan, paitsi jos hakuehto on pesän nimi jota ei ole -> virheilmoitus (muut kentät pudotusvalikoita)	Raportti hakuehtojen mukaan, nollakentät niillä vuosilla, joilta ei tietoa saatavilla. (Olemaan pesä: pesäkenttä punaiseksi). HYVÄKSYTTÄVISSÄ
Vuosilukurajat + useampi hakuehto	Raportti. Hakuehdoista käytetään vain yksilöivintä, prioriteettijärjestys on seuraava: pesänimi, revierinimi, kunta, ympäristökeskus, suuralue.	Toivottu tulos, paitsi prioriteettijärjestyksessä kunta menee revierinimen edelle. OK
Virhesyötteitä vaapaavalintaisiin kenttiin (vuosiluvut ja pesän nimi)	virheilmoitus	kentät punaisiksi. OK

Raportti S

SYÖTE	TOIVOTTU TULOS	SAATU TULOS
raportti valittu, kentät tyhjänä	virheilmoitus	Toivottu tulos
raportti valittu, täytetty alkuvuosi: 2007	virheilmoitus	Toivottu tulos
raportti valittu, täytetty loppuvuosi: 2007	virheilmoitus	Toivottu tulos

raportti valittu, täytetty vuodet väärässä järjestyksessä alkuvuosi: 2007, loppuvuosi 2006	virheilmoitus	Toivottu tulos
raportti valittu, täytetty vuodet joilta ei ole tietoja 1922-1925	virheilmoitus	Hauilla ei tuloksia
raportti valittu, täytetty alkuvuosi 2007, loppuvuosi 2007	tiedostoon tulostuu vaatimusmäärittelyä vastaava raportti	Toivottu tulos
raportti valittu, vuosiväli 1999-2004, suuralue valittu: A, muut kentät tyhjänä	tulostuu raportti vuosivälin ja suuralueen A osalta	Toivottu tulos
raportti valittu, vuosiväli 1999-2004, ymp.keskus valittu: LA, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun ymp.keskuksen osalta.	Toivottu tulos
raportti valittu, vuosiväli 1999-2004, kunta valittu: HAMMAR, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun kunnan osalta.	Hauilla ei löytynyt tuloksia
raportti valittu, vuosiväli 1999-2004, suuralue ja kunta valittu niin etteivät vastaa toisiaan; kunta: HOUTSK, suuralue: A , muut kentät tyhjänä	tulostuu raportti vuosivälin ja kunnan osalta.	Toivottu tulos
raportti valittu, vuosiväli 1999-2004, reviiiri valittu: Finbo, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun reviiirin osalta	Toivottu tulos
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu virheellisesti, muut kentät tyhjinä	virheilmoitus	Toivottu tulos
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu oikein: Puosletti, muut kentät tyhjinä	tulostuu raportti vuosivälin ja valitun pesän osalta	Toivottu tulos
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu oikein: Puosletti, muut kentät täytetty satunnaisilla arvoilla	tulostuu raportti vuosivälin ja valitun pesän osalta	Toivottu tulos
raportti valittu, vuosiväli 1999-2004, kentät täytetty satunnaisilla arvoilla siten etteivät osu yhteen: Suuralue A, Ymp.keskus ES, Kunta KUUSAM, reviiiri Kaurissalo, Pesän nimi Puosletti	tulostuu raportti vuosivälin ja valitun pesän osalta	Toivottu tulos

Raportti T

Syöte	Toivottu tulos	Tulos
raportti T valittu, kentät tyhjänä	virheilmoitus	OK
raportti T valittu, täytetty alkuvuosi: 2007	virheilmoitus	OK
raportti T valittu, täytetty loppuvuosi: 2007	virheilmoitus	OK
raportti T valittu, täytetty vuodet väärässä järjestyksessä alkuvuosi: 2006, loppuvuosi 2007	virheilmoitus	OK
raportti T valittu, täytetty vuodet joilta ei ole tietoja 1922-1925	virheilmoitus	tulostaa raportin vuo-

		silta 1970-2007
raportti T valittu, täytetty alkuvuosi 2007, loppuvuosi 2007	tiedostoon tulostuu vaatimusmäärittelydokumentin liitteen 3 mukainen raportti T, csv-muotoisena vuodelta 2007	OK
raportti T valittu, vuosiväli 1999-2004, suuralue valittu: A, muut kentät tyhjänä	tulostuu raportti T vuosivälin ja suuralueen A osalta	OK
raportti T valittu, vuosiväli 1999-2004, ymp.keskus valittu: LA, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun ymp.keskuksen osalta.	OK
raportti T valittu, vuosiväli 1999-2004, kunta valittu: HAMMAR, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun kunnan osalta.	OK
raportti T valittu, vuosiväli 1999-2004, suuralue ja kunta valittu niin etteivät vastaa toisiaan; kunta: HOUTSK, suuralue: A, muut kentät tyhjänä	tulostuu raportti vuosivälin ja kunnan osalta.	ei löydy tuloksia lainkaan
raportti T valittu, vuosiväli 1999-2004, reviiiri valittu: Finbo, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun reviiirin osalta	OK
raportti T valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu virheellisesti, muut kentät tyhjinä	virheilmoitus	OK
raportti T valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu oikein: Puosletti, muut kentät tyhjinä	tulostuu raportti vuosivälin ja valitun pesän osalta	OK
raportti T valittu, vuosiväli 1999-2004, kentät täytetty satunnaisilla arvoilla siten etteivät osu yhteen: Suuralue A, Ymp.keskus ES, Kunta KUUSAM, reviiiri Kaurissalo, Pesän nimi Puosletti	tulostuu raportti vuosivälin ja valitun pesän osalta	ei löydy mitään tuloksia

Raportti U

SYÖTE:	TOIVOTTU TULOS:	TULOS:
raportti valittu, kentät tyhjänä	virheilmoitus	Vuosilukukentät muuttuvat punaiseksi. OK
raportti valittu, täytetty alkuvuosi: 2007	virheilmoitus	Loppuvuoden kenttä muuttuu punaiseksi. OK
raportti valittu, täytetty loppuvuosi: 2007	virheilmoitus	Alkuvuoden kenttä muuttuu punaiseksi. OK
raportti valittu, täytetty vuodet väärässä järjestyksessä alkuvuosi: 2007, loppuvuosi 2006	virheilmoitus	Loppuvuoden kenttä muuttuu punaiseksi. OK
raportti valittu, täytetty vuodet joilta ei ole tietoja 1922-1925	virheilmoitus	Ilmoitus: "Haulalla ei löytynyt tuloksia". (Tosin ohessa on vakioilmoitus raportin tulos-

		tuksesta kansioon, vaikka tulostusta ei tehdä). OK
raportti valittu, täytetty alkuvuosi 2007, loppuvuosi 2007	tiedostoon tulostuu vaatimusmäärittelyä vastaava raportti	OK.
raportti valittu, vuosiväli 1999-2004, suuralue valittu: A, muut kentät tyhjänä	tulostuu raportti vuosivälin ja suuralueen A osalta	OK.
raportti valittu, vuosiväli 1999-2004, ymp.keskus valittu: LA, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun ymp.keskuksen osalta.	OK.
raportti valittu, vuosiväli 1999-2004, kunta valittu: HAMMAR, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun kunnan osalta.	OK.
raportti valittu, vuosiväli 1999-2004, suuralue ja kunta valittu niin etteivät vastaa toisiaan; kunta: HOUTSK, suuralue: A , muut kentät tyhjänä	tulostuu raportti vuosivälin ja kunnan osalta.	OK.
raportti valittu, vuosiväli 1999-2004, reviiiri valittu: Finbo, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun reviiirin osalta	OK.
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu virheellisesti, muut kentät tyhjinä	virheilmoitus	Pesäkenttä punaisiksi. OK
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu oikein: Puosletti, muut kentät tyhjinä	tulostuu raportti vuosivälin ja valitun pesän osalta	OK.
raportti valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu oikein: Puosletti, muut kentät täytetty satunnaisilla arvoilla	tulostuu raportti vuosivälin ja valitun pesän osalta	OK.
raportti valittu, vuosiväli 1999-2004, kentät täytetty satunnaisilla arvoilla siten etteivät osu yhteen: Suuralue A, Ymp.keskus ES, Kunta KUUSAM, reviiiri Kaurissalo, Pesän nimi Puosletti	tulostuu raportti vuosivälin ja valitun pesän osalta	OK.
Raportti valittu, vuosiväli sallituissa rajoissa, pesän nimi ja kunnan nimi tyhjänä, reviiiri valittu, suuralueen ja ympäristökeskuksen valinnoilla ei merkitystä. (=Reviiri valittu muttei kuntaa, pesäkenttä tyhjä.)	Tulostuu raportti valitun vuosivälin ja reviiirin osalta.	Kaatuu: valkoinen sivu. VIRHE, mutta harvinainen poikkeustapaus. Informoidaan asiakasta.

Raportti V

Syöte	Toivottu tulos	Tulos
raportti V valittu, kentät tyhjänä	virheilmoitus	OK
raportti V valittu, täytetty alkuvuosi: 2007	virheilmoitus	OK
raportti V valittu, täytetty loppuvuosi: 2007	virheilmoitus	OK
raportti V valittu, täytetty vuodet väärässä järjestyksessä alkuvuosi: 2006, loppuvuosi 2007	virheilmoitus	OK
raportti V valittu, täytetty vuodet joilta ei ole tietoja 1922-1925	virheilmoitus	tulostaa V raportin 1970-2007
raportti V valittu, täytetty alkuvuosi 2007, loppuvuosi 2007	tiedostoon tulostuu vaatimusmäärittelydokumentin liitteen 5 mukainen raportti T, csv-muotoisena vuodelta 2007	OK
raportti V valittu, vuosiväli 1999-2004, suuralue valittu: A, muut kentät tyhjänä	tulostuu raportti T vuosivälin ja suuralueen A osalta	OK
raportti V valittu, vuosiväli 1999-2004, ymp.keskus valittu: LA, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun ymp.keskuksen osalta.	OK
raportti V valittu, vuosiväli 1999-2004, kunta valittu: HAMMAR, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun kunnan osalta.	OK
raportti V valittu, vuosiväli 1999-2004, suuralue ja kunta valittu niin etteivät vastaa toisiaan; kunta: HOUTSK, suuralue: A, muut kentät tyhjänä	tulostuu raportti vuosivälin ja kunnan osalta.	ei tuloksia
raportti V valittu, vuosiväli 1999-2004, reviiiri valittu: Finbo, muut kentät tyhjänä	tulostuu raportti vuosivälin ja valitun reviiirin osalta, pesald:llä 260 vuonna 2002: 2 poikasta	OK
raportti V valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu virheellisesti, muut kentät tyhjinä	virheilmoitus	OK
raportti V valittu, vuosiväli 1999-2004, pesän nimi kirjoitettu oikein: Puosletti, muut kentät tyhjinä	virheilmoitus; pesän mukaan ei voi hakea (vaatimus)	Tulostaa kaikki pesät valitulla vuosivälillä
raportti V valittu, vuosiväli 1999-2004, kentät täytetty satunnaisilla arvoilla siten etteivät osu yhteen: Suuralue A, Ymp.keskus ES, Kunta KUUSAM, reviiiri Kaurissalo, Pesän nimi Puosletti	tulostuu raportti vuosivälin ja valitun reviiirin osalta	Ei tuloksia

Kuvahaku

SYÖTE (reviiri_ID):	TOIVOTTU TULOS:	TULOS:
Tyhjä syötekenttä	Tulostaa kaikki kuvat tietoiheen, ilmoitus tulostuksesta.	OK
Olemassa oleva reviirinnumero, reviiriin liittyviä kuvia	Tulostaa reviirin kuvat tietoiheen, ilmoitus tulostuksesta.	OK.
Olemassa oleva reviirinnumero, reviiriin ei liity kuvia	virheilmoitus	Ilmoitus:" Reviiriin ei liity kuvia", haku siirtyy pestietohauun, reviirikenttä esitetyt kuvat hakuun syötetyllä ID-numerolla. OK.
Olemattoman reviirinreviirinnumero, (numerosyöte).	virheilmoitus	Ilmoitus:" Reviiriin ei liity kuvia", haku siirtyy pestietohauun, reviirikenttä esitetyt kuvat hakuun syötetyllä ID-numerolla. OK.
Satunnaisia ei-numeerisia merkkejä.	virheilmoitus	Haku siirtyy pestietohauun, reviirikenttä esitetyt kuvat hakuun syötetyllä ID-numerolla. OK

Lisää uusi kuva

SYÖTE:	TOIVOTTU TULOS:	TULOS:
Kelvolliset syötteet joka kohtaan	Kuva ja tiedot tallentuvat tietokantaan. Onnistumisilmoitus näytetään.	OK.
Puuttuva pakollinen tieto (jokainen erikseen)	Puuttuva kenttä muuttuu punaiseksi, ei tallennusta	OK.
Virheellinen pvm (31.11.2007), muut tiedot oikein.	Pvm-kenttä muuttuu punaiseksi, ei tallennusta	Ei ilmoitusta, ei tallennusta.
Muut testatut tapaukset, joissa pakollinen kenttä ollut virheellinen muutoin kuin edellisessä testitapauksessa.	Puuttuva kenttä muuttuu punaiseksi, ei tallennusta	OK.

Koordinaattitarkastusten testaus 13.12.07

Syöte	Toivottu tulos	Tulos
yhtenäiskoordinaatti joka on kunnan säteen sisällä: pesän kunta "KUMLIN", astekoordinaatit: pit: "3164024" lev: "6712122"	tiedot tallentuvat tietokantaan	OK
yhtenäiskoordinaatti joka ei ole kunnan säteen sisällä: pesän kunta "KUMLIN", astekoordinaatit: pit: "3170444" lev: "6781001"	tiedot eivät tallennu tietokantaan ja tooltip kertoo, että pesä ei ole kunnan sisällä	OK

yhtenäiskoordinaatti joka on sama kuin kunnan keskipiste: pesän kunta "KUMLIN", astekoordinaatit: pit: " 3155097" lev: " 6707738"	tiedot tallentuvat tietokantaan	OK
--	---------------------------------	----

8.3 Hyväksymistestaus

Hyväksymistestaus toteutettiin muutama päivä ennen projektin päättymistä ja järjestelmän luovutusta, joten isoihin muutoksiin ei ollut resursseja. Siitä huolimatta ryhmä osoitti erinomaista venymistä korjatessaan viimeisen vuorokauden aikana useita sinänsä pieniä, mutta järjestelmän käytettävyyden kannalta merkittäviä järjestelmäominaisuuksia. Formaalisen asiakaspalautteen puutteessa oheen liitetään (painovapaus ym. asiaan liittyvien säännösten sen sallien) viimeisen 30 tunnin aikana ennen työn luovutusta toteutunutta asiakasryhmäläiskirjeenvaihtoa. Tarkoituksena on osittaa asiakkaan perustyytyväisyys toteutettuun tuotteeseen sekä ryhmän joustavuus ja hyvä tahto viime hetken muutostöissä, riippumatta siitä, oliko havaittu järjestelmäpuutos aikanaan kirjattu vaatimusmäärittelyyn.

Kirjeenvaihdon perusteella järjestelmään on edelleen jäänyt muutamia pienemmän puoleisia puutteita, lisäksi käyttöliittymänäkymän ilmentymä eri selaimissa/selainversioissa/näytöissä vaikuttaa hieman huolestuttavan asiakasta. Toisaalta yleisellä tasolla ryhmän aikaansaannoksia kiitetään ja paneutumista tehtävänantoon arvostetaan.

Tämän kirjeenvaihdon myötä kiitos asiakkaillemme.

Järjestelmän hyväksymistestaukseen liittyvät asiakas-työryhmä sähköpostit :

-

Asiakaskommentit normaalilla fontilla, ryhmän vastauskommentit boldattuna ja italicilla.-

Hyvät,

Matti L-S ("museovirkailija") on tänään jonkin verran testailut käyttöliittymää ja jatkaa huomenna (to) ja saanen lisää kommentteja to noin puolen päivän tienoilla. Havainnot:

1. Aivan lopussa kohdassa "Näytteitä otettu" on jäänyt yksi rivi pois. Tällä rivillä piti olla Valokuva linnusta K/E; Kuvaaja;; Valokuva pesästä K/E; Kuvaaja:.

- LISÄTTY MOLEMMAT 13.12.2007. OK

("Nyt lomakkeessa on kentät: "Kuvia linnuista", "Kuvia pesästä" sekä molempien yhteydessä "Kuvaaja". Mitä nopeasti ehdin testaamaan niin näyttäisi toimivan (lisäys ja muutos). Lisäksi lomake huomaa jos kuvaajan kohdalle laitetaan id joka ei löydy tietokannasta (kuten kaikki muutkin tarkastaja-kentät tekevät). Testailkaa mielellänne myös näitä kenttiä jos vielä kokeillette.)

2. Sitten on kohta, joka ei ole (muistini mukaan) vaatimusmäärittelyssä. Tämä koskee vanhan (eli siis nyt tuotantokäytössä olevan) käyttöliittymän kenttiä osiossa: TIEDOT AIKUISISTA. Tämän osion alimmalla rivillä on:

Koiras tai naaras Vasemman jalan renkaan Tunnus Väri Oikean jalan renkaan Tunnus Väri Rengas nähty

Nyt haluttaisiin:

- a) Jos rivin yhteenkään kenttään ei ole talletettu mitään ja Rengas nähty = E, niin ei toimenpiteitä.
- b) Jos rivin yhteen tai useampaan kenttään on talletettu jotakin tai Rengas nähty = K, niin talletetaan tietokantaan tekemäänne "Tiedot aikuisista"-osan kommenttikenttään "Ei renkaita ? tai ? tms" rivin tiedot muodossa: Koiras tai naaras: Vas rengas: <tunnus kannasta>, vas väri:<tunnus kannasta>. Oik rengas: <tunnus kannasta>, oik väri: <tunnus kannasta>. Rengas nähty <tunnus kannasta>.

- TEHTY TIETOKANTA-AJO TOIVOMUSTEN MUKAISESTI 13.12.2007. OK

Nämä nyt on tähän mennessä huomattu.

terv
HeLo

Muuta:


1. Matti L-S on tänään ja eilen kokeillut tekemäänne käyttöliittymää ja on siihen kovin tyytyväinen. Hän huomasi sellaisen pikkuasian, että relaskooppitietojen kentät ovat kovin eri levyisiä. Tämä lienee ainoastaan kosmeettinen juttu, mutta jos se on aivan helppo korjata, niin tallettajan esteettinen katse lepäisi paremmin ao kohdassa.

2. Matti huomasi, että tarkastajan nimen muuttaminen ei onnistu. Mistähän tämä johtuu? Onko käyttöympäristöllä jotakin tekemistä asian kanssa?

KORJATTU 13.12.2007:

"Tämäkin on nyt korjattu. Hyvä että tuli esille, koska sama virhe esiintyi myös muissa päivityskohdissa (mm. aputaulujen attribuutteja ei voinut muokata)"

3. Olen keskustellut Totin ja Matin kanssa. Olemme tulleet siihen tulokseen, että meillä on valmiudet siirtyä uuteen käyttöliittymäänmuutamalla ehdolla:

a) Hali2: tietokanta ja ohjelmat talletetaan sellaiseen paikkaan, että tuohon järjestelmään voidaan palata, mikäli uusi järjestelmä jostain odottamattomasta syystä osoittaa jotakin sellaista virhetoimintaa, jota on pidettävä fataalina. (Tämän voinee hoitaa laitoksen ylläpito , mutta nyt tuotantokäytössä olevasta versiosta ei saa hävitä mitään.)

Tuo lienee toteutettavissa käytännöllisimmin niin, että siirtymäjän aikana (=kunnes päätätte uskalletteko siirtyä uuteen versioon) kaikki tiedot tallennetaan sekä uuteen että vanhaan järjestelmään (joka pysyy entisellään, myös tietokannan suhteen). Varmasti kevätseason

alkuun mennessä asia selkiää puoleen tai toiseen. Ylläpito on oikea osoite molempien järjestelmien ylläpitämiseen yhtäaikaan. Tietysti nykyjärjestelmään kuuluvista lisäattribuuteista on pidettävä varmuuskopioita, mutta nuo varmuuskopiot lienevät turvallisinta sorttia, eli paperiraportit kentältä.

b) Saamme selvityksen tässä epostissa olevista avoimista asioista ja mahdollisuuksistanne näiden korjaamiseen tavalla tai toisella.

c) Toivoisimme, että joku ryhmäläisistä voisi toimia yhteyshenkilönä mikäli tarvitsemme lähiaikoina (talvella...alkukevällä) vastauksen johonkin meitä askarruttavaan kysymykseen. Ymmärrän, että tämä ei kuulu kurssin piiriin, mutta tässä on kyse myös tuotannossa olevasta sovelluksesta. Mikäli vastaus mahdolliseen kysymykseen saattaisi edellyttää jotakin työpanosta, niin tällaisesta ei ole kyse - jos sellaista ilmenisi, niin neuvottelut käytäisiin erikseen ja eri pohjalta.

Se mitä todennäköisemmin on mahdollista, mutta siitä ei ole ryhmän sisäistä päätöstä. Lisäksi kysymys on pitkälti siitä, mikä järjestelmän ominaisuus kaipaa konsulttiota, sillä varsinkin projektin loppuvaiheessa toteutusroolit olivat jo pitkälle erikoistuneet.

d) Siirtyminen Hali3:een dokumentoidaan operaatioineen.

Odottelen ehdotuksia ja vastauksia.

Lopuksi: Tässä vaiheessa vaikuttaa siltä, että olette tehneet erinomaista työtä. Tässä epostissa esitetyt pikkuasiat eivät sitä muuta. Olemme vaikuttuneita. Kiitos!

HeLo

Jostain syystä pöytäkoneeni selaimen ilmestyy käyttöliittymässä lomakkeen ykkössivun reunoihin (oikealle ja alas) vierityspalkit ja lomakkeen ykkössivu menee jonkun verran (häiritsevästi kuitenkin) rikki. Fontin pienentäminen selaimessa vie vierityspalkit pois ja lomake paranee, mutta toisaalle lomakkeessa tulee sitten pikkuisen (häiritsevästi) pöperöä. Tämä lienee merkki siitä, että käyttöliittymä saattaa olla herkkä selaimen (tuleville) muutoksille? Osaatteko ottaa kantaa.

Ainoa, mitä vierityspalkki vierittää näkyviin, on relaskoopitaulukko. Poistuisikohan ongelma, jos sitä hieman kaventaisi?

Kannettavassa tietokoneessani kaikki näkyy hienosti niinkuin pitääkin.

Ihan vaan sivumennen todettuna saamistani aika monista eposteista (ilmeisesti olen saanut vain pikkuriikkisen osan ...): venymiskykyne on ihailtavaa ja arvostan sitä erittäin korkealle. Tulen kertomaan siitä Totille ja muulle WWF:n merikotkatyöryhmälle. Joudun olemaan hieman varovainen Totin kanssa, ettei hän järkyty liikaa ... joudun ehkä varaamaan punaviiniä (laseihin) Totin järkytystä helpottamaan... 8-)

kiitos

HeLo

Jep. Olin jo avannut uuden selaimen jne ja nyt kävin vielä tuolla yhdessä luokassa kokeilemassa ja rikki on. Liitän alle kuvan tilanteesta. Kuva näkyy ainakin Thunderbirdissä, mutta selostan vielä:

1. Huomattakoon, että Thunderbirdissä kaappaamani osa näkyy hyvin vaikka selaimessa menee hiukan pöperöksi.
2. Tämä on siis tilanteessa jossa Haku-sivulta on menty Pesätiedot-sivulle. Tällöin pesätiedot -näkyvässä on vierityspalkit ykkössivun reunoissa ja vain alapalkki vierittää. Pystypalkki ei vieritä kun ei mitään ole "piilossa". Minusta vaikuttaa siltä, että relaskooppitaulukko on kokonaisuutena niin leveä, että vierityspalkit tulevat jonkin automatiikan tekemänä näkyviin. Tästä seuraa se, että sivun leveys on vierityspalkin verran kapeampi. Tästä edelleen seuraa se, että mm tarkastajan rivillä otsikko Numero: ja sitten tarkastajan numero (luku) joutuvat eri riveille vaikka niiden pitäisi olla samalla rivillä. Muutakin hässäkkää seuraa, mutta kaikki vaikuttaa johtuvan siitä, että relaskooppitaulukko on hiukan liian leveä kokonaisuutena. Korostan, että tuolta näyttää, mutta enhän tietenkään tiedä mistä pieni puroutuminen johtuu.
3. Kannettavassa kompuuterissani kaikki siis näkyy hienosti.

HeLo

sama juttu (siis rikki) myös Uusi pesä -sivulla. Pöytäkoneessa mulla on Firefox 2.0.0.8 ja kannettavassa 2.0.0.11. Resoluutio: Jos tarkoitat näytön resoluutiota, niin se on 1280 x 1024. Jos tarkoitat selaimen text sizeä niin se on normal".

nyt mä lähden pois koneen äärestä ennenkuin viimeinen bussi jättää

hyvää yötä
HeLo