

Toteutusdokumentti

Kotkat-ryhmä

Helsinki 18.5.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Katja Astikainen
Manta Jääskeläinen
Riikka Kaven
Leena Laivaara
Säde Seppälä
Marja Silenti

Asiakas

Heikki Lokki

Johtoryhmä

Juha Taina
Turjo Tuohiniemi

Kotisivu

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

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	29.4.2004	Ensimmäinen luonnos
1.0	18.5.2004	Lopullinen versio

Sisältö

1	Johdanto	1
1.1	Erikoissanasto ja käytetyt lyhenteet	1
2	Toteutuksen rajoitteet	2
2.1	Noudatettavat standardit ja tarvittavat ohjelmat	2
3	Tiedostojen sijainti	2
3.1	Käynnistys URL	3
3.2	Tiedostot alkokrunni palvelimella:	3
4	Osajärjestelmien toteutuspoikkeamat	4
4.1	Pakkaus haliaetus.db	4
4.1.1	ConnectionPool	5
4.1.2	Table	5
4.1.3	OperationResults	5
4.1.4	SearchResults	5
4.1.5	DatabaseOperation	5
4.1.6	SearchOperation	5
4.1.7	InsertOperation	5
4.1.8	UpdateOperation	5
4.1.9	DeleteOperation	5
4.2	Pakkaus haliaetus.servlet	6
4.2.1	HaliaetusControllerServlet	6
4.3	Pakkaus haliaetus.helper	6
4.3.1	NestInspectionHelper	6
4.3.2	MunicipalityHelper	6
4.4	Pakkaus haliaetus.general	7
4.4.1	HaliProperties	7

4.5	Pakkaus haliaeetus.command	7
4.5.1	HaliaeetusGeneral	7
4.5.2	CommandDispatcher	7
4.5.3	CommandFactory	7
4.5.4	CommandInterface	7
4.5.5	LoginScreenCommand	7
4.5.6	MainScreenCommand	7
4.5.7	NestInspectionCommand	7
4.5.8	RepearNestInfoCommand	9
4.5.9	NestInformationCommand	9
4.5.10	TerritoryCommand	9
4.5.11	MunicipalityCommand	10
4.5.12	InspectorCommand	10
4.5.13	HelperTableCommand	10
4.5.14	ReportCommand	10
4.6	Pakkaus haliaeetus.log	10
4.6.1	LoggingRulesInitializer	10
4.6.2	HaliaeetusLogger	10
4.7	Muut tiedostot	10
4.7.1	Template-tiedostot	11
4.7.2	Properties-tiedostot	11
4.7.3	Tyylitiedosto	11
4.7.4	Staattiset html-tiedostot	11
5	Käyttöliittymä	11
5.1	Käyttäjän sisäänkirjautuminen	12
5.2	Haku	12
5.3	Uuden pesän lisääminen ja vanhan pesän tarkastus	12
5.4	Aputaulun päivitys	12

5.5	Historia	12
5.6	Kunnat	13
5.7	Myrkyt	13
5.8	Saaliit	13
5.9	Tarkastajat	13
5.10	Reviirit	13
5.11	Raportit	13
6	Toteuttamatta jääneet piirteet	14
6.1	Pakkaus haliaetus.command	14
6.2	Pakkaus haliaetus.helper	14
6.3	Muut tiedostot	14
7	Parannusehdotukset	15

1 Johdanto

Tämä toteutusdokumentti kuvaa toteutettavan Haliaeetus-järjestelmän teknisen toteutuksen niiltä osin, kun on poikettu suunnitteludokumentista.

1.1 Erikoissanasto ja käytetyt lyhenteet

CVS

Concurrent Versions System. Versionhallintaohjelmisto, joka on luotu helpottamaan ohjelmistojen versionhallintaa.

Haliaeetus järjestelmä, järjestelmä

Näillä tarkoitetaan koko toteutettavaa järjestelmää, joka sisältää käyttöliittymän, tietokannan ja näiden välillä olevat toiminnallisuudet.

HTML HyperText Markup Language. World Wide Webin eli WWW:n julkaisukieli.

HTTP Hypertext Transfer Protocol. Siirtokäytäntö eli protokolla, jonka varaan WWW rakentuu. Hypertekstidokumenttien siirtoa verkossa tukeva komentokieli.

HTTPS HTTP over Secure Sockets Layer. HTTP:n salakirjoitettu versio.

Istunto Samalta selaimelta tuleva sarja kyselyjä, jotka tapahtuvat määrättyssä ajanjaksossa.

Java Ohjelmointikieli, jota käytetään projektin toteutuksessa.

JDBC-ajuri Java DataBase Connectivity -ajuri, mahdollistaa tietokantakutsut palvelinsovelmista.

JDBC-yhteys

Hoitaa yhteyden tietokantaan, jonne järjestelmän tiedot on talletettu.

Käyttöliittymä

Se osa järjestelmästä, joka näkyy loppukäyttäjälle ja jolla järjestelmää käytetään.

Luokka Java-ohjelmointikielessä yksi kokonaisuus, kuten taulu tietokannassa.

Mallipohja Sama kuin template. Dokumentin runko, joka sisältää tietyllä tavalla merkityjä kohtia, joihin voidaan ohjelmallisesti lisätä vaihtuvia arvoja.

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

Java-luokasta.

Olio Java-luokan ilmentymä.

Servlet Java-ohjelmointikielellä kirjoitettu palvelinsovelma, servletti.

SQL Structured Query Language. Standardi kieli, jolla voidaan määrittää erilaisia tietokantaoperaatioita.

Tarkastus Pesille tehdään tarkastuskäyntejä muutaman kerran vuodessa, jolloin saadaan tietoa mm. poikasten määrästä ja pesässä vallitsevista olosuhteista.

Template Sama kuin mallipohja. Dokumentin runko, joka sisältää tietyllä tavalla merkityjä kohtia, joihin voidaan ohjelmallisesti lisätä vaihtuvia arvoja.

Tietokanta Jotain käyttötarkoitusta varten laadittu kokoelma toisiinsa liittyviä säilytettäviä tietoja. Tietokannan teknisiä ominaisuuksia ovat mm. tiedon riippumattomuus sitä käsittelevistä ohjelmista, tietojen samanaikainen käyttö, monipuoliset tiedonhakumahdollisuudet, tietojen suojaus, mutkikkaat riippuvuudet tietojen välillä ja automaattinen varmistus ja elpyminen häiriöistä.

TKTL Helsingin yliopisto, Tietojenkäsittelytieteen laitos.

2 Toteutuksen rajoitteet

Tässä kappaleessa kerrotaan, miltä osin suunnitellut rajoitteet poikkeavat suunnitteludokumentissa määritellyistä toteutusta koskevista rajoitteista.

2.1 Noudatettavat standardit ja tarvittavat ohjelmat

HTTPS-protokollaa ei höydynnetty toteutuksessa.

3 Tiedostojen sijainti

Tässä luvussa kuvataan järjestelmän käynnistys url ja järjestelmän käyttämien tiedostojen sijainnit. Järjestelmän käyttämät tiedostot sijaitsevat palvelimella `alkokrunni.cs.helsinki.fi`.

3.1 Käynnistys URL

Sovellus käynnistyy URL:ista

`http://db.cs.helsinki.fi/tomcat/kotkat/haliaeetus/HaliaeetusController`

3.2 Tiedostot alkokrunni palvelimella:

Kotkat-projektin tiedostot ja lähdekoodi toteutetuista näytöistä ja toiminnallisuuksista.

Juurihakemisto alkokrunnissa: `/home/kotkat`

`./tomcat/webapps/haliaeetus/WEB-INF/classes/haliaeetus:`

Konfigurointitiedostot:

`tabledata`

`./tomcat/webapps/haliaeetus/`

Poikasen ikätiedot-tiedosto

pdf-tiedostot:

`oldnest.pdf`

`newnest.pdf`

generoidut pdf-tiedostot

`./tomcat/webapps/haliaeetus/WEB-INF/classes:`

Tiedostot:

`haliResources_en.properties`

`haliResources_fi.properties`

`haliResources_sv.properties`

`haliaetuslog.properties`

`./tomcat/webapps/haliaeetus/WEB-INF/classes/haliaeetus/command:`

Pakkauksen `haliaetus.command` lähdekodit (`.java`) ja käännetyt (`.class`) tiedostot.

`./tomcat/webapps/haliaeetus/WEB-INF/classes/haliaeetus/db:`

Pakkauksen `haliaetus.db` lähdekodit (`.java`) ja käännetyt (`.class`) tiedostot.

`./tomcat/webapps/haliaeetus/WEB-INF/classes/haliaeetus/general:`

Pakkauksen `haliaetus.general` lähdekodit (`.java`) ja käännetyt (`.class`) tiedostot.

./tomcat/webapps/haliaeetus/WEB-INF/classes/haliaeetus/helper:

Pakkauksen haliaeetus.helper lähdekodit (.java) ja käännetyt (.class) tiedostot.

./tomcat/webapps/haliaeetus/WEB-INF/classes/haliaeetus/log:

Pakkauksen haliaeetus.log lähdekodit (.java) ja käännetyt (.class) tiedostot.

./tomcat/webapps/haliaeetus/WEB-INF/classes/haliaeetus/servlet:

Pakkauksen haliaeetus.servlet lähdekodit (.java) ja käännetyt (.class) tiedostot.

./tomcat/webapps/haliaeetus/WEB-INF/classes/haliaeetus/template:

Template- ja tyylitiedostot

./tomcat/webapps/haliaeetus/WEB-INF/lib/:

Kirjastotiedostot

./tomcat/webapps/haliaeetus/WEB-INF/log/:

Lokitiedostot

./tomcat/webapps/haliaeetus/WEB-INF/conf/:

Konfigurointitiedostot:

commandMapping.properties

./tomcat/webapps/haliaeetus/WEB-INF/:

Tiedostot:

web.xml

4 Osajärjestelmien toteutuspoikkeamat

Tässä luvussa kuvataan ne osat toteutetusta järjestelmästä, joiden toteutuksessa on poikettu suunnitteludokumentissa tehdyistä määräyksistä.

4.1 Pakkaus haliaeetus.db

Pakkauksen haliaeetus.db (ent. hali.db) toteutus on kaikilta osin suunnitellun mukainen.

4.1.1 ConnectionPool

Ei poikkeamia suunnitellusta.

4.1.2 Table

Ei poikkeamia suunnitellusta.

4.1.3 OperationResults

Ei poikkeamia suunnitellusta.

4.1.4 SearchResults

Ei poikkeamia suunnitellusta.

4.1.5 DatabaseOperation

Ei poikkeamia suunnitellusta.

4.1.6 SearchOperation

Ei poikkeamia suunnitellusta.

4.1.7 InsertOperation

Ei poikkeamia suunnitellusta.

4.1.8 UpdateOperation

Ei poikkeamia suunnitellusta.

4.1.9 DeleteOperation

Ei poikkeamia suunnitellusta.

4.2 Pakkaus haliaetus.servlet

Ent. hali.servlet.

4.2.1 HaliaetusControllerServlet

Ei poikkeamia suunnitellusta.

4.3 Pakkaus haliaetus.helper

Pakkauksesta haliaetus.helperin ei ole täysin noudatettu suunnittelua. Yleiskäyttöistä helper-yliluokkaa ei ole toteutettu. Ainoastaan kaksi erillistä helper-luokkaa toteutettiin.

4.3.1 NestInspectionHelper

Luokka ei tarkista kaikki kenttiä. Vain kantaan viemisen kannalta tärkeiden kenttien täytty tarkastetaan. Joitakin muitakin tarkastuksia on sen mukaan mitä asiakas on ilmoittanut. Kaikkien mumeeristen kenttien syöte tulisi tarkistaa, jotta lisäys-/päivitys-operaatio ei kaadu jos numerokentässä on esim. kirjain. Alasvetovalikoiden kohdalla tarkastuksia ei tarvita (paitsi jos kyseessä on kannan kahdalla pakollinen kenttä ja nämä tarkastukset on tehty). String syötteiden oikeellisuutta ei tarvitse tarkastaa. Syötteiden pituus on rajoitettu jo käyttöliittymässä, niin että kenttään voi laittaa vain kannan arvojen kokoisia syötteitä. Päivämäärien tarkastuksen puuttuvat. Pitää tarkastaa että syöte on annettu muodossa pp.kk.vvvv, muuten lisäys kaatuu.

Pesamuuttuva-taulun tarkastus ei kata ihan kaikkia sen kenttiä. Joidenkin kenttien nimi ei ole sama kuin kannassa on ja näiden tarkastaminen tulisi lisätä luupin ulkopuolelle, jotta mitään tietoa ei varmasti menetetä.

4.3.2 MunicipalityHelper

Ei poikkeamia suunnitellusta.

4.4 Pakkaus haliaetus.general

4.4.1 HaliProperties

Sijaitsi alunperin pakkauksessa haliaetus.helper, ei muita poikkeamia suunnitellusta.

4.5 Pakkaus haliaetus.command

4.5.1 HaliaetusGeneral

Ei poikkeamia suunnitellusta.

4.5.2 CommandDispatcher

Ei poikkeamia suunnitellusta.

4.5.3 CommandFactory

Ei poikkeamia suunnitellusta.

4.5.4 CommandInterface

Ei poikkeamia suunnitellusta.

4.5.5 LoginScreenCommand

Ei poikkeamia suunnitellusta.

4.5.6 MainScreenCommand

Ei poikkeamia suunnitellusta.

4.5.7 NestInspectionCommand

Luokka hoitaa kahta hommaa yhtä aikaa. Se sekä lisää uuden pesän (ja tietysti pesän sen vuoden tarkastuksen, koska uutta pesää ei voi lisätä ilman että siihen on samalla tehty tar-

kastus) että uuden johonkin pesään liittyvän tarkastuksen. Se kumpi toiminto toteutetaan päätellään Dispatcherilta saatavasta actionParametrilla.

Uuden pesän lisäys toimii, joidenkin kenttien arvot kuitenkin häviävät insertoinnin yhteydessä. Esim. pesävakio-kommentin arvo ei mene kantaan. Ongelma olisi hyvä selvittää ja tarkastaa arvot yksitellen.

Uuden tarkastuksen lisäys toimii siltä osin, että uusi rivi tauluun pesatarkastus (ja 4*poikanen-taulu) saadaan luotua. Pesamuuttuva-taulun insertointi sen sijaan ei onnistu. Voimassa olevan Pesamuuttuva-taulun kaikki tiedot pitää aina tarkastaa muutoksien varalta. Jos jotain on muuttunut, niin silloin viedään uusi rivi tauluun ja päivitetään ympäröiviä rivejä, jotta alku_pvm, loppu_pvm (jotka siis kertovat mikä pesamuuttuva-rivi liittyy mihinkin tarkastukseen) kenttiin ei synny rivien välisiä rakoja. Alku_pvm on aina tarkastuspäivämäärä ja loppu_pvm on joko vuoden viimeinen päivä tai seuraavan vuoden tarkastuksen tarkastus päivämäärä. Näin päivämäärät pysyvät "aukottomina". Jos muutoksia edellisen voimassa olevan pesamuuttuva taulun ja uuden tarkastuksen yhteydessä annettujen tietojen välillä ei ollut, niin kasvatetaan voimassa olevan rivin joko alku_pvm tai loppu_pvm kentän arvoa, riippuen siitä lisätäänkö tarkastus rivien alku- vai loppupäähän.) Tässä häsäkässä on jotain mätää, joko tarkastuksen osalta tai sitten päättelyn osalta, joka tehdään sen selvittämiseksi mihin kohtaan uusi rivi tulisi lisätä. Kyse voi olla boolean virheestä ta sitten sql-kyselyissä on jotain mätää. Pienellä lisä testaamisella virhe varmasti löytyy. Logiikassa ei pitäisi olla mitään vikaa.

Jo olemassaolevan pesän uuden vuositarkastuksen yhteydessä tapahtuva Pesamuuttuva-taulun rivien lisääminen/päivitys kantaan ei toimi. Eräissä uuden tarkastuksen lisäystilanteissa (takautuva tarkastus), pesamuuttuva-taulun ylläpito vaatii rivin jakamista kahdella (Pesamuuttuva-rivi kuvaa pesän harvoin muuttuvia tietoja tietyltä aikajaksolta, jonka alkupäivämäärä on alku_pvm ja loppupäivämäärä on loppu_pvm. Jos takautuvan tarkastuksen yhteydessä pesamuuttuvatiedot muuttuvat tällä aikaväliltä, niin rivi joudutaan jakamaan kahdeksi riviksi.) ja uuden rivin lisäämistä tuplatun rivin keskelle (ajanjaksollisesti keskelle). Joitakin ko. päivitykseen liittyviä osia on koodissa kommentoituna pois.

Poikanen tauluun viedään aina 4 riviä, jokaista tarkastusta kohtaa, vaikka poikaisten tietojen olisi annettu vain yhdestä tai ei yhdestäkään poikasesta. Tämä siksi että pesän tietojen korjauksen yhteydessä (RepairNestInfoCommand-luokka) voidaan poikasten tietoja lisäillä, ilman uuden rivin insertointia.

Kantaan tulee siis ylimääräisiä tyhjiä tauluja ja ratkaisua kannattaa pohtia jos pesä-tietojen korjaus toteutetaan eri tavalla.

4.5.8 RepairNestInfoCommand

Luokka on toteutettu, mutta sitä ei ole testattu sen enempää kuin että se kaatuu kesken suorituksen. Luokan tarkoitus on korjata jo kantaan syötettyjä tietoja. Se vertaa syötteenä saamiaan tietoja kannasta noutamiinsa ja jos jokin poikkeama löytyy se päivittää muuttuneen tiedon samantien ja jatkaa seuraavan tiedon tarkastamisella. Taas kerran ongelmana ovat tarkastuksen kannalta taulujen ja lomakkeen joidenkin kenttien erilaiset nimet. Nämä on pyritty minimoimaan, mutta esim. kunnan ja reviirin tarkastaminen pitää hoitaa erikseen, koska lomakkeelta tulee tunnus (String) ja kantaan on tallennettu id (number). Poikastietojen tarkastaminen ei esimerkiksi tällä hetkellä onnistu.

Tietoja ei myöskään tarkasteta tällä hetkellä mitenkään, vaan kuvitellaan että käyttäjä tietää mitä korjaa. Luokka voi tosin käyttää NestInspectionHelper-luokkaa tarkastuksien tekemiseen, kun kaikki tarkastukset on siihen saatu koodattua.

4.5.9 NestInformationCommand

Suunnitteludokumentista poiketen luokka toteuttaa vain seuraavanlaisia toimintoja:

Luokan päävastuulla on pesa.ftl:n käyttö kolmeen eri tarkoitukseen. Uuden pesän luonti, uuden vuositarkastuksen teko ja tietojen muuttaminen. Uuden pesän luonti ei saa parametreina mitään. Kannasta haetaan suurin ID jota kasvatetaan yhdellä. Uudessa vuositarkastuksessa luokka saa parametreina pesa_ID:n ja sen vuoden, minkä käyttäjä on syöttänyt tekstikenttään. Näiden perusteella tehdään kyselyt tauluihin. Tietojen muokkauksessa luokalle annetaan parametreina pesa_ID:n sekä tietty vuosi, joka on sijainnut sillä rivillä, jonka muokkaustoimintoa käyttäjä on hakusivulla painanut.

Muina toimintoina luokassa on lisäksi erilaisten virheilmoitusten näyttö sekä ilmoitus lisäys/päivitysoperaation onnistumisesta.

Koordinaatit sen sijaan ovat tällä hetkellä kommentoissa. Niitä ei siis tarkasteta ollenkaan, koska ne eivät toimi.

4.5.10 TerritoryCommand

Reviirien haku ja tietojen muuttaminen eivät toimi, mutta lisäyksen pitäisi onnistua.

4.5.11 MunicipalityCommand

Kunnille ei anneta nimeä vaan kantaan talletetaan vaan kuntatunnus.

Kuntien haku on muuttunut siten että kuntia voidaan hakea suuralueiden lisäksi kunnan tunnuksella koordinaattien perusteella tai ympäristökeskuksen perusteella.

4.5.12 InspectorCommand

Ei poikkeamia suunnitellusta.

4.5.13 HelperTableCommand

Ei poikkeamia suunnitellusta.

4.5.14 ReportCommand

Muita raportteja kuin Uusi pesä- ja Vanha pesä-raportit ei toteutettu.

4.6 Pakkaus haliaetus.log

Ent. hali.log.

4.6.1 LoggingRulesInitializer

Ei poikkeamia suunnitellusta.

4.6.2 HaliaetusLogger

Ei poikkeamia suunnitellusta.

4.7 Muut tiedostot

Tässä luvussa kuvataan muihin tiedostoihin tehdyt muutokset ja lisätyt uudet tiedostot.

4.7.1 Template-tiedostot

Toteutetut template-tiedostot:

- aputaulut.ftl
- haku.ftl
- login.ftl (ent. hali.ftl)
- mainscreen.ftl (ent. haku.ftl?)
- municipality.ftl (ent. kunnat.ftl)
- navi.ftl
- pesa.ftl
- report.ftl (ent. reportit.ftl)
- reviirit.ftl
- tarkastajat.ftl

4.7.2 Properties-tiedostot

Kielituen properties-tiedostot Tiedostot toteutettiin pääasiassa suomenkielisinä, joten kielituki ei muodostunut kovin kattavaksi.

Komentoluokkien properties-tiedostot Ei poikkeamia suunnitellusta.

4.7.3 Tyylitiedosto

Ei poikkeamia suunnitellusta.

4.7.4 Staattiset html-tiedostot

Ei poikkeamia suunnitellusta.

5 Käyttöliittymä

Tässä luvussa kerrotaan käyttöliittymään toteutusvaiheessa tehdyistä muutoksista.

5.1 Käyttäjän sisäänkirjautuminen

Ei poikkeamia suunnitellusta.

5.2 Haku

Ei poikkeamia suunnitellusta.

5.3 Uuden pesän lisääminen ja vanhan pesän tarkastus

Käyttöliittymästä puuttuvat linkit reviiirillä oleviin muihin pesiin sekä aiemmin samalle pesälle tehdyt tarkastukset. Vanhat tarkastukset vielä voisikin linkittää mutta reviiirillä olevien muiden pesien linkitys on huono idea. Jos tehdään tarkastusta vuodelle 2003 ja reviiirillä olevia muita pesiä ei ole sinä vuonna tarkastettu, niin mihin linkki johtaa? Vuoteen 2002 vai 2004, vai kenties vuoteen 1976? Suunnitteludokumentissa ei ole otettu tuohon kantaa, mutta tulevat projektit voivat miettiä halutaanko linkitystä toteuttaa.

Lisäksi kälistä on poistettu muutama varmistusalasveto, eli mm. "Onko pesä tekopesä" Tieto ei mene kantaan laisinkaan, vaikka se lomakkeelta löytyykin. On aivan tarpeellonta vaihtaa kälistä alasetovalikon arvoa, jos sitä ei mihinkään viedä. Tätäkin asiaa voivat tulevat ryhmät pohtia.

Ja on vielä eräs asia, joka kälistä puuttuu; munan pituus. Se on otettu pois, koska arvo tallettuu Myrkyt-tauluun, jota ei tämän projektin puitteissa ylläpidetä.

5.4 Aputaulun päivitys

Kun taulu-valikosta valitaan joku taulu, niin attribuutti-valikko ei päivity automaattisesti, vaan on painettava nappulaa 'hae attribuutit'.

5.5 Historia

Ei toteutettu.

5.6 Kunnat

Kunnan lisäyksen kenttien asettelu on muuttunut, ovat nyt yhdellä rivillä. Kunnan nimelle ei luonnollisesti enää ole kälissä kenttää, koska kantaan talletetaan vain kuntatunnus.

Myös kaikki hakukentät ovat yhdellä rivillä, ja hakukenttiin on lisätty Suuralue-, Ympäristökeskus-, Kuntatunnus- ja Kordinaatti-kentät.

Kunta-sivulla Tee kuntaliitos-painike ja checkbox-kentät, jotka ovat siis hakutuloksen jokaisen kunnan rivillä, ja joiden avulla kunta voidaan valita kuntaliitokseen, siirtyivät alkuperäisestä suunnitelmasta poiketen oikealle puolelle sivua. Sivulle on lisätty myös Reset-nappula, jolla voidaan tyhjentää kentät.

5.7 Myrkyt

Ei toteutettu.

5.8 Saaliit

Ei toteutettu.

5.9 Tarkastajat

Mitään virhe- tai lisäysilmoituksia ei ilmaannu, vaikka toiminnot ovatkin kunnossa.

5.10 Reviirit

Mitään virhe- tai lisäysilmoituksia ei ilmaannu, vaikka lisäystoiminto onkin kunnossa.

5.11 Raportit

Kälistä poistettiin checkbox-kentä 'Koko maa' tarpeettomana.

6 Toteuttamatta jääneet piirteet

Tässä luvussa luetellaan suunnitteludokumentissa toteutettavaksi suunnitellut järjestelmän osat, joita ei ole toteutettu.

6.1 Pakkaus haliaetus.command

Seuraavia suunnitteludokumentissa mainittuja kohtia ei toteutettu ollenkaan

- PoisonCommand
- PrayCommand
- HistoryCommand

6.2 Pakkaus haliaetus.helper

- CheckHelper
- CheckPoisonHelper
- CheckPrayHelper
- CheckHistoryHelper
- CheckInspectorHelper
- CheckTerritoryHelper
- CheckReportHelper
- Coords

6.3 Muut tiedostot

Seuraavat template-tiedostot jäivät toteuttamatta.

- myrkyt.ftl
- saalis_vanhat.ftl

- saalis_yllapito.ftl
- historia.ftl

7 Parannusehdotukset

Keskeneräisessä tuotteessa suurin parannus olisi tehdä se valmiiksi eli toteuttaa puuttuvat toiminnot

Lisäksi useammassakin yhteydessä, ennen kaikkea suurensuuria pesälomakkeita käsitellessä, tuli esiin ajatus yleisten tarkastusmetodien ohjelmoinnista. Metodeja voisi käyttää esim. päivämäärien ja numeerisuuden varmentamiseen.

Pesämuuttuva-taulun käsittelyä voisi muuttaa niin, että jokaista tarkastusta kohden olisi oma rivi pesämuuttuva-tilussa. Rivimäärä jonkin verran kasvaa ja vuosi-tiluun pitää lisätä uusi kenttä (pesämuuttuva-tilun avain pesämuuttuva_id), mutta tällä muutoksella ohjelma toimisi jo nykyin oikein ja kenttä kentältä tehtävän tarkastuksen (joka on tehtävä jokaisella lisäys kerralla) aikavaativuuskin on varsin kunnioitettava. Riski väärin toimimisesta myös pienenesi kun käsittelylogiikka yksinkertaistuisi huomattavasti.

RepairNestInfoCommand-luokan rakenne suunniteltiin päivityksen kannalta sellaiseksi kuin se on, koska luokan ajateltiin olevan typojenkorjausluokka, jolloin montaa kenttää ei kerrallaan muutettaisi. Luokka kuitenkin mahdollistaa myös tietojen myöhemmän lisäyksen, joten tältä kannalta toimintaa voisi muuttaa hieman yhtenäisemmäksi. Esim. kerätty muuttuneet tiedot yhteen HashMap:in ja viedä kantaan kerralla yksittäisten operaatioiden sijaan. Uusia rivejä ei tauluihin tarvitse lisätä.

Pesätarkastus-lomakkeen alavetovalikoiden toteutusta voisi myös kehittää. Kyllä-/ei-tyyppisten attribuuttien käyttöliittymäesitys on toteutettu alavetovalikkona, jonka arvojoukkona on Q=tyhjä, K=kyllä ja E=ei. Vaikka kieltä vaihtaisi, K ja E säilyvät valikossa. QKE-alavetovalikoiden arvot generoidaan paikallisesti NestInformationCommand-luokassa, eikä niitä haeta Aputauluista kaikkien muiden alavetovalikoiden arvojen tapaan. Ongelmana on se, että joidenkin valikoiden arvot löytyvät silti myös Aputaulusta, sillä niitä ei ole sieltä ehditty vielä poistaa. Haitta on pieni, mutta jos halutaan toteuttaa toimiva järjestelmä, niin Aputauluja voisi noilta osin siistiä.

Alavetovalikoiden arvot haetaan tällä hetkellä jokaikinen kerta, kun sivu ladataan. Tämä tekee systeemistä häiritsevän hitaan. Tulevaisuudessa arvot kannattaisi hakea heti järjestelmään kirjauduttaessa, tallettaa sessioon ja päivittää sinne, jos muutoksia tulee. Se nopeuttaisi järjestelmää huomattavasti.

Uuden pesän lisäyksen jälkeen sivu latautuu uudestaan ja ylhäällä on ilmoitus: Uuden pesän lisäys onnistui. Sivun alalaidassa on kuitenkin yhä Tallenna-nappula. Järjestelmä saattaa kaatua, jos kyseistä toimintoa koettaa suorittaa uudestaan. Asiaa ei tosin ole testattu. Tuleville polville voisi kuitenkin jättää mietittäväksi, minkälaiselle sivulle lisäyksen jälkeen tullaan. Pitäisikö nappula disabloida vai kenties suorittaa siirtymä täysin uudelle näytölle?

Myös kielituki olisi syytä ottaa laajamittaisempaan käyttöön tulevissa järjestelmissä.

Raportoinnissa itse pdf-lomake ei sisällä kielitukea. Tätä ei ollut explisiittisesti sanottu vaatimuksissa, mutta toisaalta on luonnollista, että mikäli Haliaeetus-järjestelmä toteuttaa kielituen, myös pdf-lomakkeet tulisi pystyä tulostamaan valitulla kielellä. Itse lomake voisi olla suomeksi, ruotsiksi ja englanniksi, vaikka arvot tulevatkin kannasta vain suomeksi.

Lisäksi itse raportitkin voitaisiin tuottaa tyylikkäämmin. Käyttämällä iText open source -projektia saatiin helposti generoitua pdf-lomake, joka oli vastaava asiakkaan Vanha pesä -lomakkeen kanssa (lähinnä koska asiakkaan antama MS Word-lomake konvertoitiin pdf-muotoon, joten itse lomaketta ei tarvinnut generoida kooditse).

Parempiakin tapoja on generoida pdf-lomake Javasta, kuten Apachen FOP (<http://xml.apache.org/fop/index.html>) yms, mutta iText-ratkaisuun päädyttiin sen nopeuden ja helppouden vuoksi.