

# KÄYTTÖLIITTYMÄ SÄÄKSIEN PESIMÄTIETOJEN TIETOKANTAAN

## Ylläpitodokumentti

### Versiohistoria

| Versio | Päivämäärä | Kuvaus                        | Tekijä            |
|--------|------------|-------------------------------|-------------------|
| 1.0    | 27.4.2007  | Ensimmäinen virallinen versio | Hannu Hartikainen |

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Ohjelmistotuotantoprojekti Sorsat

# Sisällysluettelo

|   |    |
|---|----|
| Tietojenkäsittelytieteen laitos.....                  | 1  |
| Ohjelmistotuotantoprojekti Sorsat.....                | 1  |
| Ylläpitodokumentti.....                               | 1  |
| 1 Johdanto.....                                       | 1  |
| 2 Sanasto ja käytetyt lyhenteet.....                  | 1  |
| 2.1 Sääksiin liittyvä sanasto.....                    | 1  |
| 2.2 Tekniikkaan liittyvä sanasto.....                 | 2  |
| 3 Arkkitehtuuri.....                                  | 3  |
| 3.1 Järjestelmän yleiskuvaus.....                     | 3  |
| 3.2 Datamalli.....                                    | 4  |
| 3.3 Komponenttien väliset suhteet.....                | 5  |
| 3.4 Kaavio järjestelmän osien yhteistoiminnasta.....  | 6  |
| 4 Tietokanta.....                                     | 6  |
| 5 Luokat ja paketit.....                              | 7  |
| 6 Lokitiedostot ja virheiden tulostaminen.....        | 9  |
| 7 Käyttöliittymä.....                                 | 9  |
| 7.1 Yleiskuvaus.....                                  | 9  |
| 8 Järjestelmän tuottamat dokumentit ja sivut.....     | 12 |
| 8.1 PDF dokumentit.....                               | 12 |
| 8.2 HTML- sivut.....                                  | 12 |
| 8.3 Toteutetut ominaisuudet ja tunnetut puutteet..... | 13 |
| Esille tulleita ideoita jatkokehitykseen.....         | 13 |

# Ylläpitodokumentti

## 1 Johdanto

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

## 2 Sanasto ja käytetyt lyhenteet

### 2.1 Sääksiin liittyvä sanasto

- **Kalasääski** ks. sääksi.
- **Maastotarkastus** ks. tarkastus
- **Pesä.** Sääksen pesä on Suomessa yleensä hyvin kookas risupesä suuren puun latvassa, josta näkyvyys on hyvä. Myös jokin muu korkea paikka kelpaa pesän paikaksi. Näkyvyyttä tärkeämpi tekijä pesäpaikan valinnalle on esteetön lentomahdollisuus pesälle. Pitkine siipineen sääksi on kömpelö lentäjä. Pariskunnan pesien käyttö määrää reviirien muodostumista. Kts. Reviiri.
- **Reviiri.** Yhdellä reviirillä elää yksi sääksipariskunta. Yleensä pariskunnalla on reviirinsä alueella useampia pesiä. Näistä pesistä yleensä yksi on kerrallaan käytössä. Reviiri voi jakaantua useamman kunnan alueelle ja pesien käytöstä riippuen päätetään, perustetaanko uusia reviirejä vai yhdistetäänkö vanhoja. Ks. Pesä.
- **Sääksi.** Latinalaiselta nimeltään Pandion haliaetus, on pitkä, kapeasiipinen petolintu, jonka pituus on 53- 61 cm ja siipien kärkiväli 140 - 165 cm. Sääksi on eräs maailman laajimmalle levinneistä lintulajeista. Sääksi kelpuuttaa ympäristökseen lähes minkä tahansa paikan, kunhan se on riittävän rauhallinen ja sieltä löytyy sopiva pesäpuu. Sääksi munii yleensä 2-3 munaa, mutta etenkin nuoren sääksen ensimmäisinä pesävuosina pesintä usein epäonnistuu.
- **Tarkastaja.** Vapaaehtoinen rengastaja jolla on rengastajanumero ja lupa tehdä tarkastuksia sääksien pesille. Maastotarkastaja tarkastaa vuosittain lintujen pesiä ja palauttaa tiedot tarkastuslomakkeella. Ks. Tarkastus.
- **Tarkastus.** Yleensä pesille tehdään tarkastuskäyntejä pesimisaikaan kerran vuodessa. Ihannetilanne olisi kuitenkin, että pesällä käytäisiin kesän aikana kolmeen kertaan. Kaikista käyntikerroista koostetaan kuitenkin vain yksi maastoraportti. Tarkastuskäynneillä poikaset pyritään mittaamaan ja rengastamaan. Käynneillä kerätään näytteitä ja tietoja pesimäpaikasta myöhempää analyysia varten. Kerätyt tiedot palautetaan tarkastuslomakkeella.
- **Tarkastuslomake** Järjestelmästä tulostettava, esitäytetty lomake, johon tarkastaja kirjaa tarkastuksen yhteydessä tekemänsä havainnot. Täytetyn raportin tarkastaja joko lähettää kirjeitse rengastustoimistoon tai sitten tarkastaja voi siirtää tiedot sähköiselle lomakkeelle, jolla lähetetyt tiedot rengastustoimiston vastaava henkilö hyväksyy ja siirtää tietokantaan.
- **Tekopesä.** Tekopesäksi kutsutaan ihmisen tekemää pesää sääkselle. Sääksille on rakennettu tekopesiä tukeväksi pesimispaikaksi. Tällä keinolla pyritään takaamaan pesimismahdollisuus, koska vanhoista männyistä on pulaa.
- **Uusi pesälomake** Tyhjä tarkastuslomake, johon kirjataan pesän tiedot, jota ei vielä ole kannassa.

## 2.2 Tekniikkaan liittyvä sanasto

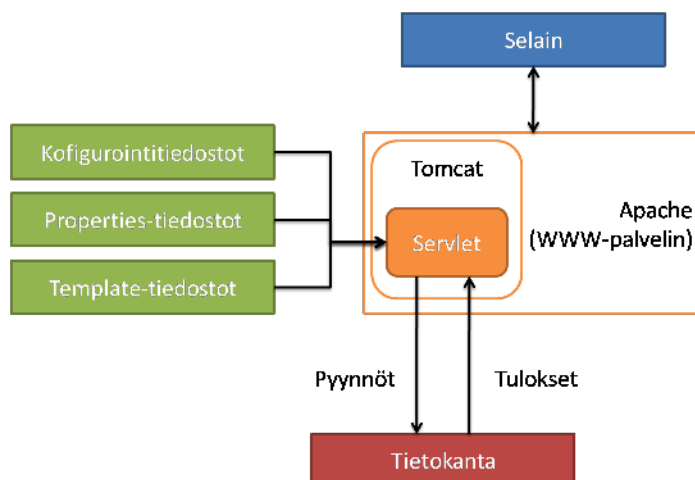
- **Apache tomcat palvelin** Avoimen lähdekoodin palvelin, joka tuo käyttäjän koneelle tietokone ohjelman, tyypillisesti HTML:lään välityksellä.
- **Cascading Style Sheets, CSS.** Erityisesti www- ohjelmille kehitetty tyyliohjeiden laji. Annetut säännöt ehdottavat kuinka dokumentti voidaan esittää. Varsinaisia tyylikieliä ovat CSS1 ja seuraaja CSS2.
- **Datamalli.** Kts. tietokartta.
- **Document Object Model DOM.** Rajapinta HTML tai XML-sivujen muokkaamiseen ohjelmointikieltä käyttäen.
- **ECMAScript** Ecma internationalin standardoima selainten käyttämä komentosarjakieli.
- **FreeMarker.** Java-kirjasto, jolla voidaan valmiita pohjia täydentämällä luoda erilaisia dokumentteja.
- **HTTPS.** HTTP: n (Hypertext Transfer Protocol) salattu versio. Tätä muotoa käytetään suojattuun tiedon siirtoon www:ssä. Tiedot salataan ennen lähettämistä SSH- protokollan avulla.
- **Hypertext Markup Language, HTML.** Standardoitu kuvauskeli, jolla pystytään kuvaamaan hyperlinkkejä sisältävää tekstiä. HTML mahdollistaa tekstin rakenteen merkinnän, esimerkiksi mikä osa on otsikko ja mikä taas leipätekstiä. Merkintä tapahtuu tekstin sekaan kirjoitettavilla elementeillä ja niiden sisällä olevilla määrittelyillä.
- **JDBC.** Java database connectivity. Ohjelmointirajapinta, jota käytetään kommunikoitaessa järjestelmän tietokannan kanssa.
- **Javascript** Komentosarjakieli joka toimii pohjana selaimissa nykyisin käytyille ECMAScript standardia noidattaville komentosarjakielille.
- **Järjestelmä.** Tässä dokumentissa järjestelmällä tarkoitetaan pääasiallisesti toteutettavaa ohjelmistoa tai jo valmista ohjelmistoa kts. Tietokantajärjestelmä.
- **Kehitysympäristö.** Ohjelmoijien omalla koneella käytössä oleva versio ohjelmasta, tarkoitettu muutosten todentamiseen ja testaamiseen.
- **Kansalaisten Karttapaikka.** Maanmittauslaitoksen tarjoama verkkopalvelu.
- **Käyttöliittymä.** Ne välineet ja toiminnot, joilla käyttäjä on yhteydessä tietojärjestelmään eli käytännössä ohjelmistoon.
- **Merikotkajärjestelmä** Alun perin sääksijärjestelmästä kehitetty versio merikotkien seurantaan.
- **Mallipohja** ks. Template
- **Open Database Connectivity, ODBC.** Microsoftin määrittelemä rajapinta (API) tietokannoille. Tämän avulla sovellusten on mahdollista kommunikoida tietokantapalvelimen kanssa.
- **Oracle 10.** Oracle-relaatiotietokannan versio 10g, jolla sääksiseurannan tietokanta toteutetaan.
- **PDF.** Portable Document Format on esitystapa dokumenteille.
- **Properties tiedosto** Lähinnä Java ympäristössä käytetty tiedostotyyppi, joka mahdollistaa esim. ohjelman lokalisoinnin varsin helposti. Sisältää ohjelmassa esimerkiksi alusvetovalikoiden termit.
- **Relaatiotietokanta** Relatiotietokantakoostuu joukosta loogisesti yhteenkuuluvia taulukkoja, joissa on rivit ja sarakkeet kuten taulukkolaskennan tauluissa. Jokaisessa taulukossa on ainakin yksi *avainkenttä*, jonka perusteella tietueita voi lajitella, yhdistää ja valikoida. Tyypillinen avainkenttä on henkilötunnus, joka on jokaiselle henkilölle eri. Kaksi taulua liittyy toisiinsa yhden yhteisen kentän - toisen taulukon avainkentän - avulla. Linkitettyjen taulujen avulla tietokannan tietoja voidaan yhdistää ja lajitella monin eri tavoin. Se mahdollistaa tietosisällön oikeellisuuden tarkistamisen (tieto esiintyy vain yhdessä paikassa) ja estää tiedon moninkertaisen syötön.
- **Sähköinen lomake** Tarkastuslomake, joka on sähköisessä muodossa.
- **Sääksijärjestelmä** Järjestelmä sääksien pesintätietojen seurantaan, jonka tämä vaatimusdokumentti kuvailee.

- **Selain.** Tietokoneohjelma, joka mahdollistaa käyttäjälleen katsella ja lähettää tekstiä sekä kuvia. WWW- selaimet kommunikoivat WWW- palvelimien kanssa käyttäen http-protokollaa. Suosittuja selaimia ovat esimerkiksi Internet Explorer (IE) ja Mozilla Firefox.
- **Structured Query Language, SQL\*.** IBM:n kehittämä standardoitu kyselykieli, jota käytetään projektin toteutuksessa. Kielellä voidaan tehdä relaatiokantaan esimerkiksi hakuja, muutoksia ja lisäyksiä.
- **Sorsat.** Kevään 2008 ohjelmistotuotantoprojektiryhmä, jonka tavoitteena on toteuttaa käyttöliittymä tämän vaatimusmäärittelyn pohjalta (vrt. Tipu4 ja Tiput).
- **Template** Freemarkerin käyttämä mallipohja dokumenttien luomiseen.
- **Tietokanta.** On 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.
- **Tietokantajärjestelmä.** Tietokantajärjestelmällä tarkoitetaan tietokannan, tietokannanhallintajärjestelmän ja tietokantaa käyttävän sovellusohjelman luomaa kokonaisuutta.
- **\*Tietokartta.\*Map-** tyyppinen olio joka pitää sisällään HashMap ja ArrayList- tyyppisiä olioita. Käytetään parametrivälityksessä järjestelmän sisällä, mutta päätehtävä on välittää kulloinkin tietosisältö mallipohjille käyttöliittymän näkymän luomiseksi. Merikotkajärjestelmä käyttää tietokartasta nimityksiä datamalli tai datamodel.
- **Tipu4.** Ryhmä, joka keväällä 2004 toteutti sääksijärjestelmän ohjelmistotuotantoprojektina.
- **Tiput.** Vastaavan ohjelmistotuotantoprojektiryhmän nimi syksyllä 2006
- **Tru64 Unix.** Digital Unixin nimellä aikaisemmin tunnettu 64 bittinen HP:n omistama Unixin versio.
- **Tuotanto ympäristö** Ilmaisulla tarkoitetaan valmista ohjelmistoa, eli asiakkaan näkemä kokonaisuus.
- **World Wide Web, WWW.** Maailmanlaajuinen verkko on internetissä toimiva hypertekstijärjestelmä. WWW on yksi palvelumuoto, jota käytetään internetin avulla.

## 3 Arkkitehtuuri

### 3.1 Järjestelmän yleiskuvaus

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



Järjestelmä perustuu käyttäjän selaimen, servlet- palvelimen ja tietokantapalvelimen väliseen vuorovaikutukseen. Yhteys käyttöliittymän ja tietokannan välillä on toteutettu kolmitasoisella arkkitehtuurilla.

Taso 1: Tasolla 1 oleva selain on vuorovaikutuksessa sekä käyttäjän että tasolla 2 olevan ohjelmiston kanssa.

Taso 2: Palvelinkoneella pyörii Apache WWW- palvelin, sekä Tomcat- palvelin. Palvelinkoneella sijaitsevat JDBC- ajuri, servletit, JDBC- tietokantaluokat, Freemarker kirjastoluokat sekä staattiset mallipohjat- ja properties- tiedostot. Apache- Tomcat - pari käyttää HTTPS- protokollaa kommunikoidessaan käyttäjän selaimen kanssa ja JDBC- ajuria kommunikoidessaan tietokannan kanssa. Servletit käyttävät tietokantaa JDBC- tietokantaluokkien avustuksella. Servletit myös generoivat dynaamista HTML:ää ja kommunikoivat käyttäjän selaimen kanssa. Servletit saavat tulostamiensa HTML- sivujen rungot mallipohjista, ja esitettävät tiedot tietokannasta.

Taso 3: Tasolla 3 on Sääksi tietokanta, jossa täytyy olla vastaava JDBC- ajuri, joka tukee JDBC API:a.

## 3.2 Datamalli

Datamalli on puutietorakenne, jonka avulla siirretään dataa järjestelmän sisällä. Datamalli on Map-olio, jonka sisällä voi olla sekä Map- olioita että List- oliota. Järjestelmän toteutuksessa Map-olioina käytetään Map- rajapinnan toteuttavia HashMap- luokan ilmentymiä ja List- oliona List-rajapinnan toteuttavia ArrayList- luokan ilmentymiä. Map- ja List- rajapintojen käyttäminen mahdollistaa kuitenkin kaikkien rajapinnan toteuttavien luokkien käytön data- mallissa. Datamallissa muuttujiin viitataan merkkijonoilla. Kirjoitusvirheiden välttämiseksi nämä merkkijonot on siirretty Const- luokkien vakioiksi.

Datamalli sisältää järjestelmän toiminnassa välitettäviä parametreja. Järjestelmän luokat muuttavat datamallin sisältöä tarpeen mukaan, eli se sisältää aina vain tarvittavat parametrit. Datamalli voi sisältää tunnuksen ja salasanan, tietokantaan talletettavia attribuutteja, tietokantahakuun tarvittavia määreitä, virheilmoituksia, tietoja tuotettavien html- sivujen dropdown- listoista tai konfigurointitiedoston polkumääreen.

Datamalli kuvataan seuraavasti puuna:

- (root)
  - solmu
    - muuttuja
  - sekvenssi[]
    - solmu
      - muuttuja1
- muuttuja2

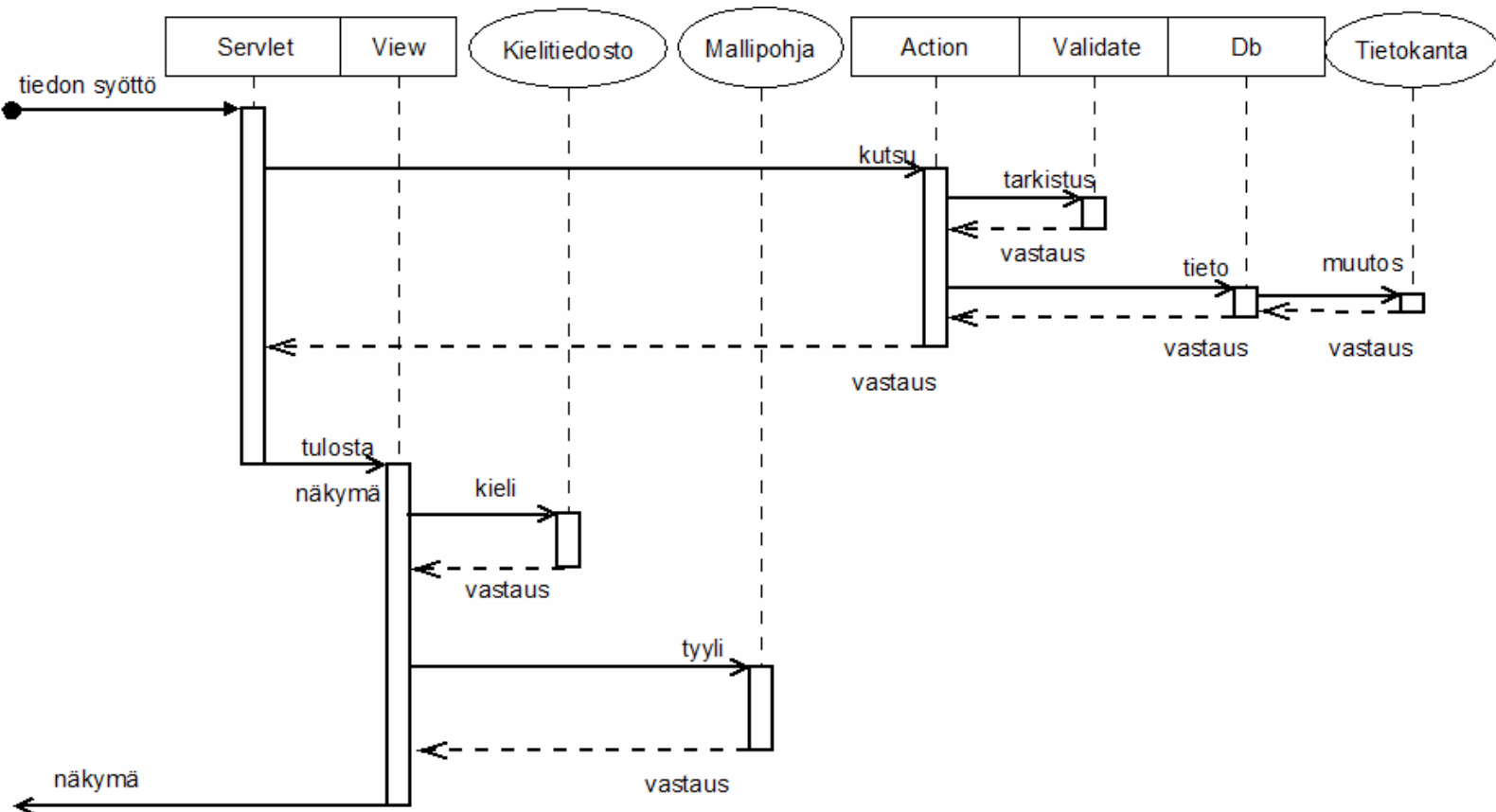
Mallissa jokainen puun normaali solmu voi toimia tyhjänä solmuna (=hakemisto) tai muuttujana (sisältää arvon). Sekvenssityyppiset solmut taas toimivat kuin taulukko. Esimerkissä sekvenssi toimii kuin taulukko, ja viittaus sekvenssi[1] viittaa sekvenssin toiseen muuttujaan/alkioon. Sekvenssin alkiot voivat toimia myös hakemistorakenteina, ja yhdessä alkiossa voi olla monta muuttujaa. Sekvenssirakenteet toteutetaan List-rajapinnan toteuttavan luokan ilmentymien avulla. Tavallisiin muuttujiin viitataan mallipohjassa esim. mu.muuttuja.

Toteutuksen kannalta oleelliset oksat ovat:

- **datamalli.action** Oksaan laitetaan servletin (TBD:nimi) saamat toimintoparametrit (syötetty html- lomakkeella), jotta varsinaiset toiminnon suorittavat luokat voivat päätellä mitä toimintoa tarkalleen halutaan.
- **datamalli.text** Oksa sisältää näyttöihin tulevat tekstit muodossa datamalli.\_text näyttö.kenttä.
- **datamalli.tietokantataulu** Jokaiselle tietokantataululle löytyy oma oksansa. Arvot on

- *esitetty muodossa `datamalli.tietokantataulu.attribuutti`.*
- **datamalli.data** Oksa sisältää tietokantahauissa tarvittavat hakuparametrit, jotka tulevat lomakkeelta. Tässä oksassa pidetään myös kenttien tiedot, jotka halutaan säilyttää servlet-pyyntöjen välillä.
- **datamalli.searchResults** Oksa sisältää tietokantahakujen tulokset.
- *`datamalli.error.hasErrors` Muuttuja kertoo onko validoinnissa tapahtunut jokin virhe. Se asetetaan validointiluokissa.*

### 3.3 Komponenttien väliset suhteet



Kaavio komponenttien välisistä suhteista

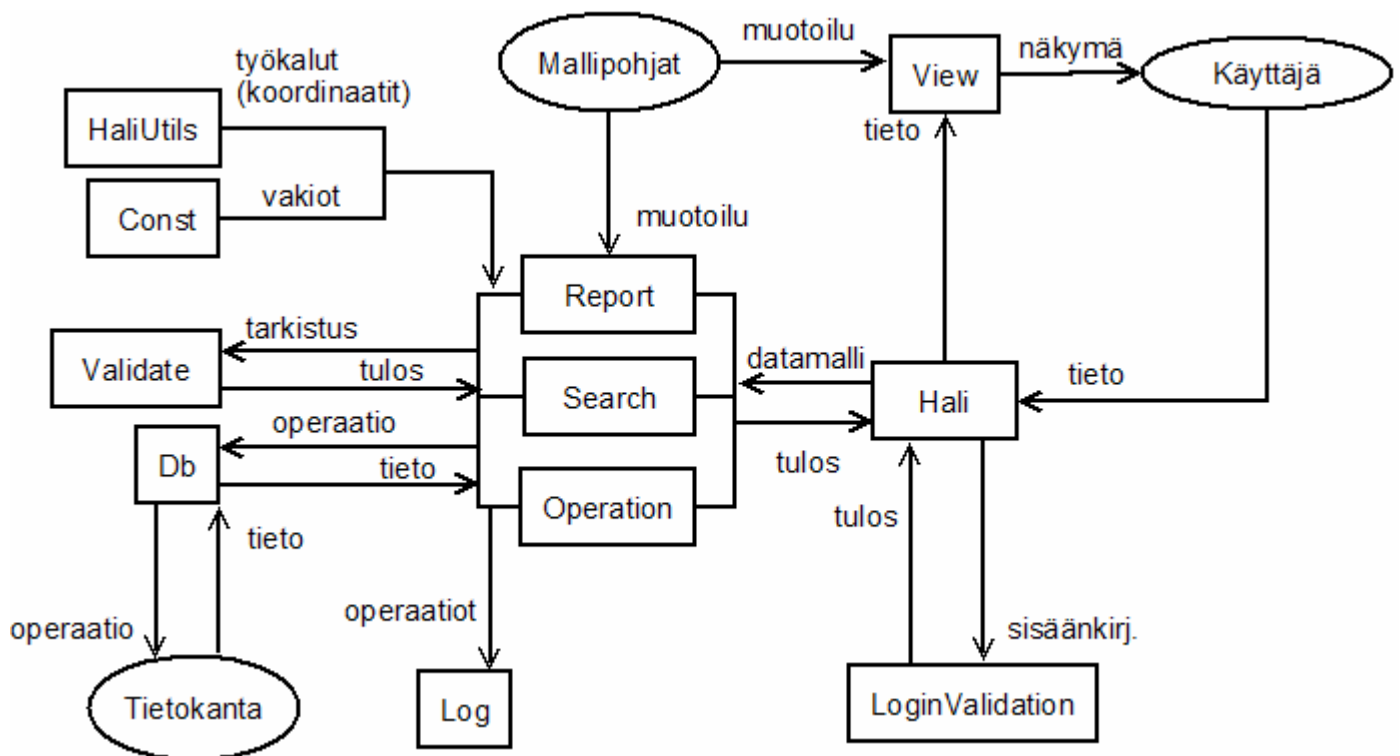
Käyttäjän syöttämä informaatio lähetetään selaimen toimintopainikkeista pakkauksen `fi.hy.hali.servlet.servlet-` luokalle, joka toimii kontrolliservletinä. Servletti kutsuu tarvittavia `fi.hy.hali.Action-` pakettien luokkia ja `fi.hy.hali.View-` luokkaa, jotka toteuttavat kutsun mukaisen toiminnallisuuden. Action- luokat tarkistavat hakuehdot ja kantaan syötettävät tiedot `fi.hy.hali.Validate-` pakkauksen luokkien avulla ja suorittavat tietokantaoperaatiot pakkauksen `fi.hy.hali.Db-` tietokantaluokkien avulla. Suoritettava tietokantaoperaatio voi olla haku-, lisäys- tai muokkausoperaatio. Action- luokat palauttavat tietokantaluokilta saamansa vastauksen servletille, joka välittää sen View- luokalle uuden näytön generoimista varten. Näytön luonnissa View- luokka käyttää apuna valmiita mallipohjia sivun ulkoasun määrittelemisessä ja tietojen oikeaan kenttään kohdentamisessa. Kielituki toteutetaan View- luokassa, joka hakee tarvitsemansa erikieliset tekstit `properties-`tiedostoista. Konfigurointitiedot laitetaan tiedostoon `(TBD:nimi).config`, josta niitä voidaan lukea.

Järjestelmä sisältää seuraavat keskeiset komponentit:

- General sisältää luokat jotka suorittavat järjestelmän tarvitsemia aputoimintoja (vakiot ja aputyökalut)

- Db sisältää tietokantaluokat. Luokkien avulla suoritetaan haku-, lisäys- ja muokkausoperaatiot JDBC- yhteyden kautta.
- Servlet sisältää kontrolliservletin Hali. Hali on järjestelmän ainoa servlettiluokka, ja se hoitaa järjestelmän toiminnanohjauksen.
- Action on muiden action- luokkien abstrakti ylliluokka.
- Action.Search sisältää hauille yhtenäisiä muuttujia ja metodeita.
- Action.Operation sisältää päivitysten ja lisäysten tarvitsemia yhteisiä muuttujia ja metodeita.
- Validate tarkistaa tietokantaan vietävien tietojen sekä siihen kohdistuvien hakujen oikeellisuuden. Komponentti sisältää myös luokan sisäänkirjautumisen tarkistamiseen.
- Log:n tehtävänä on kirjata tietokannan taulujen attribuutteihin kohdistuvat muutokset tekstitiedostoihin. Lisäykset ei kirjata; ainoastaan arvojen muutokset.
- View on luokka jonka tehtävänä on tuottaa kaikki järjestelmän tarvitsemat näkymät. Luokka yhdistelee saamansa parametrit sekä näkymään tulevat otsikkotekstit ja välittää ne itse tulostusosalle joka käyttää mallipohjia niiden tulostamiseen. Toistuvat näkymät (navigointiosa) tuotetaan erillisissä metodeissa.

### 3.4 Kaavio järjestelmän osien yhteistoiminnasta



Kaavio järjestelmän osien yhteistoiminnasta havainnollistaa eri komponenttien suhdetta toisiinsa.

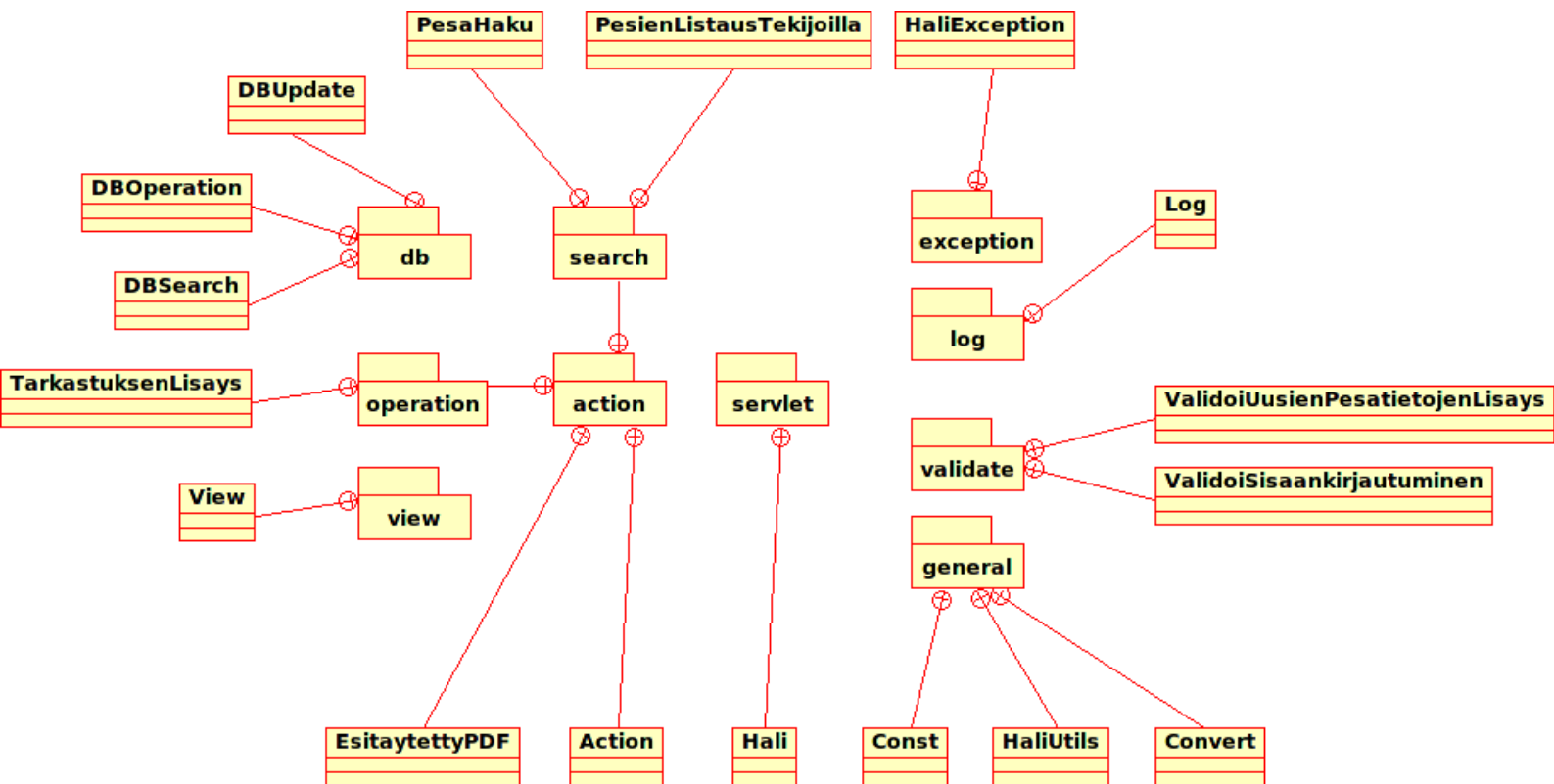
## 4 Tietokanta

Tietokantana toimii olemassaoleva sääksitietokanta. Tarkempi kuvaus tietokannan rakenteesta löytyy vaatimusdokumentista. Tietokannan rakenteeseen on tehty joitakin muutoksia. Kaikkiin aputauluihin on lisätty järjestys-kenttä, jotta valinnoille voidaan antaa oikea järjestys käyttöliittymässä. Pesä-taulun ”yht\_pituus”-kenttä on muutettu 6-numeroisesta 7-numeroiseksi, jotta koordinaatti mahtuu kokonaisuudessaan kantaan. Aputauluun on lisätty ”apukoord\_mittaustapa” gps- ja karttapaikannuksen koodeja varten. Apukoord\_tarkkuus-tauluun on lisätty vaihtoehdot Euref-koordinaatteja ja tarkemmin mitattuja yhtenäiskoordinaatteja varten.

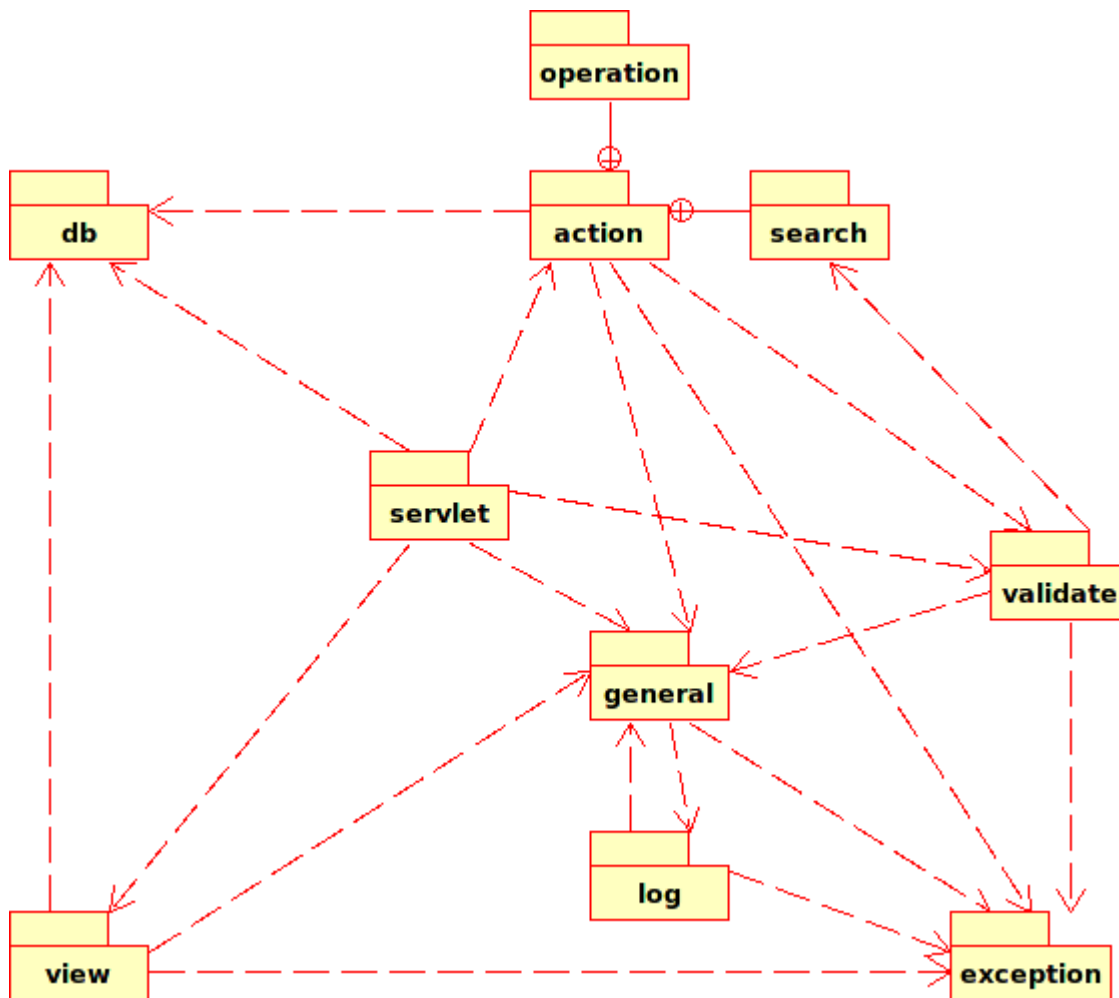


# 5 Luokat ja paketit

Tässä luvussa kuvataan lyhyesti toteutettavan järjestelmän käyttämät paketit. Toiminnan kannalta oleellisin luokka on Hali, joka Action ja View luokan kanssa tuottaa käyttäjälle sivujen ulkoasun ja toiminnallisuuden.



*Kaavio järjestelmään kuuluvista luokista ja paketeista*



Kaavio pakettien välisistä riippuvuussuhteista

Tarkemmat kuvakset luokista löytyvät PDF-tiedostosta. Ohessa lyhyet kuvaukset kaavioissa esiintyvistä paketeista:

#### **fi.hy.hali.action**

Sisältää operation ja search -paketit sekä esitetyt PDF -lomakkeiden tuottamiseen käytetyt luokat.

#### **fi.hy.hali.action.operation**

Sisältää luokat, jotka muokkaavat sääksijärjestelmän tietoja annettuun dataan perustuen.

#### **fi.hy.hali.action.search**

Sisältää luokat, jotka etsivät tietoja annettujen hakuehtojen perusteella.

#### **fi.hy.hali.db**

Sisältää järjestelmän tietokantakutsut suorittavat luokat.

#### **fi.hy.hali.exception**

Sisältää järjestelmässä esiintyvät poikkeusluokat.

### **fi.hy.hali.general**

Sisältää järjestelmän yleisiä apuluokkia.

### **fi.hy.hali.log**

Sisältää lokiin kirjoittamiseen liittyvät luokat.

### **fi.hy.hali.servlet**

Sisältää järjestelmän ainoan Servlet-luokan Hali. Hali on pääluokka, joka muita luokkia käyttäen toteuttaa järjestelmän toiminnallisuuden.

### **fi.hy.hali.validate**

Sisältää luokat, jotka validoivat ja muokkaavat käyttäjän selaimeen syöttämät tiedot.

### **fi.hy.hali.view**

Sisältää luokat, jotka tuottavat Freemarker-lomakepohjia käyttäen selaimelle palautettavat sivut.

## **6 Lokitiedostot ja virheiden tulostaminen**

Järjestelmä ei ylläpidä erillisiä itse luomiaan lokitiedostoja.

Tietokantaan tapahtuvista muutoksista tallentuu tieto pelkästään tietokannan omaan lokiin. Järjestelmän kautta ei ole pääsyä tietokannan pitämään lokiin.

Virhetilanteissa virheilmoitukset tulostetaan Tomcatin lokiin eli tiedostoon .../tomcat/logs/catalina.out. Virheilmoituksien tulostukseen käytetään fi.hy.hali.general-luokan debugMessage(message, className)-metodia, joten virheilmoitukset ovat yhtenäisiä. Virheilmoitukset alkavat sanalla DEBUG ja sisältävät aikaleiman.

## **7 Käyttöliittymä**

Tämä kappale kuvailee käyttöliittymän toiminnan ja toteutuksen. Käyttöliittymä jakautuu kolmeen päähaaraan, joiden toiminta on kuvailtu erikseen ensimmäisessä kappaleessa. Toinen kappale kuvailee käyttöliittymän teknisen toteutuksen. Tässä esitelty käyttöliittymän lopullinen toteutus poikkeaa hieman vaatimusmäärittelydokumentista esitellystä.

### **7.1 Yleiskuvaus**

Käyttöliittymä jakautuu seuraavaksi esiteltäviin kolmeen päähaaraan. Päähaarat jakautuvat aliosioihin.

#### **Uusien pesätietojen lisäys**

Tästä haarasta löytyvät kaikki varsinaisten pesätietojen, eli pesien tietojen, olosuhdetietojen, reviirien ja tarkastusten kantaan syöttöön tarvittavat työkalut. Osiossa on kolme alaosiota, jotka ovat erilaisia tapoja käyttäjälle lisätä pesätietoja. Kaikki tavat käyttävät kuitenkin lopulta samaa lomaketta tietojen syöttämiseksi.

Lomakkeen toiminta on seuraava:

Lomake on käyttötilanteesta poiketen ensin joko täysin tyhjä, tai sille on esitäytetty tietyn pesän perustiedot ja edelliset olosuhdetiedot. Nykyinen vuosi on aina esitäytettynä tarkastuksen

päivämäärä -kohtaan. Lomakkeen lähettämisen jälkeen syötetyt tiedot validoidaan. Jos tiedoissa havaitaan virheitä, palautetaan käyttäjä lomakkeelle ja tulostetaan virheestä kuvaava teksti lomakkeen yläosaan. Lisäksi virheelliset kentät merkitään punaisella värillä. Tarkastuksen syötön onnistuessa käyttäjä palautetaan siihen paikkaan käyttöliittymää, josta käyttäjä saapui lomakkeelle. Lisäksi paluusivun yläosaan tulostetaan onnistumisteksti ja linkki (pesän id), josta käyttäjä voi palata tarkastelemaan ja muokkaamaan juuri lisäämäänsä tarkastusta.

Lomakkeessa käyttäjällä on kaksi mahdollisuutta valita reviiri: joko käyttäjä valitsee yhden vanhoista reviireistä, tai käyttäjä kirjoittaa uuden reviirin nimen sille varattuun tekstikenttään, jolloin siis luodaan uusi reviiri ja tarkastuksen kohteena oleva uusi tai vanha pesä liitetään tälle vuodelle siihen reviiriin. Reviirien valinnan apuna tarkastuslomake tarjoaa mahdollisuuden hakea "lähiympäristön pesiä". Käyttäjä syöttää pesän koordinaatit ja säteen lomakkeelle, ja saa listauksen säteen sisällä olevista muista pesistä. Käyttäjälle tarjotaan myös linkki lisätä löydettyjä pesiä karttapalveluun ("Kansalaisten karttapaiikka").

Kaikki edellä mainitut pätevät myös vanhojen tarkastusten tietojen muokkaamiseen, johon käytetään samaa lomaketta. Tässä tapauksessa lomakkeelle on luonnollisesti esitätetty kaikki kyseisen tarkastuksen tiedot. Tarkastusten muokkausten tapauksessa käyttäjälle tarjotaan myös linkit selata pesän muiden vuosien tarkastuksia.

Ensimmäinen alahaara on "Uusi pesä" -haara, josta lisätään täysin uuden pesän tiedot ja ensimmäinen tarkastus tietokantaan. Tämä haara tuo käyttäjän näkyviin suoraan tyhjän tarkastuslomakkeen.

Seuraavat kaksi alahaaraa tarjoavat vaihtoehtoisia tapoja edetä lisäämään tarkastuksia vanhoille pesille.

Alahaaran "(lisäys) Tarkastajan tiedoilla" -valinta pitää sisällään tekstikentän ja lähetyispainikkeen. Tekstikenttään käyttäjä voi kirjoittaa joko tarkastajan numeron tai nimen (tai nimen osan). Jos käyttäjä kirjoittaa numeron ja lähettää lomakkeen, tulostetaan kyseisen rengastajan aktiiviset pesät. (Aktiivinen pesä: kaikki pesät joiden viimeisin pesintäkoodi ei ole 'D'). Jos käyttäjä kirjoittaa nimen, etsitään tekstiä vastaavat rengastajanimet joilla on tarkastettuja sääksen pesiä. Mikäli osumia löytyy vain yksi, siirrytään käyttöliittymä suoraan kyseisen tarkastajan aktiivisten pesien listaukseen, mutta useamman osuman tapauksessa näytetään lista osumia vastaavista tarkastajista, joista käyttäjä voi sitten valita yhden aktiiviset pesät listattavaksi.

Alahaara "(lisäys) Pesän tiedoilla:" -valinta on edellisen kaltainen, mutta tekstikentän avulla ei haeta pesiä tarkastajan vaan pesän tietojen mukaan. Käyttäjä voi syöttää kenttään pesän id:n, nimen, reviirin nimen, kylän nimen, kunnan nimen, vanhan pesänumeron tai läänin nimen. Jälkimmäisen tapauksessa vaaditaan tarkka osuma, muut tekstisyötteet etsitään osahakuna. Jos käyttäjä syöttää pesäid:n (numeerinen arvo) siirrytään suoraan tarkastuslomakkeeseen. Muissa tapauksissa näytetään pesälistausta hakuehdon täyttävistä pesistä.

Molempien ylläkuvattujen pesälistauksen listaustapa on samanlainen. Löydetty pesät listataan läänin, kunnan, reviirin ja pesän nimen mukaan järjestettyinä. Jos listatulle pesälle ei ole vielä nykyiselle vuodelle lisätty tarkastusta, tarjotaan käyttäjälle linkki pesän nimen kohdalle siirtyä lisäämään uutta tarkastusta kyseiselle pesälle. Jos tarkastus nykyiselle vuodelle on jo tehty, ei edellistä linkkiä tulosteta, vaan pesän nimen oikealle puolelle tulostetaan "Tarkastele/Muokkaa" -linkki, josta käyttäjä voi siirtyä tarkastelemaan tai muokkaamaan pesän kyseistä tarkastusta.

## **Pesätietojen haku ja käyttö**

"Pesätietojen haku ja käyttö" -pääosio tarjoaa käyttäjille työkalut etsiä pesiä erilaisten hakutekijöiden perusteella. Lisäksi tarjotaan tarkastusten hallinnoinnissa tarvittavia työkaluja.

"Pesien haku tekijöillä" -alaosiossa tarjotaan käyttäjille tarkastuslomaketta muistuttava tapa hakea pesiä. Kaikki kantaan syötettyjä tietoja (ja niiden yhdistelmiä) voidaan hakea tämän lomakkeen avulla. Haku tukee vain "and" -hakua. Hakuehtoinen täsmäävät pesät listataan järjestettynä läänin, kunnan, reviirin ja pesän mukaan. Listauksesta käyttäjä voi joko siirtyä tarkastelemaan kyseisen

pesän tiettyä tarkastusta, tai siirtyä myös tätä kautta lisäämään kyseiselle pesälle uutta tarkastusta. Käyttäjällä on myös mahdollisuus valita kaikki tai osa pesistä "haaviin". Haavissa oleville pesille voidaan suorittaa erinäisiä toimintoja. Näiden toteutus on vielä tämän projektin puitteissa epävarmaa.

"Tarkastustyökalut" -alaosiosta käyttäjä voi tulostaa jonkin vuoden kaikkien aktiivisten pesien tarkastuslomakkeet järjestettynä tarkastajan, läänin, kunnan, reviirin ja pesän nimen mukaan, tai vain tietyn tarkastajan lomakkeet. Lisäksi mahdollisesti tarjotaan tämän projektin puitteissa mahdollisuus listata tietyn vuoden tarkastamattomat pesät (ja tarkastettujen/tarkastamattomien pesien määrät).

## **Aputietojen hallinta**

"Aputietojen hallinta" -pääosio tarjoaa käyttäjälle työkaluja muokata järjestelmän aputietoja. Näiden toteutus tämän projektin puitteissa on vielä epävarmaa. Muokattavia tietoja olisivat ainakin: reviirien nimet, kuntien nimet, kuntaliitoksien teko, aputaulujen kenttien sisältö ja rengastajatiedot.

## **Tekninen toteutus**

### **Templatet**

Käyttöliittymä on toteutettu käyttämällä Freemarker-templateja. Templaten tarvitsemat tiedot annetaan tietorakenteessa (tietokartta), jonka ylin haara koostuu HashMap-oliosta. HashMapin avaimet voivat sisältää muuttujien arvoja, kuten tekstiä tai numeroita, tai ArrayList-olioita jotka puolestaan pitävät sisällään uusia HashMap-olioita.

Käyttöliittymän pääsivujen tuotoille on jokaiselle oma templatetiedosto, jotka käyttävät apuna (include) aina kolmea muuta template-tiedostoa: `_runkoAlku` ja `_runkoLoppu` pitävät sisällään käyttöliittymän header ja footer -tiedot, jotka ovat samat kaikille templateille. Lisäksi templaten alkuun lisätään kyseisen templaten `_alaosiovalikko`. Templatetiedosto itse pitää sitten sisällään varsinaisen rungon.

### **Monikielisyys**

Käyttöliittymä tukee kaikilta osin mahdollisuutta tehdä järjestelmästä monikielinen. Tietokarttaan lisätään kaikki käyttöliittymän tarvitsemat tekstit kielitiedostosta (`haliResources.fi.properties`). Teksteille on annettu nimi jolla niitä voidaan käyttää templatetiedostossa, sekä itse teksti. Jos haluttaisiin lisätä tuki toiselle kielelle, luotaisiin kopio `_fi` -kielitiedostosta, korvattaisiin sen tekstit toisen kielisillä teksteillä ja View-luokassa ladattaisiin parametrina saatua kieltä vastaava kielitiedosto.

### **Toiminnanohjaus**

Käyttöliittymässä on tarvetta kahdenlaiselle toiminnanohjaukselle: paluulinkin tarjoaminen tarkastuslomakkeelle sinne sivulle, jolta käyttäjä saapui lisäämään tarkastusta ja paluu onnistuneen tarkastuksen lisäyksen jälkeen samoin sille sivulle, joilta käyttäjä saapui lisäämään tarkastusta. Tässä apuna käytetään seuraavia tietoja, jotka välitetään joko osoitteessa tai hidden -kenttinä aina sivulta toiselle: `OHJAUS_HAUNTYYPPI` pitää sisällään paluusivun tunnisteiden. Eri hakujen tunnisteet ovat niin ikään vakiokenttiä. `OHJAUS_ID`-kenttään tallennetaan paluusivun tuottanut id (esimerkiksi rengastajan numero, pesän id, ..). `OHJAUS_ALKUPERAINENHAKUEHTO` -alkuperäinen hakuehto, esimerkiksi rengastajan nimi. Alkuperäistä hakuehtoa ei todennäköisesti varsinaisesti käytetä missään käyttöliittymän osiossa. Alkuperäinen idea oli, että jos esimerkiksi rengastajan nimellä pesiä listattaessa saadaan useita osumia rengastajista joista käyttäjä sitten valitsee yhden, niin hakuehtona säilyisi silti käyttäjän kirjoittama hakuehto, mutta paluu tapahtuisi silti yksilöivästi rengastajan numeron perusteella. `OHJAUS_OLI_YKSILOIVA` -kentässä on jotain tekstiä, jos haku tuotti vain yhden osuman ja siirryttiin välittömästi haun jälkeen

tarkastuslomakkeelle: tällöin ei ole mitään sivua jolle palata, vaan palataan uuden pesän luontiin ja ei tulosteta paluulinkkiä. Toiminnanohjauksen suunnittelu ja toteutus on puutteellista ja vaikeasti ylläpidettävää.

"Uusien pesätietojen lisäys" -sivulla käytetään painetun napin nimeä erottamaan, onko pesälistaus tarkastajan ja pesän tietojen välillä. Näille napeille on omat vakiotekstinsä, joiden oikeat arvot pitää välittää linkeille/lomakkeille, jotka osoittavat kyseisille sivuille.

## 8 Järjestelmän tuottamat dokumentit ja sivut

### 8.1 PDF dokumentit

Sääksijärjestelmä tuottaa PDF- raportteja, joiden ominaisuudet täsmentyvät toteutuksen seuraavassa vaiheessa. Järjestelmän ei ensimmäisessä vaiheessa tuota raportteja.

PDF-dokumenttien tuottamiseen käytetään Javan iText-rajapintaa.

Esitäytettyjä lomakkeita varten luodaan täyttämätön kaavake, johon täytetään tiedot iText-rajapinnan avulla. Täytetty lomake tallennetaan PDF-tiedostoksi käyttäen iTextiä. Täyttämättömään mallikaavakkeeseen luodaan valmiita tyhjiä kenttiä, joita täytetään iTextillä. Kenttien nimet ovat ennalta sovittu, jotta kenttiin voi iTextillä syöttää sisältöä.

Esitäytetyn lomakkeen olisi myös voinut luoda kokonaan iTextillä, tai kenttien sisällön olisi voinut syöttää valmiiseen mallipohjaan myös koordinaattien avulla, mutta käyttämällä kenttien nimiä on mallipohjaa myöhemmin helpompi muokata millä tahansa toimisto-ohjelmistolla.

### 8.2 HTML- sivut

Järjestelmän käyttöliittymä on toteutettu HTML- sivuina ja järjestelmä tuottaa näin ollen suuren määrän erilaisia HTML- sivuja. HTML- sivut on tehty noudattaen XHTML 1.0 transitional -spesifikaatiota, sikäli kun mikään painava syy ei pakota standardista poikkeamiseen. Mahdolliset poikkeamat dokumentoidaan erikseen. HTML-sivut luodaan FreeMarker mallipohjista.

#### Staattisesti HTML- sivuihin lisätyt tiedostot

Sivut hyödyntävät CSS- tyylitiedostoja ja JavaScript- komentosarjoja. Elementeille on annettu class- ja id-määritteitä vain tarvittaessa. Käytettyjen CSS-tyylejen nimet on nimetty kuvaavasti, kuvaten elementin tyyppiä ja käyttötarkoitusta. Id:nä käytetään soveltuvin osin tietokannan nimistä koostettuja tekstejä: taulu\_sarake. ID:n käytön tarve on tässä vaiheessa hyvin vähäistä. Lisäksi tässä vaiheessa järjestelmään toteutetaan yksi iframe -elementti, josta tarkempi kuvaus seuraavassa kappaleessa.

#### Dynaaminen HTML

Dynaamisen HTML:n käyttöä on vältetty ja DOM- rajapintaa käyttävät kutsut on rajattu niihin tilanteisiin, jossa saavutetaan merkittävää hyötyä vastaavaan staattiseen toteutukseen nähden.

Järjestelmään on toteutettu seuraavat dynaamiset toiminnallisuudet:

- Tarkastuslomakkeessa rengastajan valinta dropdown menulla / dropdown menun päivitys kirjoitettaessa tekstikenttään rengastajan numero
- Valitun koordinaattityypin korostaminen tyyleillä
- Lähiympäristön pesät: koordinaattikenttien yhteydessä olevasta painikkeesta käynnistetään funktio joka päivittää viereisen iframen sisällön antaen sille parametreiksi koordinaattikenttien arvot

- Estetään tarkastuslomakkeen lähettäminen enter-painikkeella. IE-selaimessa enter toimii tab-painikkeena.

Kaikki skriptit on toteutettu sivujen head -osiossa. Funktioita käyttävät eventit on määritelty HTML-elementeissä itsessään.

## 9 Toteutetut ominaisuudet ja tunnetut puutteet

Kaikki prioriteettien P1 ja P2 toiminnalliset vaatimukset on täytetty. Prioriteettien P3, P4 ja P5 ominaisuuksia ei ole toteutettu. Toteutetut ja puuttuvat ominaisuudet on listattu seuraavaksi. Tarkemmat tiedot löytyvät vaatimusdokumentista.

Toteutetut ominaisuudet:

- J1, Järjestelmä tuottaa tulostettavan tiedoston, jossa on esitetyt raportit tarkastajille
- J2, Järjestelmästä voi valita hakuehdoilla tietyt pesät
- J5, Paperilomakkeen tiedot voi syöttää järjestelmään
- J6, Pesän tietoja pitää pystyä hakemaan spatiaalisesti
- J10, Järjestelmän pitää tunnistaa jos yhdelle pesälle tehdään useampi tarkastus

Toteuttamatta jääneet ominaisuudet:

- J3, Pesän tarkastamisen voi siirtää toisen tarkastajan vastuulle
- J4, Tarkastajien sähköisen tarkastuslomakkeen tietojen hyväksyminen tietokantaan
- J4.1, Tarkastajien sähköisen tarkastuslomakkeen tietojen korjaaminen
- J5.1 Sähköisen lomakkeen arvot voi syöttää tietokantaan
- J7, Kuntatietojen yhdistäminen
- J8, Kunnan rajojen muokkaaminen
- J9, Pesistä voidaan tuottaa yhteenveto
- J11, Tietokantaan voi lisätä kuvan pesästä
- J12, Pesästä otettuja kuvia voi tarkastella

### 9.1 Esille tulleita ideoita jatkokehitykseen

Sääksijärjestelmää kehitettäessä esille tuli uusia ominaisuuksia, joiden toteuttaminen tulevaisuudessa voi olla hyödyllistä:

- Järjestelmän tulisi varmistaa, ettei kunnassa ole kahta saman nimistä reviiriä.
- Järjestelmästä pitäisi olla mahdollista poistaa pesä.
- Järjestelmästä pitäisi olla mahdollista poistaa reviiri.
- Reviirin nimen muokkaus.
- Aputietojen kuten kuntataulun, rengastajataulun ja aputaulujen tietojen muokkaus.
- Vuoden tarkastamattomien pesien haluttu toiminnallisuus tulisi selvittää ja toteuttaa. (Ajankohtaisempaa sitten kun kanta on siistimmässä kunnossa.)
- Tällä hetkellä pesän tarkastuksen tietojen muokkauksessa selataan tarkastuksia askel kerrallaan eteen tai taakse. Dropdown -valikko olisi kätevämpi.
- Käyttöliittymäkuvauksessa kuvattu haavi jäi toteuttamatta. Lähinnä on epäselvää mikä sen haluttu toiminnallisuus olisi.
- Tarkastustyökalut -sivun esitetyt lomakkeiden tulostustyökaluissa pitäisi olla rengastajista dropdown menu rengastajanumeron tekstikentän sijaan.
- Järjestelmän pitäisi kauttaaltaan käyttää unix timestamppeja. Lokalisaatio/päivämäärien parsiminen aiheuttaa ongelmia.
- Samaan kuntaan ei pitäisi pystyä lisäämään kahta samannimistä reviiriä.
- Pesähaun koordinaattien validointi vajaavainen (mahdollisuus hakea koordinaatteja väliltä

xxxxx-xxxxxx, yyyy-yyyy muuttaa validointia tarkastuksen syötön vastaavasta).

- Järjestelmään jääneet System.out.println -viestit pitäisi joko siivota pois tai muuttaa debug -viesteiksi, ja toteuttaa mahdollisuus kytkeä debug viestien tulostus päälle tai pois.
- Käyttöliittymä tuotettujen PDF-tiedostojen käyttämiseksi.