

## **Ylläpitodokumentti**

KotKot

Helsinki 14.12.2008

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (9 + 1 op)

**Projektiryhmä**

Tuomas Puikkonen

Matti Seise

Paula Mäenpää

Olga Karmanov

Jonne Kohvakka

**Asiakas**

Heikki Lokki

**Johtoryhmä**

Sampo Yrjänäinen

**Kotisivu**

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

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
1.0	14.12.2008	Valmis ylläpitodokumentti

## Sisältö

<b>1 Johdanto</b>	<b>1</b>
1.1 Sanasto	1
1.2 Pesiin liittyvä sanasto	1
1.3 Tekniseen toteutukseen liittyvä sanasto	1
<b>2 Ohjelmiston yleiskuvaus</b>	<b>3</b>
<b>3 Toteutetut ja toteuttamatta jääneet toiminnallisuudet</b>	<b>3</b>
3.1 Toteutetut	3
3.1.1 Käyttöliittymä	3
3.1.2 Pesätietolomakkeen tallennus palvelimen kansioon	3
3.1.3 Pesätietolomaketietojen nouto palvelimen kansioista	3
3.1.4 Järjestelmään kirjautuminen	4
3.1.5 Järjestelmään rekisteröityminen ja omien tietojen muokkaaminen	4
3.1.6 Rengastajan lisääminen	4
3.1.7 Karttapaikka	4
3.2 Toteuttamatta jääneet	4
3.2.1 Kuntaliitokset	4
3.2.2 Raportit	4
3.2.3 Käyttöliittymä	4
3.2.4 Pesien lisäys järjestelmään ja pesätietojen muokkaaminen	4
3.2.5 Pesätietolomakkeen tietojen välitys	5
3.2.6 Pesätietolomakkeiden hyväksyminen	5
3.2.7 Tietojen salaus	5
3.2.8 Loki	5
3.2.9 Reviirit	5
<b>4 Ohjelmiston toimintaympäristö ja käyttöönotto</b>	<b>5</b>
<b>5 Esivaatimukset</b>	<b>5</b>
5.1 Asennus	5
5.2 Ohjelman käynnistys	6

# 1 Johdanto

Tämä dokumentti sisältää Haukka-järjestelmän ylläpitoon ja jatkokehittämiseen tarvittavia tietoja. Dokumentissa on kuvattu järjestelmän rakennetta ja toimintaa sekä toteutuksen ja testauksen aikana ilmi tulleita puutteita. Ylläpidodokumentti on tarkoitettu luettavaksi yhdessä muiden dokumenttien ja ohjelmakoodin dokumentin kanssa.

## 1.1 Sanasto

### 1.2 Pesiin liittyvä sanasto

#### **Lomake**

Lomakkeella tarkoitetaan pesän tarkastuksen yhteydessä täytettävää tietolomaketta, johon kirjataan kaikki havaitut tiedot pesästä. Lomakkeen tiedot syötetään tietokantaan.

#### **Pesä**

Paikka, johon petolinnun pesä on rakennettu. Pesä voi sijaita puussa, tekopesässä, luonnonkolosissa, "pötkelössä", kallionkielekkeellä tai maassa. Muita mielenkiitosisia pesäpaikkaan vaikuttavia tekijöitä ovat alueet, joille se on rakennettu. Esim. suo, saari, vaara, puisto jne.

#### **Petolintu**

Muita eläimiä tai hyönteisiä syövät linnut ovat petolintuja. Järjestelmän kannalta merkittäviä petolintuja ovat pöllöt ja haukat.

#### **Raportti**

Raporteilla tarkoitetaan järjestelmän palauttamia tietoja, joista kootaan tekstimuotoinen dokumentti.

#### **Reviiri**

Alue pesän ympärillä, jolla sijaitsevat yhden lintuparin pesä ja vaihtopesät. Yhdessä pesässä voi olla käyttäjinä vuoden aikana usean lajin edustajia, jolloin sama pesä kuuluu useaan eri reviiriin. Järjestelmän tuottamiin raportteihin valitaan vain pesinnän lopputulokseltaan paras pesä.

#### **Tarkastus**

Pesille tehdään tarkastuskäyntejä - usein juuri pesimisaikaan. Tarkastuskäynneillä poikaset mitataan ja rengastetaan sekä kerätään näytteitä ja tietoja pesimäpaikasta myöhempää analyysia varten. Etenkin Pohjois-Suomessa tarkastuksiin käytetään myös lentokonetta, pesinnän tai sen puuttumisen toteamiseen pesintäkauden alkuvaiheessa.

### 1.3 Tekniseen toteutukseen liittyvä sanasto

#### **Apache tomcat palvelin**

Avoimen lähdekoodin palvelin, joka tuo käyttäjän koneelle tietokone ohjelman, tyypillisesti HTML:n välityksellä.

#### **Admin**

Museovirkailija, jolla on pääsy kaikkialle (paitsi käyttäjien salasanoihin). Vain admin voi lisätä uusia museovirkailijoita järjestelmään.

”Cascading Style Sheets, CSS ” www- ohjelmille kehitetty tyyliohjeiden laji. Annetut säännöt ehdottavat kuinka dokumentti voidaan esittää.

### **DB**

Database eli 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.

### **FreeMarker**

Java-kirjasto, jolla voidaan valmiita pohjia täydentämällä luoda erilaisia dokumentteja.

### **Haukka-järjestelmä**

Tässä dokumentissa määritellyn tietokantajärjestelmän nimi.

### **XHTML**

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

### **HTTP**

Hypertext Transfer Protocol. Siirtokäytäntö eli protokolla, jonka varaan WWW rakentuu. Hyper-tekstidokumenttien siirtoa verkossa tukeva komentokieli.

### **HTTPS**

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

### **Java**

Ohjelmointikieli, jota käytetään projektin toteutuksessa.

### **Javascript**

Komentosarjakieli joka toimii pohjana selaimissa nykyisin käytyille ECMAScript standardia noitattaville komentosarjakielille.

### **JDBC**

Java Database Connectivity. Ohjelmointirajapinta, joka mahdollistaa pääsyn lähes mihin tahansa tietolähteeseen Java-ohjelmointikielestä.

### **Järjestelmä**

Tässä dokumentissa järjestelmällä tarkoitetaan pääasiallisesti toteutettavaa ohjelmistoa tai jo valmista ohjelmistoa.

### **KotKot**

Tässä dokumentissa määritellyn tietokantajärjestelmän toteuttavan ohjelmistotuotantoprojektin nimi.

### **Mokkula-komponentti**

Kertakäyttöiseksi tarkoitettu tiedon siirtoväline, jolla saadaan olemassa olevat pesätiedot siirrettyksi uuteen tietokantaan.

### **Selain**

Tietokoneohjelma, joka mahdollistaa käyttäjälleen katsella ja lähettää tekstiä sekä kuvia. WWW-

selaimet kommunikoiivat WWW- palvelimien kanssa käyttäen http-protokollaa. Suosittuja selaimia ovat esimerkiksi Internet Explorer (IE) ja Mozilla Firefox.

### **Servlet**

Java-ohjelmointikielellä kirjoitettu palvelinsovelma, servletti. Servletti on www-palvelimella oleva javaohjelma, joka käynnistetään kutsumalla sitä selaimella.

### **Template**

Freemarkerin käyttämä mallipohja dokumenttien luomiseen.

### **Tietokantajärjestelmä**

Ks. järjestelmä.

## **2 Ohjelmiston yleiskuvaus**

Järjestelmän keskeisin osa on www-palvelin, joka on kytketty Tomcat-ohjelmistoon. Tietokantapalvelimella on käynnissä itse ohjelmisto, joka käyttää erillisiä konfiguraatiodietoja ja mallipohjia tuottaakseen käyttäjälle ulkoasun ko. tietokannan käyttämiseen. Käyttäjä on yhteydessä järjestelmään selaimella. Käyttäjän lähettämät ja haluamat tiedot haetaan tietokantapalvelimelta, joka on yhteydessä vain tietokantapalvelimella pyörivään ohjelmistoon.

## **3 Toteutetut ja toteuttamatta jääneet toiminnallisuudet**

Järjestelmään saatiin luotua suuri määrä irrallisia komponentteja, joiden implementoiminen järjestelmään jäi aikataulutuksen takia kesken.

### **3.1 Toteutetut**

#### **3.1.1 Käyttöliittymä**

Käyttöliittymän templatet ovat valmiita. Samoin valmiina ovat käyttöliittymän parsimiseen käytettävät komponentit (Javascriptit).

#### **3.1.2 Pesätietolomakkeen tallennus palvelimen kansioon**

Rengastajan ilmoittamia tarkastuskäynnin tietoja ei saa tallentaa suoraan tietokantaan, vaan piti kehittää joku "välitallennuspaikka", josta museovirkailijan käyttöliittymä noutaa tiedot ja tallentaa ne kantaan tarkistusten jälkeen. Teknisesti lomakkeen talletus palvelimen Tarkistukset-kansioon on kunnossa, mutta toimintoa ei ehditty implementoida järjestelmään.

#### **3.1.3 Pesätietolomaketietojen nouto palvelimen kansioista**

(Ks. edellinen) Teknisesti lomakkeen tietojen nouto palvelimen Tarkistukset-kansioista on kunnossa, mutta toimintaa ei ehditty implementoida järjestelmään.

### **3.1.4 Järjestelmään kirjautuminen**

Toteutettu ja implementoitu.

### **3.1.5 Järjestelmään rekisteröityminen ja omien tietojen muokkaaminen**

Toteutettu ja implementoitu.

### **3.1.6 Rengastajan lisääminen**

Toteutettu käyttöliittymässä, implementointi kantaan tekemättä.

### **3.1.7 Karttapaikka**

Pesätietolomakkeen yhteenvetosivulta on linkki karttapaikkaan aiemmin annetuilla koordinaateilla.

## **3.2 Toteuttamatta jääneet**

### **3.2.1 Kuntaliitokset**

Kahdesta vanhasta kunnan nimestä on mahdollista luoda yksi isompi kunta omassa näkymässään museovirkailijan käyttöliittymässä. Kuntaliitokset-näkymään on linkki näytön yläreunassa muiden linkkien ohella.

### **3.2.2 Raportit**

Tekstimuotoisia raportteja saadaan antamalla ohjelmalle 1-2 hakuehtoa. Raportit-näkymä on museovirkailijan etusivulla.

### **3.2.3 Käyttöliittymä**

Käyttöliittymästä puuttuvat reviirit-, kuntaliitokset- ja museovirkailijan etusivu (jolla myös raporttien ajo ja pesätietolomakkeiden hyväksyminen) -näkymät. Myös museovirkailijan merkitseminen "Adminiksi" toteuttamatta.

### **3.2.4 Pesien lisäys järjestelmään ja pesätietojen muokkaaminen**

Käyttöliittymä suurimmaksi osaksi valmis, ainoastaan muutamia Javascript-tarkistuksia puuttuu. Tietokanta- ja tarkistuskomponentit ovat valmiita. Käyttöliittymästä siirtyvät attribuutit ovat sidottuina järjestelmän omaan tietorakenteeseen. Parametrien välitystä tietokantaan ja takaisin käyttöliittymälle ei ehditty implementoida.

### 3.2.5 Pesätietolomakkeen tietojen välitys

Rengastajan täyttämä pesätietolomake lähetetään museovirkailijalle tarkistettavaksi. Lomake tallennetaan ensin JSON-tekniikalla XML-tyyppiseksi tiedostoksi palvelimen Tarkastukset-kansioon, josta museovirkailijan käyttöliitymä hakee ne.

### 3.2.6 Pesätietolomakkeiden hyväksyminen

Museovirkailijan käyttöliitymän etusivulla on listaus linkkejä hyväksymättömistä rengastajien lähettämistä tarkastuskerroista. Yhtä otsaketta klikkaamalla aukeaa näkymä, jossa virkailija voi muuttaa tietoja, tai vaihtoehtoisesti poistaa lomakkeen. Kun virkailija hyväksyy lomakkeen tietojen tallentamisen tietokantaan, poistetaan automaattisesti kyseinen XML-tiedosto Tarkastukset-kansiosta.

### 3.2.7 Tietojen salaus

Koordinaattikentät pystytään Oraclen puolella salaamaan, kunhan ko. tietokannan asetustiedostoa pääsisi muuttamaan.

### 3.2.8 Loki

Lokiin tallentuvat tekstimuodossa kaikki (pesiin liittyvät) kantaan tehdyt muutokset. Lokiin pääsee museovirkailijan käyttöliitymästä.

### 3.2.9 Reviirit

Museovirkailija määrittää pesien reviirit. Reviirit-näkymä on museovirkailijan käyttöliitymässä ja sinne päästään reviirit-linkistä näytön yläreunassa.

## 4 Ohjelmiston toimintaympäristö ja käyttöönotto

Toimintaympäristönä toimii Oracle-tietokanta, sekä Javaa tulkitseva välikieli. Välikielenä toimii Tomcat-ohjelmisto.

## 5 Esivaatimukset

Tomcatin CLASSPATH:ssa pitää olla Oraclen ojdbc-ajuri  
(parametri -classpath <tomcat-base>/webapps/Haukka/WEB-INF/lib/ojdbc14.jar)

### 5.1 Asennus

Ohjelma asennetaan kopioimalla hakemisto "/Haukka" tomcatin haluttuun alihakemistoon. Tämän jälkeen tomcat pitää käynnistää uudelleen.



## 5.2 Ohjelman käynnistys

Ohjelman tiedostojen kopioinnin ja Tomcatin uudelleenkäynnistyksen jälkeen ohjelmaan pääsee käsiksi selaimella osoitteesta: <http://<valittupolku>/Haukka>, missä "<valittupolku>" on se osoitepolku, jonne ohjelman tiedostot asennettiin.