



Kasi

# Ylläpitodokumentti

Helsinki 6.5.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

## **Kurssi**

581260 Ohjelmistotuotantoprojekti (9+1op/6ov)

## **Projektiryhmä**

Lauri Holmas  
Veli-Pekka Kestilä  
Joni Lahtinen  
Tuukka Palomäki  
Markus Penttilä  
Antti-Pekka Sarin  
Ilkka Tikkala

## **Asiakas**

Harri Laine  
Petri Kutvonen

## **Vastuhenkilö**

Kimmo Simola

## **Ohjaaja**

Jaakko Saaristo, Ilari Moilanen

## **Kotisivu**

[www.cs.helsinki.fi/group/kasi](http://www.cs.helsinki.fi/group/kasi)

## **Versiohistoria**

| <u>Versio</u> | <u>Päiväys</u> | <u>Tehdyt muutokset</u>        |
|---------------|----------------|--------------------------------|
| 1.0           | 6.5.2007       | MP: Dokumentti julkaisukuntoon |

# Sisällysluettelo

|  |    |
|--|----|
| 1 Johdanto.....  | 1  |
| 2 Asennusohjeet.....                                       | 2  |
| 2.1 Ostoliittymä.....                                      | 2  |
| 2.2 Raporttiliittymä.....                                  | 3  |
| 3 Järjestelmän asetukset.....                              | 5  |
| 3.1 Yleisasetukset.....                                    | 5  |
| 3.2 Tietokannan kyselylauseet.....                         | 5  |
| 3.3 Monikielisuuden toteutus.....                          | 5  |
| 4 Ohjelmakoodin ulkoasu.....                               | 7  |
| 5 Virheensietokyky.....                                    | 8  |
| 5.1 Ulkoiset järjestelmät.....                             | 8  |
| 5.2 Tiedon eheys.....                                      | 8  |
| 5.3 Virheiden kirjaaminen.....                             | 8  |
| 6 Virheet.....   | 10 |
| 6.1 Luokkien pituus.....                                   | 10 |
| 6.2 XHTML-kielen oikeellisuus.....                         | 10 |
| 6.3 JavaScript.....  | 10 |
| 6.4 Tietokantayhteyksien virnehallinta.....                | 11 |
| 7 Puutteet.....  | 12 |
| 7.1 Tuotteiden yksikkökoko ei ole kolmella kielellä.....   | 12 |
| 7.2 Tyhjät tuoteryhmät ja tuoteryhmien järjestys.....      | 12 |
| 7.3 Kielen vaihtaminen ei säilytä järjestelmän tilaa.....  | 13 |
| 7.4 Asiakasliittymä ei näytä vietävän tuotteen hintaa..... | 13 |
| 8 Huomioita.....   | 14 |
| 8.1 Tuotteiden ja ryhmien deaktivointi.....                | 14 |
| 8.2 Käyttäjien poistaminen.....                            | 14 |
| 8.3 Testauksesta.....                                      | 14 |
| 9 Ulkopuoliset komponentit.....                            | 15 |
| 9.1 Griaule GrFinger Java SDK 4.5.....                     | 15 |
| 9.2 Substance.....   | 15 |
| 9.3 PostgreSQL JDBC-ajuri.....                             | 15 |
| 9.4 Oracle JDBC-ajuri.....                                 | 15 |
| 9.5 Log4j.....   | 16 |
| 9.6 JUnit.....   | 16 |
| 9.7 cURL.....  | 16 |
| 10 Kehitysideoita.....                                     | 17 |
| 10.1 Varastosaldon seuraaminen.....                        | 17 |
| 10.2 Käyttöliittymien skaalautuvuus.....                   | 17 |
| 10.3 Raportit.....   | 18 |
| 10.4 Monisäikeisyyden kehittäminen.....                    | 18 |
| 10.5 Käyttäjien tunnistaminen.....                         | 18 |
| 11 Lisenssitietoja.....                                    | 19 |

# 1 Johdanto

Tässä ylläpidodokumentissa kuvataan kummankin osajärjestelmän, Java-kielellä toteutetun, kahvihuoneeseen tarkoitetun asiakasliittymän, ja PHP-kielellä toteutetun verkkopalvelimen kautta käytettävän raportointiliittymän asennus ja asetusten säätö.

Lisäksi dokumentissa on otettu huomioon järjestelmän jatkokehittäjien mahdolliset tarpeet. Dokumentissa kuvataan miten järjestelmä on tuotettu, mitä työkaluja on käytetty ja millaisia ratkaisuja on projektissa tehty.

Dokumentissa pyritään myös antamaan jatkokehittäjille kehitysideoita ja selvittämään niistä mahdollisimman tarkkaan taustat, jotta idean toteuttaminen olisi mahdollisimman selkeä operaatio.

Ohjelmistossa olevat selkeät virheet ja puutteet on myös dokumentoitu mahdollisimman laajasti ja tarkasti.

## 2 Asennusohjeet

Tässä luvussa käsitellään ostoliittymän ja raporttiliittymän asennusta. Molemmista käsitellään vaadittavien asetustiedostojen muokkaamista valittuun käyttöympäristöön sopiviksi.

### 2.1 Ostoliittymä

Ostoliittymä voidaan asentaa kopioimalla CD-levyltä hakemiston *dist* sisältö haluttuun paikkaan kiintolevyllä. Lisäksi pitää hakea / ostaa Griaulen verkkosivuilta [www.griaule.com](http://www.griaule.com) Griaule GrFinger Java SDK 4.5, jonka asennushakemistosta pitää kopioida `grfingerjava.jar` asiakaskäyttöliittymän asennushakemiston alihakemistoon *lib*.

Windows-ympäristössä on lisäksi asennettava Griaulen sivuilta niin ikään löytyvä sormenjälkilukijan ajuri. Microsoftin sormenjälkilukijan kanssa käytetään FingerCap USB Driver -ajuria.

Mikäli halutaan käyttää Oracle-tietokantaa, on lisäksi haettava Oraclen verkkosivuilta JDBC-ajuri, jonka `ojdbc14.jar` on kopioitava samaiseen paikkaan.

Lopuksi on säädettävä kuntoon asetustiedosto *settings.xml*, joka löytyy alihakemistosta *settings*. Jokaisen asetuksen XML-tiedostossa on sijaittava `<entry>` `</entry>` -elementtien välissä.

Asetustiedostosta löytyvät seuraavat asetukset:

`<entry key="loglevel">` Lokituksen taso (Oletus INFO)

`<entry key="logpath">` Lokitiedoston sijainti

`<entry key="queryfile">` Tiedosto, josta löytyvät SQL-kyselyt.

Käytettävä `settings/dbqueries_postgres.xml` tiedostoa PostgreSQL-tietokannan kanssa ja `settings/dbqueries_oracle.xml` Oracle-tietokannan kanssa.

`<entry key="stringfile-fi">` Suomenkieliset tekstit. (Oletus `settings/finnish.xml`).

`<entry key="stringfile-se">` Ruotsinkieliset tekstit. (Oletus `settings/swedish.xml`)

`<entry key="stringfile-en">` Englanninkieliset tekstit. (Oletus `settings/english.xml`)

`<entry key="idletimeout">` Aika, jonka jälkeen käyttäjä kirjataan ulos, jos tämä ei tee mitään. Annetaan sekunteina.

`<entry key="paytimeout">` Aika, jonka jälkeen suosikkituotteen automaattinen veloitus tapahtuu. Annetaan sekunteina.

**<entry key="errortimeout">** Kuinka kauan virheilmoitus tunnistamattomasta sormenjäljestä näytetään ruudulla. Annetaan sekunteina.

**<entry key="databasedriver">** Käytettävä JDBC-tietokanta-ajuri. PostgreSQL kanssa käytettävä org.postgresql.Driver, Oraclen kanssa oracle.jdbc.OracleDriver

**<entry key="databaseuri">** Tietokannan osoite. PostgreSQL:n kanssa muotoa jdbc:postgresql://osoite:portti/kannan\_nimi Oraclella jdbc:oracle:thin:@osoite:portti:kannan\_nimi

**<entry key="dbuser">** Tietokannan käyttäjätunnus

**<entry key="dbpassword">** Tietokannan salasana

**<entry key="cancelTimeout">** Aika, jonka jälkeen käyttäjä ei voi enää peruuttaa edellistä vientitapahtumaansa (ostostaan). Annetaan sekunteina.

**<entry key="nativelibrarypath">** Griaule GrFingerin natiivikirjastopolku, oletuksena Griaulen asennushakemiston alihakemisto *bin*.

**<entry key="licensepath">** Polku hakemistoon, jossa on Griaule GrFingerin lisenssitiedosto. Oletuksena sama kuin natiivikirjastopolku.

**<entry key="authenticationurl">** Verkkosivu, jota käytetään käyttäjien tunnistautumiseen. Verkkosivun pitää käyttää Basic Auth -tunnistautumista ja tietoturvasyistä olla HTTPS-yhteyden takana. Tarkoituksena on käyttää jotakin tietojenkäsittelytieteen laitoksen "intranetissa" sijaitsevaa sivua (eli muotoa https://www.cs.helsinki.fi/i/username/sivu.html), jolloin autentikointi onnistuu laitoksen käyttäjätunnuksilla.

**<entry key="keystorepath">** Osoite Javan sertifikaattitiedostoon, johon on asennettu tietojenkäsittelytieteen laitoksen (tai muu tarvittava) sertifikaatti. (Oletus: mukana toimitettava keystore/HelsinkiJavaKeystore)

**<entry key="keystorepassword">** Sertifikaattitiedoston salasana. (Oletustiedoston salasana: kasikasi)

## 2.2 Raporttiliittymä

Raporttiliittymä vaatii toimiakseen www-palvelimen, joka suorittaa PHP-sivuja. Liittymän toimivuus on testattu vain TKTL:n alkokrunni-palvelimen ympäristössä, joten se toimivuutta muualla ei voida taata. Todennäköisesti se toimii useimmilla Linux-pohjaisilla Apachea käytävillä palvelimilla. Palvelimen on lisäksi tuettava HTTPS-yhteyksiä, koska raportointiliittymää on tarkoitus käyttää sellaisen kautta.

Raporttiliittymä asennetaan kopioimalla kaikki CD-levyn *web*-hakemistossa sijaitsevat tiedostot haluttuun asennuspaikkaan. Asennuspaikan on luonnollisesti syytä olla jokin, jonka www-palvelin näyttää ulospäin.

Laitoksen palvelimella, jossa PHP ei ole asennettu suoraan Apachen moduuliksi, on tie-

dostolle *php.cgi* on annettava käyttäjän luku- ja suoritusoikeudet ja kaikille muille ai-noastaan lukuoikeudet. *php.cgi*-tiedostossa on käytettävä UNIX-rivinvaihtoja. Lisäksi asennushakemistoon on luotava tiedosto **.htaccess**, jonka sisältö on seuraava:

```
AddHandler cgi-script cgi
AddHandler php-cgi php
Action php-cgi /polku/asennushakemistoon/php.cgi
```

Tiedosto on myös asennushakemistossa valmiina nimellä `htaccess_RENAME_TO_.htaccess`.

Lopuksi on säädettävä asetukset kuntoon. Perusasetukset sijaitsevat *settings*-alihakemiston *settings.xml*-tiedostossa. Tiedosto on samankaltainen kuin asiakasliittymässä, ja siinä on seuraavat asetukset:

**<entry key="dbServer">** Tietokantapalvelimen osoite. (Oletus: localhost)

**<entry key="dbName">** Tietokannan nimi, käytetään vain PostgreSQL:n kanssa.

**<entry key="dbUser">** Tietokannan käyttäjätunnus

**<entry key="dbPassword">** Tietokannan salasana

**<entry key="dbPort">** Tietokantapalvelimen portti

**<entry key="log\_dir">** Hakemisto, johon lokit kirjoitetaan. *Ei käytössä ohjelman lopullisessa versiossa.*

**<entry key="colSeparator">** Merkki, jota käytetään erottamaan sarakkeet kun raportti talletetaan tekstimuodossa. (Oletus: ;)

**<entry key="rowSeparator">** Merkkijono, jota käytetään rivin vaihtoon raportti talletettaessa. *Ei käytössä lopullisessa versiossa.*

**<entry key="authenticationUrl">** Verkkosivu, jota vasten autentikointi tehdään. Suositellaan käytettäväksi samaa sivua kuin asiakasliittymässä.

Oletuksena raporttiliittymä on säädetty käyttämään PostgreSQL-tietokantaa. Oracle-tietokantaa käytettäessä on tehtävä muutama muutos. *include* hakemistossa sijaitseva *connection\_oracle.php* on nimettävä uudestaan nimelle *connection.php*. Samoin *settings* hakemistossa sijaitseva tietokantakyselytiedosto *dbqueries\_oracle.xml* on nimettävä uudestaan nimelle *dbqueries\_oracle.xml*.

Lopulta on vielä vaihdettava *javascript*-hakemistossa sijaitseviin tiedostoihin *product.js* ja *usr.js* oikeat absoluuttiset polut käytettyjen PHP-tiedostojen osoitteisiin. Käytetyt osoitteet on määritelty kummankin tiedoston alussa. Oletusarvoisesti ne ovat muotoa [https://db.cs.helsinki.fi/~tkk\\_kasi/cgi-bin/tiedoston\\_nimi.php](https://db.cs.helsinki.fi/~tkk_kasi/cgi-bin/tiedoston_nimi.php). Näistä siis ennen tiedoston\_nimi.php oleva osa on muutettava käytetyn polun mukaiseksi.

## 3 Järjestelmän asetukset

Sekä asiakaskäyttöliittymä, että raportointiliittymä käyttävät useita XML-muotoisia asetustiedostoja, joiden avulla järjestelmän toimintaa ja erikielisiä tekstejä voidaan säätää. Asetustiedostot sijaitsevat juurihakemistojen alla *settings*-hakemistossa.

### 3.1 Yleisasetukset

Kumpikin osajärjestelmä sisältää oman *settings.xml*-tiedostonsa, jossa määritellään mm. tietokannan osoite, käyttäjätunnukset ja autentikointiin käytettävä verkkosivu. Asiakasliittymässä voidaan tämän tiedoston kautta lisäksi määritellä muiden asetustiedostojen nimi ja sijainti. Näiden tiedostojen rakenne on kuvattu edellisessä luvussa.

### 3.2 Tietokannan kyselylauseet

Tietokantaoperaatioihin käytetyt SQL-lauseet on kummassakin osajärjestelmässä määritetty tiedostossa *dbqueries.xml*. Näin mahdollistetaan käytettyjen kyselyjen helppo muokattavuus, mikäli taustalla toimivaa tietokantaa vaihdetaan. Ohjelman mukana toimitetaan PostgreSQL- ja Oracle-tietokannoille sopivat tietokantalauseet.

Raportointiliittymän puolella käytetyn tietokannan vaihtaminen vaatii lisäksi *connection.php*-tiedostossa sijaitsevan *Connection*-luokan muokkaamista sopivaksi. Ohjelman mukana toimitetaan kummallekin yllä mainitulle tietokannalle soveltuva *Connection*-luokka. Käytössä olevan tiedoston nimi on muutettava *connection.php:ksi*.

### 3.3 Monikielisyyden toteutus

Järjestelmä päätettiin toteuttaa tukemaan suomea, ruotsia ja englantia. Asiakas- ja raportointiliittymän näyttämät tekstit sijaitsevat kielten mukaan nimetyissä XML-tiedostoissa, asiakasliittymässä *finnish.xml*, *english.xml* ja *swedish.xml*, raportointiliittymässä *fin.xml*, *eng.xml* ja *swe.xml*. Näitä tekstejä voidaan haluttaessa vapaasti muokata. Kummassakin osajärjestelmässä on omat apuluokkansa, jotka hakevat tekstit näistä tiedostoista teksteille annettujen nimien perusteella.

Käyttöliittymiin olisi sinänsä kohtuullisen yksinkertaista lisätä muitakin kieliä, mutta suurimman rajoituksen asettaa tietokanta, jossa tuotteiden yms. nimet sijaitsevat aina kolmessa näiden kielten mukaan nimetyssä sarakkeessa. Tällaiseen kiinteäähköön kolmen kielen ratkaisuun päädyttiin, koska todettiin, että uuden kielen lisäyksen myötä ongelmaksi tulisi syöttää kaikille järjestelmän tuotteille nimet myös uudella kielellä.

Englannin- ja ruotsinkielisiä tekstejä ei ole tarkastettu näiden kielten ammattilaisilla tai niitä äidinkielekseen puhuvilla ihmisillä. Etenkin ruotsinkielisissä teksteissä saattaa



esiintyä kielioppivirheitä, ja jotkin raportointiliittymän ohjetekstit on jätetty kokonaan toteuttamatta ruotsiksi ajanpuutteen vuoksi. Ne voidaan kuitenkin helposti lisätä tiedostoon swe.xml.

## 4 Ohjelmakoodin ulkoasu

Järjestelmässä on sekä Java- että PHP-kielellä toteutettua koodia. Molemmista kielistä oli pienimuotoinen ohjelmointiopas saatavilla ohjelmoijille.

Java-koodissa on käytetty normaalia Java-tyylistä niin sanottua CamelCase-nimeämistä niin muuttujissa, metodeissa kuin luokkien nimissäkin. Muuttujat on pyritty nimeämään niin kuvaavasti, ettei niiden käyttötarkoitus jää epäselväksi. Saman luokan muuttujiin ja metodiin viitattaessa on käytetty *this*-määrettä ymmärrettävyyden parantamiseksi.

PHP-koodissa on käytetty paljon samankaltaista nimeämistapaa kuin Java-koodissakin, mutta lisäksi niin sanottua unkarilaista notaatiota, eli muuttujien eteen on lisätty tarkoitettun tyyppin mukaisesti s\_ (string), i\_ (int), c\_ (class), b\_ (boolean) ja niin edelleen. Metodien ja luokkien nimet on kirjoitettu pääasiassa Java-suositusten mukaisesti.

Koodirivin maksimipituutena on pyritty pitämään 80 merkkiä, mutta tämän tarkistamisesta on ajanpuutteen vuoksi jouduttu PHP-koodissa joustamaan.

Ohjelmakoodissa käytetty luonnollinen kieli on englanti, niin nimeämisen kuin kommentoinninkin osalta.

# 5 Virheensietokyky

Tässä luvussa kerrotaan miten järjestelmä on varautunut mahdollisiin virheisiin.

## 5.1 Ulkoiset järjestelmät

Järjestelmän suunnittelussa ja/tai toteutuksessa ei ole otettu huomioon kaikkia mahdollisia virhetilanteita ja ympäristön ongelmia.

Eräs potentiaalinen ongelma on hidas yhteys tietokantaan tai sormenjälkitunnistukseen. On täysin mahdollista, että käytetty tietokantapalvelin on liiallisen kuorman tai hitaan yhteyden vuoksi liian hidas asiakasliittymälle. Sormenjälkitunnistus taas vaatinee suurilla sormenjälkimäärillä laitteistolta laskentatehoa, jolloin heikkotehoisella koneella tunnistaminen saattaa kiusallisesti hidastua.

Etenkin tietokantayhteyden heikkoudesta voi olla seurauksena asiakaskäyttöliittymän hyytyminen pitkiksi ajoiksi ja pahimmillaan jopa sen kaatuminen. Normaalisti toimivalta tietokantayhteydellä ongelmia ei kuitenkaan pitäisi esiintyä.

Parannus olisi laittaa kaikki toiminnot, jotka mahdollisesti voivat viedä kauan aikaa erillisiin säikeisiin, joissa ne palvelisivat käyttäjää antamalla statustietoa järjestelmän saamista palautteista ulkopuolisilta komponenteilta. Näin graafisen käyttöliittymän toiminnot eivät suoraan jumiutuisi yhteysongelmien vuoksi. Tätä ei kuitenkaan ajanpuutteen vuoksi ehditty enää toteuttamaan.

## 5.2 Tiedon eheys

Järjestelmä ei itse pidä minkäänlaista versiotietoa omasta datastaan, vaan tietokannan ylläpitäjän on huolehdittava mahdollisista varmuuskopioinneista ja tiedon palauttamisesta.

Mikäli jatkokehittäjät näkevät tähän tarvetta, valmiita ratkaisumalleja ei valitettavasti osata tarjota.

## 5.3 Virheiden kirjaaminen

Virheet kirjataan Javalla toteutetussa asiakaskäyttöliittymässä log4j-järjestelmällä loki-tiedostoon. PHP:llä toteutetussa raportointiliittymässä virheitä ei kirjata mihinkään, koska loggausta ei ajanpuutteen vuoksi ehditty toteuttaa kunnolla.

PHP-tiedostoissa sinänsä on helppo asettaa virheiden näyttäminen päälle kirjoittamalla tiedoston alkuun `ini_set("display_errors", true)`, jolloin normaalin PHP-sivun lomaan

tulostetaan ajonaikaiset virheet ja varoitukset.

# 6 Virheet

Tarkoista vaatimusmäärittelyistä ja suunnitelman tarkastuksista huolimatta projektin tuotokseen on jäänyt joitakin virheitä. Virheet pyritään selostamaan tarkasti ja niiden korjaamiseen esitetään ratkaisumalleja.

## 6.1 Luokkien pituus

Jotkin luokat ovat ylittäneet projektissa hiljaisesti määritetyn maksimipituuden. Maksimipituus on jäänyt näkymättömäksi määritelmäksi ohjelmoijille, koska aika ei ole riittänyt pituuksien korjaamiseen ja toisaalta toteutusvaiheessa luokkiin on jouduttu lisäämään uutta toiminnallisuutta. Tähän ratkaisuun on päädytty, koska valmista tuotetta on pidetty tärkeämpänä kuin jatkokehityksen helppoutta.

Luokkien, joiden pituus on päässyt huomattavan suureksi, korjaamiseksi kannattanee eritellä niiden toteuttamat rajapinnat omiksi luokikseen. Tämä olisi varmasti yksinkertaisin vaihtoehto etenkin graafisen käyttöliittymän GuiMain-pääloukan osalta.

Tietokantaluokan (DBCommunication) kohdalla sen toteuttamaa toiminnallisuutta voisi jakaa useammille aliluokille, vaikka rajapinnan toteuttaminen säilyttäisiinkin kokonaisuudessaan pääloukassa.

Myös PHP-puolella on muutama suhteellisen pitkä luokka, jotka voisi käyttötarpeen mukaan jakaa useiksi luokiksi.

## 6.2 XHTML-kielen oikeellisuus

Raportointiliittymän sivut on pyritty tekemään XHTML 1.0 Transitional -standardin mukaiseksi. Ajanpuutteen vuoksi tähän ei ole kuitenkaan ainakaan kaikkien sivujen osalta päästy, eli sivut eivät täytä standardin vaatimuksia täydellisesti. Kohdeselaimilla testatessa tämän ei ole kuitenkaan huomattu vaikuttavan sivunmuodostukseen.

Pelkästään PHP-kielellä tuotetut sivut on suhteellisen helppo korjata standardin mukaiseksi. Tuote- ja käyttäjäsiivuilla osa koodista kuitenkin generoidaan JavaScriptin avulla, jolloin jo pelkästään sen tarkastaminen on paljon vaikeampaa, ja korjaaminen vaatii muutosten tekemistä JavaScript-tiedostoihin.

## 6.3 JavaScript

JavaScript-kielen toteutus vaihtelee jonkin verran selaimittain. Koska järjestelmä on suunniteltu käytettäväksi Mozilla Firefoxilla ja Internet Explorer 7:lla, on JavaScript-tiedostot pitänyt tehdä toimiviksi näillä kummallakin. Tässä ei ole kuitenkaan täysin on-

nistuttu, vaan sivuilla esiintyy pieniä ulkoasuongelmia. Lisäksi käyttäjän poistaminen ei toimi Internet Explorerilla.

Kaiken kaikkiaan raportointiliittymä toimii oikein Firefoxilla, johtuen kyseisen selaimen paremmasta tuesta standardeille, ja toisaalta projektin ohjelmoijien Linux-käytöstä, jolloin Firefox on ollut pääasiallinen testausselein. Lienee kuitenkin mahdollista korjata JavaScript-tiedostot toimimaan oikein myös Internet Explorerilla.

## **6.4 Tietokantayhteyksien virnehallinta**

Alkuperäisen suunnitelman mukaan asiakaskäyttöliittymän pitäisi siirtyä erityiseen virhetilaan, jossa käyttäjän ei sallita suorittaa mitään toimintoja, jos tietokantayhteys menee poikki kesken tietokantaoperaation. Virhetilasta ohjelman pitäisi palautua normaaliin tilaan, jos yhteys saadaan palautettua takaisin toimintakuntoon.

Tätä ominaisuutta ei koskaan kuitenkaan ehditty toteuttaa luotettavasti, vaan tietokantayhteyden ongelmat kirjataan järjestelmän lokiin. Käyttöliittymässä ominaisuus on kuitenkin olemassa, joten sopivat poikkeukset sieppaamalla ja käyttöliittymän metodeita kutsumalla virhetila saataneen toimimaan suunnitellun mukaisesti.

# 7 Puutteet

Järjestelmään jäi myös puutteita. Jokaisen puutteen kohdalla kerrotaan millaisesta puutteesta on kysymys ja selvitetään miten puute on korjattavissa.

## 7.1 Tuotteiden yksikkökoko ei ole kolmella kielellä

Tuotteiden koon yksikköä (siis kg, kpl jne.) ei ole järjestelmässä tuettu kolmella kielellä, mikä voi olla ongelma, jos järjestelmässä halutaan pitää tuotteita, joiden määrään lyhenne on erilainen eri kielillä. Tällainen on lyhenne on esimerkiksi kappale.

Puutteen korjaaminen aiheuttaisi muutoksia sekä raporttiliittymässä että ostoliittymässä. Muutoksia tulisi tietenkin tehdä myös tietokantaan.

Lähtökohtana on, että lisätään rawproduct-tauluun nimien kaltaiset sarakkeet raw\_sizeunit sarakkeelle (raw\_finsizeunit, raw\_swesizeunit ...). Näin tietokantaan saadaan tarvittavat tiedot.

Asiakasliittymässä tulisi muokata kyselyä, jolla rawproduct-tilusta haetaan raw\_sizeunit. Lisäksi erikieliset koot pitäisi lisätä ImportProduct-tieto-olioon ja korjata myös DBCommunication-luokka luomaan olio uudella tavalla ja vastaavasti GUI:n luokat hakemaan tieto oliosta oikealla kielellä.

Raportointiliittymässä pitäisi tuotetietojen muokkaukseen lisätä muutama uusi kenttä ja toisaalta säätää tuotettavat raportit hakemaan koon yksikkö kulloinkin valitulla kielellä, kuten tällä hetkellä tehdään tuotteiden nimien kanssa.

## 7.2 Tyhjät tuoteryhmät ja tuoteryhmien järjestys

Asiakaskäyttöliittymässä esitetään turhaan käyttäjälle myös tuoteryhmät, joihin ei ole liitetty ainuttakaan tuotetta. Tämä ei onneksi ole suuri puute, sillä ylläpitäjä voi helposti poistaa raportointiliittymän kautta tällaiset tuoteryhmät käytöstä, jolloin niitä ei enää esitetä.

Toinen ongelma on, että tuoteryhmiä ei asiakaskäyttöliittymässä järjestetä mihinkään tiettyyn järjestykseen, vaan ne esitetään ruudulla siten kuin ne tietokannasta sattuvat tulemaan. Pienellä ryhmämäärällä tämä ei ole ongelma, mutta suuremmalla se saattaa sitä ollakin.

Kumpikin näistä puutteista olisi korjattavissa asiakaskäyttöliittymän tietokantauseiden ja tietokantaluokan pienellä muokkauksella.

### **7.3 Kielen vaihtaminen ei säilytä järjestelmän tilaa**

Sekä asiakas- että raportointiliittymässä kielen vaihtaminen luo kulloisenkin sivun kokonaan uudestaan, jolloin se palautuu alkutilaansa. Olisi parempi, että kulloinenkin tila säilyisi samanlaisena kieli vaihdettaessa, mutta tämän toteuttaminen olisi vaatinut huomattavasti hankalampien ratkaisujen kehittämistä kummassakin osajärjestelmässä, joten siitä luovuttiin. Puutteen korjaamiseksi ei myöskään osata suoraan esittää ratkaisuehdotuksia.

### **7.4 Asiakasliittymä ei näytä vietävän tuotteen hintaa**

Järjestelmä tukee muilta osiltaan sitä, että vietäville tuotteille on määritelty ”hintaa”, eli että ne voivat veloittaa käyttäjän tiliä eri määrän. Ostoliittymässä ei kuitenkaan näytetä vietävän tuotteen hintaa silloin kun jokin tuote on valittu, koska alun perin oletettiin vietävien tuotteiden olevan aina ”yhden kupin” arvoisia. Jos hinta siis onkin jotain muuta kuin yksi, se tulee käyttäjälle yllätyksenä.

Tämä on toki varsin helppo korjata muokkaamalla hieman käyttöliittymän ostonäkymän luokkien koodia.



# 8 Huomioita

## 8.1 Tuotteiden ja ryhmien deaktivointi

Järjestelmä ei tarjoa laisinkaan mahdollisuutta poistaa tuotteita (tai tuoteryhmiä) järjestelmästä kokonaan. Jos siis tuoteryhmiä jostakin syystä luotaisiin kovin paljon, ja ne päädyttäisiin poistamaan käytöstä deaktivoimalla ne, ne voivat lopulta tehdä tuotesivusta varsin pitkän, sillä kaikki passivoidutkin tuoteryhmät näytetään listassa niiden palauttamisen mahdollistamiseksi.

Passivoidut tuoteryhmät eivät ilmesty kaikkiin raportteihin, joten jos halutaan raportti, joka sisältää kaikki mahdolliset ryhmät, pitää ryhmät ensin aktivoida.

## 8.2 Käyttäjien poistaminen

Kun järjestelmästä poistetaan käyttäjä, poistetaan tietokannasta myös kaikki tämän sormenjäljet ja saldotiedot. Käyttäjän tapahtumat jätetään kuitenkin järjestelmän lokiin, mistä syystä on jouduttu jättämään tietokannasta pois viite-eheyden säilytys lokiin kirjattun käyttäjän ja käyttäjätaulussa olevien käyttäjätunnusten välillä.

## 8.3 Testauksesta

Asiakaskäyttöliittymän yksikkötestaus on päätetty jättää tekemättä GUI- ja Main-komponenteille. Graafisen käyttöliittymän toteuttavat GUI:n luokat soveltuvat huonosti yksikkötestaukseen, koska ne koostuvat lähinnä näkyvät luovasta Swing-koodista. Main-komponentti taas toimii lähinnä viestien välittäjänä järjestelmän muiden komponenttien välillä, joten sen yksikkötestaus olisi varsin hyödytöntä. GUI- ja Main-komponenttien testaus on tullut riittävän kattavasti suoritettua järjestelmätestauksen yhteydessä.

PHP:llä toteutetun raportointiliittymän osalta yksikkötestaus on jouduttu jättämään kokonaan pois, joten se on sille on suoritettu vain järjestelmätestaus. PHP-luokille oli suunniteltu tehtäväksi testauksen lausekattavuuden selvittäminen käyttämällä erillistä Logger-luokkaa, jonka avulla olisi saatu selville missä lohkoissa on käyty. Tämä koodi on kuitenkin poistettu lopullisista versioista, koska yksikkötestausta ei ehditty suorittaa ja toisaalta raportointiliittymän eri käyttötapaukset läpi käymällä tulee melko suoraviivaisesti selville, että luokkien metodit toimivat.

Tarkemmat tiedot testauksesta löytyvät testausraportista.

## 9 Ulkopuoliset komponentit

Asiakaskäyttöliittymässä käytetään useita ulkopuolisia komponentteja, jotka on esitelty seuraavassa.

### 9.1 Griaule GrFinger Java SDK 4.5

Griaulen (<http://www.griaule.com>) kaupallista sormenjälkitunnistuskomponenttia käytetään järjestelmässä sormenjälkien tunnistamiseksi. Lisenssisyistä kyseistä komponenttia ei voida levittää tämän ohjelman mukana, vaan käyttäjän on haettava se edellä mainituilta verkkosivuilta ja asennettava koneelleen. Tämän jälkeen on kopioitava asennushakemistosta `grfingerjava.jar` asiakasliittymän asennushakemiston alihakemistoon `lib` ja asetettava `settings.xml`:ssä Griaulen kirjastopolku osoittamaan Griaulen asennushakemiston alihakemistoon `bin`. Myös lisenssipolku on säädettävä kuntoon.

GrFingeria voi käyttää epäkaupallisesti kokeilulisenssillä 90 päivän ajan.

### 9.2 Substance

Substance Java Look & Feel -kirjastoa (<https://substance.dev.java.net/>) käytetään käyttöliittymän ulkoasun luomiseen. Kyseessä on alustariippumaton BSD-lisenssoitu apukirjasto. Se toimitetaan ohjelman mukana. BSD-lisenssi on luettavissa osoitteessa <http://www.opensource.org/osi3.0/licenses/bsd-license.php>.

### 9.3 PostgreSQL JDBC-ajuri

PostgreSQL:n JDBC-ajuri toimitetaan valmiiksi ohjelman mukana. Se mahdollistaa tietokantayhteyksien ottamisen PostgreSQL-tietokantaan. Ajuri on BSD-lisenssoitu.

### 9.4 Oracle JDBC-ajuri

Oraclen JDBC-ajuria ei lisenssisyistä voida levittää ohjelman mukana. Käyttäjän on hankittava ajuri tarvittaessa itse Oraclen verkkosivujen kautta ja kopioitava `ojdbc14.jar` asiakasliittymän asennushakemiston alihakemistoon `lib`. Ajuri mahdollistaa tietokantayhteyksien ottamisen Oracle-tietokantaan.

## 9.5 Log4j

Log4j-lokikirjastoa käytetään asiakasliittymän virhelokin kirjoittamiseen. Se toimitetaan ohjelman mukana ja on Apache Licensen alainen. Lisenssi löytyy osoitteesta <http://www.apache.org/licenses/LICENSE-2.0>.

## 9.6 JUnit

Yksikkötestien suorittamiseen käytetään JUnit-kirjastoa, joka tulee toimitetaan ohjelman mukana. JUnit on Common Public Licensen alainen (<http://junit.sourceforge.net/cpl-v10.html>).

JUnitin lähdekoodi on saatavissa osoitteesta <http://junit.org/>.

CPL:n ehtojen mukaisesti JUnitin tekijät eivät anna minkäänlaisia takuita ohjelmiston toiminnasta eivätkä ole vahingonkorvausvelvollisia ohjelmiston mahdollisesta väärästä toiminnasta.

## 9.7 cURL

Raportointiliittymä suorittaa käyttäjätunnusten autentikoinnin hyödyntäen useimmista Linux-asennuksista löytyvää cURL-ohjelmaa. Kyseisen ohjelman löytyminen palvelinkoneen polkumuuttujasta on siis välttämättömyys autentikoinnin toimivuudelle. Näin on jouduttu toimimaan, koska vaatimuksiimme on kuulunut käyttäjien autentikointi tietojenkäsittelytieteen laitoksen käyttäjätunnuksilla, mutta meille ei ole tarjottu mitään suoraa mahdollisuutta autentikoinnin suorittamiseen.

# 10 Kehitysideoita

Kehitysideat ovat järjestelmän lisätoimintoja tai ominaisuuksia, jotka voisivat olla asiakkaalle hyödyllisiä, mutta joita ei kuitenkaan ole ehditty tai haluttu toteuttaa projektin puitteissa järjestelmän muiden toimintojen lisäksi.

## 10.1 Varastosaldon seuraaminen

Tällä hetkellä jokaiseen vietävään tuotteeseen voi liittää tuotavia tuotteita raporttiliittymän kautta. Liittäessä ei kuitenkaan voi määrittää paljonko yhden tuotteen vienti kuluttaa jotakin tuotavaa tuotetta, joka siihen on liitetty (esim. kahvikupillinen kahvijauhetta). Ominaisuutta ei kuitenkaan päätetty toteuttaa kokonaisuudessaan, koska epäiltiin ettei varastokirjanpidosta saataisi kuitenkaan millään tasolla tarkkaa tällä tavalla.

Ajan kuluessa kulutussuhteet voisi kuitenkin saada säädettyä sen verran hyvin kohdalleen, että ominaisuudesta saattaisi olla oikeasti hyötyäkin, joten järjestelmässä päätettiin kuitenkin säilyttää mahdollisuus tuotteiden liittämiseksi.

Ominaisuuden täydentämiseksi tarvitaan tauluun, jossa vietävä tuote liitetään tuotaviin tuotteisiin myös määrän kertova kenttä. Samalla on lisättävä tuontituotteiden tauluun niiden varastosaldon kertova kenttä, jota sopivasti voidaan vähentää ja lisätä vientien ja tuontien mukaisesti.

Ominaisuuden hallintaan olisi myös hyvä tehdä käyttötapaus raporttiliittymään, missä järjestelmän ylläpitäjät voisivat muuttaa lopputuotteen ominaisuuksia kulutuksen kannalta. Lisäksi oikeiden kulutussuhteiden hakemiseksi olisi mahdollisuus kehittää joitakin raportteja, joissa laskettaisiin paljonko tuontituotetta on kulunut suhteessa vienteihin.

## 10.2 Käyttöliittymien skaalautuvuus

Ostoliittymä suunniteltiin alun perin tukemaan 800 x 600 pikselin resoluutiota. Sittemmin selvisi, että kohdelaitteiston kosketusnäyttö käyttää 1280 x 1024 pikselin resoluutiota, joten käyttöliittymän ulkoasua päivitettiin sen mukaisesti. Lopullinen liittymä vaatii vähintään 1024 x 768 pikselin resoluution ja näyttäneen parhaimmalta 1280 x 1024 pikselin resoluutiolla.

Raporttiliittymä tuottaa noin 1000 pikselin levyisiä sivuja, joten sekin vaatii parhaimman käytettävyyden saavuttamiseksi vähintään 1024 x 768 pikselin resoluution käyttämistä.

Etenkin ostoliittymän skaalautuvuutta eri resoluutioille voisi vielä kohentaa, sillä ulkoasu muuttuu turhan paljon käytetyn resoluution mukaan. Tätä ei ajanpuutteen vuoksi ehditty hioa täysin kuntoon.

### 10.3 Raportit

Järjestelmästä tulostettaville raporteille asetetut vaatimukset säilyivät koko projektin ajan hieman epämääräisinä, joten erilaisten raporttien ideointi jäi paljolti projektiryhmän vastuulle.

Järjestelmään voisi olla mahdollista kehittää toiminto, jolla voidaan luoda vapaamuotoinen raportti tietokantaan tallennetun tiedon mahdollistamissa rajoissa. Yksinkertaisimmillaan tämän voisi toteuttaa sallimalla käyttäjän syöttää vapaamuotoisia SQL-kyselyitä ja hieman hankalemmin luomalla sivun, josta voidaan valita halutut kentät ja filterin käyttöliittymän toimintojen kautta.

Myös valmiita raportteja voisi kehittää vielä lisää, ja niihin voisi liittää haluttaessa erilaisia graafisia kuvaajia tai muita mielenkiintoisia toimintoja.

### 10.4 Monisäikeisyyden kehittäminen

Kuten aiemmin mainittiin, voi järjestelmä tietyissä tilanteissa lamaantua, jos graafisen käyttöliittymän säie juuttuu suorittamaan pitkällistä tietokanta- tai muuta operaatiota. Tämän estämiseksi olisi mahdollista lisätä esimerkiksi Javan SwingWorker-luokan säikeitä, jotka suorittaisivat pitkät operaatiot käyttöliittymäelementeistä riippumatta.

### 10.5 Käyttäjien tunnistaminen

Koska projektin vaatimukseen kuului, että käyttäjät tunnistautuvat järjestelmään Helsingin yliopiston tietojenkäsittelytieteen laitoksen myöntämällä käyttäjätunnuksilla, mutta tätä varten ei tarjottu mitään virallista tapaa, jouduimme kehittämään oman hieman erikoisen virityksemme. Käyttäjät tunnistetaan siten, että järjestelmä ottaa yhteyttä TKTL:n käyttäjätunnuksilla suojatulle verkkosivulle, ja mikäli sinne kirjautuminen onnistuu, toteaa järjestelmä käyttäjätunnukset oikeiksi.

Tästä seuraa tarpeettoman monimutkaisuuden lisäksi se ongelma, että järjestelmään ei voi kirjautua käyttäjätunnuksilla (sormenjäljellä tunnistautuminen kyllä toimii), jos käytetty verkkopalvelin on alhaalla. Lisäksi raportointiliittymässä joudutaan käyttämään edellä kuvattua cURL-ohjelmaa apuna.

Kehitysideana siis käyttäjien tunnistamisen voisi tulevaisuudessa toteuttaa paremmin, esimerkiksi LDAP-palvelimen tai jonkin muun vastaavan avulla.

# 11 Lisenssitietoja

## CoffeeTouch (ostoliittymä ja raporttiliittymä):

Copyright © 2007 Helsingin yliopisto & Veli-Pekka Kestilä

## Substance-kirjasto:

Copyright (c) 2007, Kirill Grouchnikov

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## PostgreSQL JDBC-ajuri:

Copyright (c) 1997-2005, PostgreSQL Global Development Group  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the PostgreSQL Global Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**Log4j-kirjasto:**

Copyright 1999-2005 The Apache Software Foundation.

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.