Ohjelmistotuotantoprojekti, ylläpitodokumentti

Ryhmä Muppett

Helsinki 4.9.2008

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi:
Ohjelmistotuotantoprojekti, kesä 2008

Projekti:
Muutos- ja korjauspyyntöjen priorisointityökalu

Asiakas:
Oodi-konsortio/ Sampo Lehtinen

Ryhmä:
Arto Chydenius
Laura Haverinen
Merja Lindén
Topi Musto
Laura Ojala
Toni Sormunen

Ohjaaja:
Marko Lehtimäki

Dokumentin versiohistoria

Versio	Päiväys	Muutokset	Muuttaja
0.0	09.06.2008	Pohja	LH
0.1	01.09.2008	Sisällön siirto	LH
1.0	04.09.2008	Loput korjaukset	Muppett

Sisältö

1 Johdanto

Tässä dokumentissa on tarkoitus käsitellä Muppett-ohjelman ylläpitoon ja jatkokehittämiseen liittyviä kohtia. Dokumentti on suunnattu asiakkaan edustajille ja tuleville tässä dokumentissa käsiteltävän ohjelman kehittäjille.

Tämä ylläpitodokumentti on tarkoitettu luettavaksi yhdessä 31.7.2008 laaditun suunnitteludokumentin sekä ohjelmakoodin kanssa, ja siksi tässä dokumentissa on runsaasti viitteitä näihin kyseisiin dokumentteihin.

1.1 Yleistä ohjelman toteutuksesta ja dokumentin sisällön kuvaus

Ohjelma on tuotettu Helsingin yliopiston tietojenkäsittelytieteen laitoksen kesällä 2008 järjestämän Ohjelmistotuotanto -kurssin opiskelijoiden harjoitustyönä. Prosessimallina on käytetty vesiputousmallia, joka on valittu ensisijaisesti pedagogisista syistä. Prosessin jokaisessa vaiheissa tuotettiin siihen liittyvää dokumentaatiota. Tämä dokumentti on tuotettu toteutusvaiheen lopussa ja tähän on ensisijaisesti pyritty kokoamaan sellaiset toteutusvaiheessa esille tulleet kohdat, jotka poikkeavat suunnitteludokumentista.

Tämä ylläpitodokumentti koostuu seuraavista luvuista: Luku 1 johdanto kuvaa projektin aikana tuotettuja dokumentteja ja toteutusympäristöä, ohjelmia ja ohjelmointikieliä. Luku 2 sisältää ohjelman asennusohjeen. Luku 3 täsmentää ohjelmasta laadittua suunnitteludokumenttia. Siihen on pyritty keräämään sellaiset ohjelman toteutuksessa olevat kohdat, jotka poikkeavat suunnitteludokumentissa esitetyistä kohdista. Luvussa 4 esitetään ne ohjelmalle esitetyt vaatimukset ja suunnitelman kohdat, jotka jäivät toteuttamatta. Luku 5 käsittelee ohjelmakoodin ylläpitoon vaikuttavia asioita, luku 6 sisältää ohjelmassa havaittuja virheitä ja lopuksi luku 7 kokoaa muut ylläpitoon vaikuttavat seikat, joita ei muissa tätä edeltäneissä luvuissa ole käsitelty.

1.2 Ohjelman dokumentaatiosta

Suunnitteludokumentti on tarkastettu asiakkaan edustajan, ohjaajan ja kaikkien Muppett-ryhmäläisten läsnä ollessa 5.8.2008 järjestetyssä tarkastustilaisuudessa. Suunnitteludokumentti hyväksyttiin 7.8.2008 tehtyjen korjauksien jälkeen asiakkaan ja Muppett-ryhmäläisten toimesta.

Muita ohjelmaan liittyviä dokumentteja:

- Vaatimusmäärittelydokumentti: Ohjelman tarkoitus, toteutusvälineet ja priorisoidut käyttötapaukset.
- Testaussuunnitelma: Suunnitelma testauksen tasosta ja kuvaukset.
- Testausraportti: Kuinka järjestelmä on testattu
- Käyttöohjeet: Äänestäjälle ja ylläpitäjälle laadittu omansa siitä, kuinka järjestelmää käytetään.
- Yhteenvetodokumentti: Tiivistelmät kaikista projektin aikana laadituista dokumenteista.

Ohjelman jatkokehittämistä ajatellen myös yllämainittua dokumentaatiota kannattaa hyödyntää.

1.3 Toteutuksessa käytetyt ympäristöt, ohjelmat ja kielet

Järjestelmä on toteutettu ja testattu suunnitteludokumentin lukujen 3.1 ja 3.2 mukaisesti tietojenkäsittelytieteen laitoksen db-palvelimella.

Järjestelmän toimintalogiikka on toteutettu Java-kielen versiolla 1.6 ja ohjelman tietovaraston toteutuksessa käytetään PostgreSQL-ohjelmaa.

Järjestelmän käyttöliittymässä hyödynnetään XHTML 1.0 -merkkauskieltä ja ulkoasu määritellään pääasiassa CSS 2.1 -tyylimäärittelyillä, muutama kohta noudattaa CSS 3.0 standardia. Toiminnallisuutta on lisätty JavaScript-kielellä.

Tomcat 5.5.7 palvelinohjelmiston tehtävä on välittää järjestelmän käyttöliittymä käyttäjän selaimelle. Käyttöliittymä liitettiin toimintalogiikkaan JavaServlet- ja JSP-tekniikoiden avulla.

JSP-sivuilla käytettävä Java on versiota 1.4, koska tomcat 5.5.7 java-kääntäjä käyttää tätä. Tämän javaversion osuus on kuitenkin varsin pieni.

2 Asennusohje

2.1 PostgresSQL:n asetustiedostot

Asenna PostgreSQL tietokanta sen omien asennusohjeiden mukaisesti ja luo sille käyttäjätunnus tkt_mupp. Luo tietokanta ”muppett” ja aseta sille omistajaksi tietokannan käyttäjä ”tkt_mupp” seuraavasti:

createdb muppett -O tkt_mupp

Tietokannan taulut on luotu liitteessä 1 esitetyllä tavalla.

Tietokannan asetustiedostoihin ei ole tehty muutoksia. Kuitenkin on syytä huomata, että tietojenkäsittelytieteen laitoksen asennus voi poiketa PostgreSQL:n oletusasetuksista paljonkin.

Tietokannan ja Tomcat-palvelimen pitää käyttää samaa merkistön encodingia, jotta salasanan käsittelyt toimivat oikein.

2.2 Vaadittavat ulkoiset kirjastot

Tässä luvussa on kerrottu Muppett-järjestelmän vaatimat ulkoiset kirjastot. Kustakin kirjastosta on esitetty suluissa vaadittavat jar-paketit.

Tomcat- palvelinohjelmiston käyttö vaatii oman kirjastopakettinsa.(servlet-api.jar)

Sähköpostien lähettämiseen käytetään JavaMail- kirjastoa.(activation.jar, mail.jar)

Muutospyyntöjen tuonnissa käytetään Apache Commons FileUpload- komponenttia. (commons-fileupload-1.2.1.jar, commons-io-1.4.jar).

Jos muutospyyntöjä halutaan tuoda järjestelmään komentorivin kautta, käytettävissä on luokan PostgreDAOFactory metodi setExperimentalDataSource. (postgresgl-8.3-

603.jdbc4.jar) .

Testeissä käytetään JMock- ohjelmistoa. (jmock-2.4.0.jar, hamcrest-library-1.1.jar, hamcrest-core-1.1.jar, jmock-legacy-2.4.0.jar, cglib-nodep-2.1_3.jar, objenesis-1.0.jar, jmock-junit4-2.4.0.jar)

Yksikkötesteissä käytetään JUnit-kirjastoa.(junit-4.4.jar)

2.3 Tomcatin asetustiedostot

Tomcatin asetustiedosto "server.xml":n tulee sisältää Context- kohta, jossa on ohjelmiston hakemiston sijainti, sekä JNDI-resurssi PostgreSQL-tietokannan käyttämiseen.

```
<!-- Muppett context starts -->
<Context
        path="/tomcat/tkt_mupp/muppett"
        docBase="/home/tkt_mupp/tomcat/webapps/muppett/"
        debug="0"
        reloadable="true"
        crossContext="true"
        override="true">

  <!-- Context specific MuppettDB DataSource configuration -->
  <Resource name="jdbc/MuppettDB" auth="Container" type="javax.sql.DataSource"
    driverClassName="org.postgresql.Driver"
    url="jdbc:postgresql://localhost:13273/muppett"
    username="tkt_mupp"
    password="1muppett2"
    maxActive="8"
    maxIdle="4"
  />
```

```
</Context>
<!-- Muppett context ends -->
```

Context muuttujien selityksiä:

path: URL, joka ohjataan servletille.

docBase: ohjelmiston sijainti tiedostojärjestelmässä.

Resource muuttujien selityksiä:

url: tietokannan osoite, portti ja nimi

username: tietokannan käyttäjätunnus

password: tietokannan salasana

2.3.1 Servlettiasetukset (web.xml)

Tomcatin Muppett-järjestelmän asetustiedosto ”web.xml”:n tulee sisältää resource-ref -kohta, joka ottaa käyttöön server.xml:ssä määritellyn JDNI-resurssin. Lisäksi asetustiedoston tulee määritellä servletti FrontController ja ohjata kaikki .do -päätteiset palvelupyynnöt servletille. Mikäli JSP-sivuilla tapahtuu poikkeuksia, käyttäjä voidaan ohjata erilliselle virhesivulle error-page -määrittelyn avulla. Ominaisuus on kommentoitu pois, mutta halutessaan sen voi ottaa käyttöön poistamalla kommentit.

```
<resource-ref>
    <description>
      Resource reference to a factory for java.sql.Connection
      instances that may be used for talking to the database
      that is configured in server.xml.
  </description>
  <res-ref-name>jdbc/MuppettDB</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
```

```
  <res-auth>Container</res-auth>
 </resource-ref>

 <servlet>
  <servlet-name>FrontController</servlet-name>
  <servlet-class>muppett.controller.FrontController</servlet-class>
 </servlet>

 <servlet-mapping>
  <servlet-name>FrontController</servlet-name>
  <url-pattern>*.do</url-pattern>
 </servlet-mapping>

 <!--
 <error-page>
  <exception-type>java.lang.Exception</exception-type>
  <location>/html/error.html</location>
 </error-page>
 -->
```

2.3.2 Hakemistorakenne

Kun tomcat on asennettu, sijoitellaan tiedostoja hakemistorakenteeseen seuraavasti:

- tomcat/webapps/muppett kansioon sijoitetaan css-tiedostot (css), javascriptit (js) ja kuvat (images)
- tomcat/webapps/muppett/WEB-INF kansioon tulee jsp-sivut (jsp), javan lähdekoodit(src) ja käännetyt tiedostot (classes) sekä kirjastopaketit (lib) omiin kansioihinsa. Tähän hakemistoon tulee myös web.xml tiedosto.
- tomcat/webapps/muppett/WEB-INF/jsp hakemistoon laitetaan myös includoitavat tiedostot(html)

3 Tarkennuksia suunnitelmiin

GenericPostgreDAO

Eri PostgreDAO -luokkien metodit on toteutettu GenericPostrgeDAO -luokkaan ja joka on muiden PostgreDAO -luokkien yliluokka. GenericPostrgreDAO:n toteutusta ja käyttöä käsitellään tarkemmin luvussa 5.

Validaten poikkeukset.

Natiivityyppien validointi toteutettiin niin, että virhetilanteessa heitetään poikkeuksia eikä palauteta enumeraatiota. Tämä toteutettiin näin, koska esimerkiksi Integer luokka ei tarjoa metodeja arvonsa muuttamiseen.

Muutospyyntöjen tuonnin formaatti (IssueImportCommand.java & IssueImportCommandTest.java)

Jokainen rivi kuvaa yhden muutospyynnön tiedot. Rivillä käytetään eri tietojen erottelemiseen puolipistepistettä. Puolipisteen kirjoittaminen jonkin kentän arvoksi ei ole mahdollista. Yksi rivi on seuraavan muotoinen:

Tunniste;URL;Otsikko;Työmääräarvio;Työmääräarvaus;Tyyppi;Versio

Jokainen puolipisteellä erotettu tieto voi sisältää yhdet heittomerkit ("tieto"). Mikäli näin on, nämä heittomerkit poistetaan eli niitä ei huomioida tiedon osana. Heittomerkit eivät ole pakollisia. Rivin lopussa voi olla, tai olla olematta lopettava puolipiste. Lisäksi seuraavat ehdot on huomioitava:

- Kentän työmäärä-arvio - arvon tulee olla kokonaisluku.
- Kentän tyyppi tulee olla joko "change" (muutos) tai "fix" (korjaus).
- Muut kentät ovat tekstikenttiä, joilla on ainakin maksimipituusrajoitus. (rajoitukset määritelty tiedostossa muppett/controller/Validate.java ja tietokannan luontilauseissa).
- Ainakin toinen työmääräarviosta ja -arvauksesta on annettava.

- Tunniste, URL, Otsikko ja tyyppi on annettava.

- Versio voi olla tyhjä.

<-- Kelvollinen esimerkkitiedosto alkaa -->

"123";"www.google.com";Go;3;;"fix";"2";

456;www.google.fi;Go2;;arvio;change;2

<-- Kelvollinen esimerkkitiedosto päättyy -->

3.1 Metodimuutoksia

Vanha: ClassifyingVoting(VotingDTO voting, List<VoteDTO> votes, List<AllowedVoter-DTO> allowedVoters, List<IssueDTO> issues)

Uusi: ClassifyingVoting(VotingDTO voting, List<VoteDTO> votes, List<UniversityDTO> allowedVoters, List<IssueDTO> allowedIssues)

Vanha: Map< IssueID, Map<UniversityID, VoteValue> > votes();

Uusi: Map< IssueID, Map<UniversityID, String> > votes();

Poistettu: List<IssueDTO> listIssues(List<ExternalID> issues) throws ModelException

Uusi: public List<IssueDTO> listDeletedIssues() throws ModelException

public List<IssueDTO> listDeletedIssues() throws ModelException

Vanha: void update(VoteDTO vote) throws DAOException

Uusi: void update(VoteDTO value, VoteDTO match) throws DAOException

Vanha: void update(VotingIssueDTO votingIssue) throws DAOException

Uusi: void update(VotingIssueDTO value, VotingIssueDTO match) throws DAOException

Vanha: void update(UniversityModuleDTO universityModule) throws DAOException

Uusi: void update(UniversityModuleDTO value, UniversityModuleDTO match) throws DAOException

Uusi: abstract VotingIssueDAO createAllowedVoterDAO()

Uusi: abstract UniversityModuleDAO createUniversityModuleDAO()

Vanha: Set<VoteDTO> listVotes(VotingID voting, UniversityID university) throws ModelException

Uusi: Map<IssueID, String> listVotes(VotingID voting, UniversityID university) throws ModelException

Poistettu: void modifyVote(Set<VoteDTO> votes) throws ModelException

(Yhdistetty addVote-metodiin)

Uusi: IssueDTO listIssue(IssueID issue) throws ModelException

Vanha: VotingResult calcResult(VotingID votingID) throws ModelException

Uusi: VotingResult calculateResult(VotingID votingID) throws ModelException

Poistettu: IssueUpdateCommand extends Command

(Yhdistetty IssueAddCommand:iin, pitäisi varmaan uudelleennimetä)

Uusi: ModuleDTO listModule(ModuleID module) throw ModelException

Muuta:

OrderingVotingille sama muutos muodostimeen kuin ClassifyingVotingille.

UniversityModeliin

listUniversities-metodi ilman parametria - hakee kaikki, tarvitaan esim. AddUserissa

listUniversity-metodi, hakee yhden yliopiston tiedot

SendMail

Luokan metodit toteutettiin niin, että suunnitteludokumentin sendMail metodi toteutettiin private metodina ja erilaisten viestien lähettämiselle on lisätty omat metodinsa, jotka käyttävät sendMail() -metodia.

void sendMail(List<String> to, String subject, String message)

Uusi

void sendPasswordResetMail(`String email, String password`)

void sendUserAddMail(String email, String password)

void sendVotingReminder(`VotingID votingID`)

3.2 Luokkien muutokset ja uudet luokat

Vanha: ClassifyingVoting implements VotingResult

Vanha: OrderingVoting implements VotingResult

Uusi: abstract class VotingResultBase

Uusi: ClassifyingVoting extends VotingResultBase implements VotingResult

Uusi: OrderingVoting extends VotingResultBase implements VotingResult

Luokkaan VotingResultBase on koottu molemmille äänestystyyppiluokille yhteisiä tieto- ja ja metodeja.

Vanha ohjaus: classifying_voting.do classifying_voting.jsp

Uusi ohjaus: classifying_voting.do classification_voting.jsp

Luokka RequestWrapper lisätty, jotta requesteihin voidaan lisätä parametreja ennen dispatchaamista

Luokkaan VotingModel lisätty julkinen enumeraatio VotingType, joka määrittelee äänestystyypit. Tällä hetkellä CLASSIFYING_VOTING ja ORDERING_VOTING.

Luokka IssueMarkDeletedCommand extends Command lisätty

Luokka CreateUserCommand extends Command lisätty

ID-luokille yliluokka:

abstract class BaseID implements Comparable<BaseID>

Tälle aliluokat:

UserAccountID extends BaseID,

UniversityID extends BaseID,

IssueID extends BaseID,

ModuleID extends BaseID,

VotingID extends BaseID

Luokka ValidateException extends Exception lisätty

Tälle aliluokat:

InputEmptyException extends ValidateException

InputMalformedException extends ValidateException

InputNegativeException extends ValidateException

VotingUpdateCommand

Luokkaa ei toteutettu, koska sen toiminnallisuus toteutettiin VotingAddCommandiin.

IssueUpdateCommand
Luokkaa ei toteutettu, koska sen toiminnallisuus toteutettiin IssueAddCommandiin.

4 Toteuttamattomat vaatimukset ja muutokset vaatimusmäärittelydokumenttiin

Ryhmä toteutti lähes kaikki vaatimukset (kts. Taulukko 1). Joidenkin vaatimusten kohdalla toiminnallisuutta yksinkertaistettiin, tai pieniä osakokonaisuuksia karsittiin pois. Kokonaan pois jäi ainoastaan UC16 ja UC17, äänestyksen sulkeminen ja äänestyksen uudelleen avaaminen. Nämä toiminnot voi kuitenkin kiertää muuttamalla äänestyksen muokkaus-toiminnolla äänestyksen päättymispäivää.

Käyttötapaukset on priorisoitu asteikolla 1-4. Prioriteetin 1 käyttötapaukset ovat välttämättömiä, ja ilman niitä järjestelmä ei tuota siltä odotettuja palveluita. Prioriteetin 2 käyttötapaukset ovat keskeisiä järjestelmän käytön kannalta. Prioriteetit 3 ja 4 sisältävät käyttötapaukset, jotka pääasiassa parantavat järjestelmän käytettävyyttä. Käyttötapausten laajennukset mahdollisesti sisältävät alemman prioriteetin toimintoja.

Tunnus	**Nimi**	**Prior.**		
UC1	Järjestelmään kirjautuminen	1	Käyttäjä	OK
UC2	Äänestäminen	1	Käyttäjä	MUUTETTU
UC3	Järjestysäänestys	1	Käyttäjä	MUUTETTU
UC4	Luokitteluäänestys	1	Käyttäjä	OK
UC5	Käyttäjätietojen muokkaaminen	3	Käyttäjä	OK
UC6	Salasanan resetointi	2	Käyttäjä	OK
UC7	Käyttäjätunnuksen lisääminen	2	Ylläpitäjä	OK
UC8	Käyttäjätietojen muokkaaminen	4	Ylläpitäjä	MUUTETTU
UC9	Käyttäjätunnuksen poistaminen	4	Ylläpitäjä	OK
UC10	Muutospyynnön lisääminen	1	Ylläpitäjä	MUUTETTU
UC11	Muutospyyntöjen tuominen	1	Ylläpitäjä	OK
UC12	Muutospyynnön muokkaaminen	4	Ylläpitäjä	OK

UC13	Muutospyynnön poistaminen	4	Ylläpitäjä	MUUTETTU
UC14	Äänestyksen luominen	1	Ylläpitäjä	MUUTETTU
UC15	Äänestyksen muokkaaminen	3	Ylläpitäjä	MUUTETTU
UC16	Äänestyksen sulkeminen	3	Ylläpitäjä	EI
UC17	Äänestyksen avaaminen	3	Ylläpitäjä	EI
UC18	Äänestyksen poistaminen	4	Ylläpitäjä	OK
UC19	Äänestystietojen vieminen järjestelmästä	1	Käyttäjä	OK
UC20	Äänestyksen tietojen katsominen	2	Käyttäjä	OK
UC21	Käyttäjien selaaminen	4	Ylläpitäjä	OK
UC22	Järjestelmästä poistuminen	2	Käyttäjä	OK
	Laajennettavuus			OK
	Käytettävyys			MUUTETTU
	MailTimer			EI
	Sähköpostit			MUUTETTU

Taulukko 1: Muppett-järjestelmän käyttötapaukset

4.1 Huomioita käyttötapauksista

Seuraavassa kuvataan muutoksia vaatimusmäärittelydokumentin lupaamiin ominaisuuksiin.

4.1.1 UC2 Äänestäminen

Äänestysten tilat oli alun perin tarkoitus eritellä tuleviin, ei-äänestettyihin, äänestettyihin sekä suljettuihin ja näyttää käyttäjälle eri otsikoitten alla. Lopullisesta ratkaisusta jätettiin kokonaan pois ei-äänestetyt, ja lisäksi tulevat äänestykset näkyvät ainoastaan ylläpitäjälle.

Parannusehdotuksia: Järjestelmästä voisi käydä ilmi jollakin tavoin, missä äänestyksessä käyttäjän edustama yliopisto on jo äänestänyt. Käyttäjä ei myöskään voi suodattaa avoinna olevia äänestyksiä moduulin perusteella. Järjestelmän oli myös tarkoitus ”pakottaa” käyttäjät äänestämään loppuun ja ilmoittaa järjestämättömistä muutospyynnöistä.

4.1.2 UC3 Luokitteluäänestys

Vaatimusmäärittelydokumentin mukaan muutospyyntöjä tuli luokitella asteikolla 1-4, jonka lisäksi oli vaihtoehdot ”EOS” sekä ”tyhjä”. Muppett-järjestelmä ei sisällä ”tyhjä”-vaihtoehtoa.

4.1.3 UC 4 Järjestysäänestys

Järjestysäänestys pyrittiin toteuttamaan suunnittelun mukaan, mutta drag & drop -tekniikka vaihdettiin lopulta yksinkertaisempaan tapaan, jossa käyttäjä klikkaa muutospyynnön päällä, ja muutospyyntö siirtyy vastauskehykseen järjestyksessä aina seuraavaksi. Mikäli käyttäjä asettaa muutospyynnön liian alhaiselle prioriteetille, pitää hänen poistaa kaikki sen edellä olevat muutospyynnöt siihen asti, kunnes ollaan oikeassa kohdassa. Sen jälkeen käyttäjä voi klikata halutun muutospyynnön oikeaan kohtaan. Järjestysäänestykseen on jäänyt myös logiikkaongelma, joka kieltää tyhjän äänestyksen tallentamisen: kun käyttäjä on kerran tallentanut vastauksen järjestysäänestykseen, hän ei voi enää täysin tyhjentää vastauskenttää, vaan äänestykseen tallentuu vähintäänkin yksi järjestetty muutospyyntö.

Parannusehdotuksia: Mahdollisuus siirtää muutospyyntöjä yhden kehyksen sisällä ylös ja alas klikkaamalla olisi toivottavaa. Edellisen äänestyskerran tulokset tulisi olla jollakin tavoin näkyvissä, ja lisäksi vastauskehyksessä voisi selkeämmin ilmaista, millä sijalla mikin muutospyyntö milloinkin on.

4.1.4 UC7 Käyttäjätunnuksen lisääminen

Ylläpitäjän ei tarvitse pyytää järjestelmää lähettämään tiedot käyttäjälle, vaan järjestelmä lähettää ne automaattisesti. Järjestelmä huomauttaa käyttäjän luomisen yhteydessä mikäli tarvittavien kenttien sisällöissä on virheitä, mutta ei muistuta oikeasta muodosta.

4.1.5 UC8 Käyttäjätietojen muokkaaminen (ylläpitäjä)

Ylläpitäjä ei pääse suoraan käsiksi omiin tietoihinsa samalla tavalla kuin tavallinen

käyttäjä, vaan ylläpitäjän pitää valita tietonsa käyttäjälistauksesta muokattavaksi. Ylläpitäjä voi muuttaa myös omaa yliopistoaan.

4.1.6 UC10 UC11 Muutospyynnön lisääminen, muutospyyntöjen tuominen

Järjestelmä ei anna ylläpitäjän lisätä muutospyyntöä, jonka tunniste löytyy jo järjestelmästä. Vaatimusmäärittelydokumentti kertoo, että järjestelmän tuli ilmoittaa, mikäli jokin lisättävistä muutospyynnöistä eroaa jo lisätystä samalla tunnisteella olevasta muutospyynnöstä ja varmistaa korvataanko vanha uudella.

Parannusehdotuksia: Kun järjestelmässä on paljon muutospyyntöjä, tietyn muutospyynnön löytäminen tunnisteen perusteella on erittäin vaikeaa, koska muutospyyntölistauksessa näytetään muutospyynnöstä ainoastaan nimi ja moduuli.

Muutospyyntöjen versiotiedon voi jättää tyhjäksi, koska asiakas koki muutospyynnön version tallentamisen turhaksi. Muutospyynnölle voi tehdä versiomerkinnän, mutta varsinaista versiohistoriaa ei tallenneta järjestelmään.

4.1.7 UC13 Muutospyynnön poistaminen

Kun ylläpitäjä yrittää poistaa muutospyyntöä, joka on mukana meneillä olevassa äänestyksessä, tulisi järjestelmän ilmoittaa asiasta sekä näyttää muutospyynnön ja äänestyksen tiedot. Tällä hetkellä järjestelmä merkitsee muutospyynnön poistetuksi kysymättä mitään ylläpitäjältä. Poistetuksi merkitty muutospyyntö jää äänestykseen, ellei sitä käy erikseen poistamassa äänestyksen tiedoista.

4.1.8 UC14 Äänestyksen luominen

Käytettäessä aikaisempaa äänestystä uuden äänestyksen pohjana tulisi olla mahdollista kopioida myös muutospyynnöille annetut äänet. Muppett-järjestelmästä puuttuu kyseinen ominaisuus.

4.1.9 UC15 Äänestyksen muokkaaminen

Äänestystyypin muuttaminen kesken äänestyksen on disabloitu Muppett-järjestelmässä. Siihen tarvittava koodi on osittain olemassa, mutta vaatii jatkokehitystä. Koska äänestystyyppiä ei voi muokata, ääniä ei hävitetä, eikä näin ollen jo äänestäneille tarvitse ilmoittaa äänen häviämisestä, kuten vaatimusmäärittelyvaiheessa ajateltiin.

4.1.10 UC16 Äänestyksen sulkeminen

Äänestyksen sulkemiseen ei ole toteutettu omaa toimintoa, mutta äänestyksen voi sulkea asettamalla äänestyksen päättymispäivämääräksi kuluvan päivän. Äänestys sulkeutuu annettuna päivämääränä automaattisesti. Äänestyksen sulkeutuessa ylläpitäjältä ei kysellä enää äänestystuloksen laskuun liittyviä kohtia, vaan tulos lasketaan sen hetkisten merkintöjen mukaisesti. Laskutapaa voi muuttaa jälkeenpäin muokkaamalla äänestystä.

4.1.11 UC17 Äänestyksen avaaminen

Äänestyksen avaamiseen ei ole toteutettu omaa toimintoa, mutta äänestyksen voi avata uudestaan muokkaamalla sen alku- ja loppupäivämääriä. Äänestyksen voi myös kopioida uuden pohjaksi, mutta tällöin uusi äänestys ei enää sisällä vanhan äänestyksen jo annettuja ääniä.

4.1.12 Sähköpostit ja MailTimer

Järjestelmä lähettää käyttäjälle sähköpostiviestin ainoastaan silloin, kun käyttäjä on juuri lisätty järjestelmään, tai kun käyttäjä resetoi salasanansa. Alun perin sähköpostiviesti piti lähettää myös äänestyksen alkaessa, muistutuksena äänestyksestä äänestysajan puolessa välissä ja hieman ennen äänestysajan päättymistä, äänestyksen päättymisajankohdan tai äänestystyypin muuttumisesta ja äänestystuloksen julkaisemisesta.

Parannusehdotuksia: Asian voi korjata nopeasti lisäämällä muistuta-napin, jota klikkaamalla lähetetään muistutusviesti sellaisille äänestyksen äänestäjille, jotka eivät ole vielä äänestäneet kyseisessä äänestyksessä.

4.1.13 Yleisiä ongelmia

Käyttäjän tulee olla varovainen täyttäessään lomaketietoja sivuilla, joiden vasempaan laitaan on sijoitettu käytettävyyttä tehostava valikko. Mikäli käyttäjä kesken lomakkeen täyttämisen aktivoi valikosta linkin, lomakkeen tiedot häviävät sivun latautuessa uudelleen.

Parannusehdotuksia: Lomakkeen tietojen mahdollisesta häviämisestä tulisi ilmoittaa käyttäjälle ja antaa käyttäjän tehdä valinta, tallennetaanko tiedot ennen linkin avaamista.

4.2 JSP-sivut

Muppett-järjestelmä koostuu seuraavissa aliluvuissa esitellyistä JSP-sivuista.

4.2.1 classification_voting.jsp

- Ulkoisista linkeistä johtuvista virheistä johtuen tuotos ei ole välttämättä validia XHTML-merkkauskieltä, ei muita virheitä

JSP:n lukemat POST-parametrit:

voting_id

issueN_value_error

status

JSP:n lukemat GET-parametrit:

voting_id

JSP:n lähettämät POST-parametrit:

issueN_value

issueN_id

issue_count

voting_id

4.2.2 frontpage.jsp

- tuotos validia XHTML- merkkauskieltä

JSP:n lukemat POST-parametrit: -

JSP:n lukemat GET-parametrit: -

JSP:n lähettämät POST-parametrit: -

4.2.3 edit_issue.jsp

- Ulkoisista linkeistä johtuvista virheistä johtuen tuotos ei ole välttämättä validia XHTML-merkkauskieltä, ei muita virheitä.

JSP:n lukemat POST-parametrit:

issue_external_id_error

issue_header_error

issue_working_time_error

issue_working_time_guess_error

issue_type_error

issue_url_error

issue_version_error

issue_module_error

status

JSP:n lukemat GET-parametrit:

load: issue_external_id

issue_header

issue_work_time

issue_working_time

issue_working_time_guess

issue_type

issue_url

issue_version

issue_module

issue_id

(submitted)

JSP:n lähettämät POST-parametrit:

issue_external_id
issue_header

issue_work_time

issue_working_time

issue_working_time_guess

issue_type

issue_url

issue_version

issue_module

issue_id

Muuta:

Sisältää create_issue-lomakkeen, joilla on kaksi eri actionia:

issue_add.do ja issue_update.do

4.2.4 edit_user.jsp

- tuotos validia XHTML- merkkauskieltä

JSP:n lukemat POST-parametrit:

UID

user_first_name_error

user_last_name_error

user_email_error

user_university_error

user_password_error

status

JSP:n lukemat GET-parametrit:

UID?

JSP:n lähettämät POST-parametrit:

UID

uid

firstname

lastname

username

universities2 (select)

password1

password2

admin

Muuta:

Sisältää edit_user-lomakkeen, joilla on kaksi eri actionia:

user_update.do ja user_add.do

4.2.5 edit_voting.jsp

- tuotos validia XHTML- merkkauskieltä

JSP:n lukemat POST-parametrit:

error_voting_name

error_voting_module

error_voting_type

error_start_time

error_end_time

error_weighted

error_public_votes

voting

JSP:n lukemat GET-parametrit:

original_voting

voting_submitted

voting_module

voting_module_previous

voting_name

voting_type

start_time

end_time

voting_weighted

voting_public_votes

university_N

issue_N

voting

JSP:n lähettämät POST-parametrit:

original_voting

voting_submitted

voting_module

voting_module_previous

voting_name

voting_type

start_time

end_time

voting_weighted

voting_public_votes

voting_information_update

university_N

issueN

voting

Muuta:

Sivulla olevan lomakkeen action voting_add.do sisältää sekä lisäyksen ja muokkauksen.

4.2.6 forgotten_password.jsp

- tuotos validia XHTML- merkkauskieltä

JSP:n lukemat POST-parametrit:

invalid

JSP:n lukemat GET-parametrit: -

JSP:n lähettämät POST-parametrit:

user

4.2.7 login.jsp

- tuotos validia XHTML- merkkauskieltä

JSP:n lukemat POST-parametrit:

invalid

JSP:n lukemat GET-parametrit:

user

JSP:n lähettämät POST-parametrit:

user

password

4.2.8 logout.jsp

- tuotos validia XHTML- merkkauskieltä

JSP:n lukemat POST-parametrit: -

JSP:n lukemat GET-parametrit: -

JSP:n lähettämät POST-parametrit: -

4.2.9 ordering_voting.jsp

- Tuotos validia XHTML-merkkauskieltä lukuun ottamatta sitä, että id-kentän arvot eivät saa alkaa numerolla. Ulkoisista linkeistä saattaa aiheutua myös virheitä.

JSP:n lukemat POST-parametrit:

voting_id

status

JSP:n lukemat GET-parametrit:

voting_id

JSP:n lähettämät POST-parametrit:

issue_count

voting_id

issueN_value

issueN_id

Muuta:

Lomaketta lähetettäessä luodaan javascriptillä hidden-elementtejä, joilla äänen tallentamiseen tarvittavat arvot viedään käsittelevälle sivulle.

4.2.10 upload_issues.jsp

- Ulkoisista linkeistä johtuvista virheistä johtuen tuotos ei ole välttämättä validia XHTML-merkkauskieltä. Nähtävästi myös jokin ongelma divien ynnä muiden sellaisten sulkemisten kanssa. Vikaa ei saatu kartoitetuksi.

JSP:n lukemat POST-parametrit:

module_error

status

lines

lineN_error

JSP:n lukemat GET-parametrit:

issue_import_module

JSP:n lähettämät POST-parametrit:

file

issue_import_module

4.2.11 voting_report.jsp

- Ulkoisista linkeistä johtuvista virheistä johtuen tuotos ei ole välttämättä validia XHTML-merkkauskieltä

JSP:n lukemat POST-parametrit:

voting_id

file

JSP:n lukemat GET-parametrit:

voting_id

file

JSP:n lähettämät POST-parametrit:

voting (form: voting_edit_form)

original_voting

voting (form: export_voting)

4.2.12 voting.jsp

- tuotos validia XHTML- merkkauskieltä

- sivu on uusi

JSP:n lukemat POST-parametrit: -

JSP:n lukemat GET-parametrit: -

JSP:n lähettämät POST-parametrit: -

4.3 Sisällytettävät tiedostot

Seuraavat jsp- ja html-sivut liitetään useilla sivuilla jsp:n include-käskyllä:

4.3.1 footer.html

Sisältää sivun footer-divin, joka lisätään kaikkiin muihin sivuihin, paitsi login.jsp ja forgotten_password.jsp. Diviin on kapseloitu <noscript>-tagiin ilmoitus javascriptin tarpeellisuudesta järjestelmän sivuilla.

4.3.2 head.html

Tulostaa joka sivulle saman <head></head>-tagien sisällön, jotta yleiskäyttöiset css- ja js-tiedostot on helppo vaihtaa.

4.3.3 headImage.html

Järjestelmän kuvan voi vaihtaa vaihtamalla tiedostoon tallennetun kuvan lähteen.

4.3.4 issue_list.jsp

Tulostaa muutospyyntölistauksen, joka lisätään vasempaan laitaan edit_issue.jsp:ssä, upload_issues.jsp:ssä

JSP:n lukemat POST-parametrit:

module

show_only_deleted

JSP:n lukemat GET-parametrit:

deleted_issues

JSP:n lähettämät POST-parametrit:

module

show_only_deleted

issue_id

4.3.5 navigation.jsp

Tulostaa järjestelmän ylävalikon sivuille, jossa se on tarpeen. Hakee käyttäjän tiedot sessiosta, ja ne ovat siten käytettävissä kaikilla sivuilla, joihin navigation.jsp on liitetty.

4.3.6 user_list.jsp

- tuotos validia XHTML- merkkauskieltä

Tulostaa käyttäjälistauksen, joka lisätään vasempaan laitaan edit_user.jsp:ssä.

JSP:n lukemat POST-parametrit:

university

JSP:n lukemat GET-parametrit:

university

deleted_users

JSP:n lähettämät POST-parametrit:

universities (select)

UID

university

4.3.7 voting_list.jsp

-sivu on uusi

Tulostaa äänestyslistauksen, joka lisätään vasempaan laitaan edit_voting.jsp:ssä, classifying_voting.jsp:ssä, ordering_voting.jsp:ssä ja voting.jsp:ssä..

JSP:n lukemat POST-parametrit:

JSP:n lukemat GET-parametrit:

module

JSP:n lähettämät POST-parametrit:

module

4.4 CSS

Järjestelmän CSS on jaettu muutamaan eri tiedostoon TÄYSIN IRRATIONAALISESTI!

Muppett.css – validi tyylitiedosto

navigation.css – validi tyylitiedosto

- Sisältää sivun yllä olevan navigaation tyylin

orderingvoting.css – validi tyylitiedosto

- Sisältää järjestysäänestyksen tyylin

users.css – validi tyylitiedosto

4.5 Javascript

classification_voting.js

- Luokitteluäänestyksessä tarvittavat javascript-funktiot.

clickndrop.js

- Järjestysäänestyksessä tarvittavat javascript-funktiot.

misc.js

- Sisältää javascript-funktion kentän disablointiin sekä ylävalikon valitun otsikon taustavärin vaihtamiseen.

4.6 Käyttöliittymän ulkoasu

Käyttöliittymän ulkoasua muokattiin käytettävämmäksi, ja mm. Käyttäjä- ja muutospyyntönäkymän vasempaan laitaan lisättiin tehokkaamman selaamisen mahdollistava valikko.

5 Koodin ylläpitoon liittyvät seikat

5.1 GenericPostgreDAO

Kaikki DAO-rajapinnat sisältävät samat neljä metodia, jotka käsittelevä tietovarastoa. Jokaisen rajapinnan toteuttaminen omalla luokallaan olisi tuottanut runsaasti ns. "copy-paste"-koodia, joten päätimme tehdä javan edistyneempiä ominaisuuksia, geneerisiä luokkia ja reflectionia, käyttävän luokan GenericPostgreDAO:n, joka toteuttaa kerralla kaikki DAO-rajapinnat Postgres-tietovarastolle. GenericPostgreDAO:lla on tyyppiparametri <T>, joka kertoo mitä tietokannan osaa kyseisen konkreettisen tyypin (GenericPostgreDAO<T>) pitää käsitellä. Tyyppiparametristä päätellään reflectionin avulla käsiteltävä tietokannan taulu ja sen sarakkeet. Tämä asettaa rajoitteita tyyppiparametrina käytettävälle DTO-luokalle:

- Tietokannan taulun nimen tulee vastata DTO-luokan nimeä
  (koska java-luokkien nimien kirjoitusasu ei vastaa tietokannan taulujen nimien kirjoitusasua,
  muunnetaan luokan nimi tietokannan nimeksi GenericPostgreDAO:n metodilla "destroyCamelCase".)
- Tietokannan taulun sarakkeiden tulee vastata DTO-luokan jäsenmuuttujien nimiä. (Jälleen sama kirjoitusasumuunnos.)
- DTO-luokan jäsenten tulee olla julkisia oliomuuttujia (ei natiivityyppejä)
  (nämä rajoitteet voitaisiin poistaa muuttamalla GenericPostgreDAO:n toteutusta, kunhan DTO-luokat noudattaisivat jonkinlaista sopivaa JavaBean-tyylistä säännöllisyyttä.
- DTO-luokan jäsenillä pitää olla muodostin, joka ottaa parametrikseen tietokannasta haettavan String-arvon. Tyypit Boolean ja Date on jouduttu käsittelemään tämän rajoitteen takia erikoistapauksina GenericPostgreDAO:ssa. Näitä erikoistapauksia voi tarvittaessa lisätäkin.

Selvyyden vuoksi kaikille DAO-rajapinnoille on GenericPostgreDAO:n lisäksi tehty oma toteuttava PostgreDAO-luokkansa, joka sitten perii GenericPostgreDAO:n ja joka sisältää ainoastaan yhden muodostinkutsun. Tämä mahdollistaa myös sen, että jotkin PostgreDAO-luokat voisivat olla erikseen toteutettuja, ja jotkin GenericPostgreDAO:n toteuttamia (esimerkiksi jos jokin DTO-luokka ei noudata yllä lueteltuja rajoitteita, voidaan sille tehdä oma PostgreDAO-toteutuksensa). Näin PostgreDAO-luokkien käyttäjien ei tarvitse välittää GenericPostgreDAO-luokasta.

5.2 Hasher

Salasanan tiiviste pitäisi muuttaa käyttämään esimerkiksi Base64- enkoodausta, jotta voidaan helpommin välttää tietokannan ja Tomcat-palvelimen välisiä merkistöongelmia.

5.3 Äänestystyypin lisääminen

Järjestelmään voi lisätä uuden äänestystyypin. Uutta äänestystyyppiä varten tarvitsee luoda JSP-sivu, jossa on äänestämisen käyttöliittymä, Controlleriin kuuluva äänestystyypin Command-luokka, joka ottaa vastaan JSP-sivun lähettämän www-lomakkeiden tiedot ja Modeliin kuuluva äänestystyypin tulostenlaskentaluokka, joka toteuttaa VotingResult rajapinnan. Controlleriin lisättävä uusi komento pitää myös määritellä FrontControlleriin, jotta komennon ohjaus toimii oikein.

VotingModel:n calculateResult -metodi pitää muuttaa niin, että se hyväksyy uuden enumeraation. Uuden äänestystyypin lisääminen järjestelmään tapahtuu VotingModelin VotingType-enumeraatioon uusi arvo. VotingTypen enumeraation ensimmäinen parametri määrittää tunnisteen, joka on ainoastaan järjestelmän sisäisessä käytössä. Tunnisteiden tulee olla yksilöiviä. Jälkimmäinen parametri määrittää äänestystyypille kuvaavan nimen, joka näkyy esimerkiksi ylläpidon pudotusvalikoissa.

VotingTypen käyttäminen ei ole täysin yhdenmukaista järjestelmässä, joten lisättäessä uusi äänestystyyppi tulee tarkistaa viittaukset VotingDTO:n kenttään type, koska ää-

nestystyyppien tunnisteita on kovakoodattu JSP-sivuihin ja FormHelper-luokan metodeihin. Jatkokehitettäessä järjestelmää nämä olisi hyvä yhdenmukaistaa käyttämään VotingType:ä.

5.4 Validate luokan käyttö

Vain osa Validate luokan metodeista käyttää virheellisten syötteiden ilmaisemiseen poikkeuksia, koska syötteiden käsittelyä muutettiin sen jälkeen kun osa metodeista oli jo toteutettu. Yhtenäisyyden vuoksi myös loput olisi hyvä muuttaa käyttämään poikkeuksia.

5.5 SendMail

Nykyinen toteutus ei osaa käyttää sähköpostipalvelinta, joka vaatii autentikoinnin. Tarvittavat määritykset pitää lisätä. Palvelimen määritykset olisi järkevämpi sijoittaa määrittelytiedostoon, eikä suoraan SendMail luokkaan.

5.6 Moduulin lisääminen

Uuden moduulin lisääminen järjestelmään tapahtuu SQL-lauseella, joka on määritelty luvussa 7.1.1. Koska järjestelmä käyttää lyhenteitä moduuleille, eikä näitä ole määritelty tietokannassa, lyhenteet tulee lisätä FormHelper-luokan abbreviateModuleName-metodiin. Jatkokehitettäessä järjestelmää module-tauluun ja ModuleDTO- luokkaan voitaisiin lisätä lyhenne moduulille ja muuttaa FormHelper-luokan metodi käyttämään tätä.

5.7 CSV-tiedoston viennin muuttaminen

Järjestelmästä vietävän äänestystulostiedoston formaatin muuttaminen tapahtuu muuttamalla VotingExportCommand-luokan execute-metodia.

5.8 CSV-tiedoston tuonnin muuttaminen

Järjestelmään tuotavan muutospyyntötiedoston formaatin muuttaminen tapahtuu muuttamalla IssueImportCommand-luokan parseFile-metodia.

5.9 Commandin lisääminen

Uuden komennon lisääminen järjestelmään tapahtuu luomalla uusi aliluokka Command-luokalle. Command-luokan perivien luokkien tulee toteuttaa execute-metodi, joka sisältää komennon toiminnallisuuden ja requiresLogin- ja requiresAdmin-metodit joiden paluuarvot määrittelevät tarvitaanko komennon suorittamiseen kirjautuminen järjestelmään tai ylläpitäjän oikeudet. Mikäli halutaan tehdä ainoastaan jonkin resurssin, kuten JSP- tai HTML-sivun näyttävä komento, voidaan käyttää suoraan ForwardCommand-luokkaa, jonka konstruktorille määritellään parametreina resurssin polku ja kirjautumis- ja ylläpitäjävaatimukset. Uudet komennot tulee lisätä FrontController-luokan init-metodissa commands-assosiaatiotauluun. Assosiaatiotaulun avaimet toimivat URL-osoitteina komentoihin ja arvot sisältävät ilmentymän jostain Command-luokan aliluokasta.

6 Havaitut virheet

Järjestelmän ulkoiset linkit eivät ole oikean muotoisia, mikäli niissä ei ole http:// tai muuta vastaavaa alussa. Muuten linkit viittaavat suhteelliseen osoitteeseen.

Jos syötteissä on virheitä käyttäjää lisätessä, järjestelmä hävittää jo annetut oikeellisetkin tiedot.

Kun järjestelmästä viedään CSV-tiedostona äänestyksen tulokset, Excel ja Open Office Spreadsheet tulkitsevat äänestystuloksessa olevan desimaalipisteen päivämääräerottimeksi.

Järjestelmässä olevat filtteröinnit kadottavat tallentamattomat tiedot lomakkeista.

Tietokannasta tulee välillä SQL-poikkeus:”Insufficient data left in the stream”.

Päivämäärän tarkistus ei tarkista sisältöä: järjestelmä ei osaa antaa virheilmoitusta jos päivämäärä on oikean muotoinen mutta sisällöltään omituinen, esimerkiksi 99.13.2008.

Välillä juuri lomakkeen lähettämisen jälkeen filtteröitäessä ilmenee ongelmia, koska filteröintitoiminto lähettää lomakkeen uudelleen virheellisesti.

7 Muut ylläpitoon vaikuttavat seikat

7.1 SQL-lauseet niille toiminnoille, joita ei voi käsitellä käyttöliittymästä

Kaikkia järjestelmässä olevia tietoja ei pääse käsittelemään suoraan käyttöliittymän kautta. Tällaisia tietoja on mm. yliopisto-, moduulitiedot. Näitä tarkoitus käsitellä suoraan tietokannasta ja niitä varten tarvittavat SQL-komennot on kirjattu tässä luvussa. Lisäksi suoraan tietokantaan tehtäviä operaatiota tarvitaan jos poistetuksi merkitty muutospyyntö halutaan palauttaa takaisin ja järjestelmän käyttöotossa, kun luodaan järjestelmän ensimmäinen käyttäjätunnus.

7.1.1 Tietokantaan lisääminen ja tiedon muokkaaminen

Yliopiston lisääminen:

insert into university (name, abbreviation, weight) values('Yliopiston nimi', 'lyhenne', Painoarvo_numeroina).

Yliopiston muokkaaminen:

update university set name='Uusi Nimi', abbreviation='Uusi lyhenne', weight=painoarvo_numerona

Moduulin lisääminen:

insert into module (name) values ('Modulin nimi')

Moduulin muokkaaminen:

update module set name='Uusi Nimi'

Poistetuksi merkityn muutospyynnön palauttaminen:

update issue set deleted='f' where external_id='EXTERNAL_ID';

Yliopiston liittäminen oletusäänestäjäksi moduulia koskevissa äänestyksissä.

insert into university_module (university, module) values (university_id, module_id)

Käyttäjän lisääminen

insert into user_account (first_name, last_name, password, password_salt, email, university, admin) values

(etunimi, sukunimi, salasana, salasanansuola, sähköpostiosoite, ylläpitäjä/käyttäjä)

- salasana pitää resetoida, koska sen kuuluu olla tietokannassa tiivistettynä.

7.1.2 Tietokannasta poistaminen

Tietokannasta voi tarvittaessa myös poistaa yliopisto-, muutos- ja moduulitietoja alla olevilla komennoilla. Tietueen poistaminen ei kuitenkaan onnistu jos tietueeseen on viittauksia tietokannan muista tietueista. Ennen poistokomennon suorittamista pitää tietueeseen olevat viitteet poistaa.

Yliopisto nimen mukaan:

delete from university where name='Yliopiston nimi'

Yliopisto lyhenteen mukaan:

delete from university where addreviation='Yliopiston lyhenne'

Moduuli nimen mukaan:

delete from module where name='Moduulin nimi'

Muutospyyntö external_id mukaan:

delete from issue where external_id='EXTERNAL_ID'

7.2 Tietokannan alkuarvoja

INSERT INTO university (name, abbreviation, weight) VALUES

('Helsingin yliopisto', 'HY', 6),

('Teknillinen korkeakoulu', 'TKK', 4),

('Oulun yliopisto', 'OY', 4),

('Joensuun yliopisto', 'JOY', 3),

('Helsingin kauppakorkeakoulu', 'HSE', 2),

('Lappeenrannan teknillinen yliopisto', 'LTY', 2),

('Vaasan yliopisto', 'VY', 2),

('Lapin yliopisto', 'LY', 2),

('Sibelius-Akatemia', 'SibA', 1),

('Svenska handelshögskolan', 'SHH', 1),

('Taideteollinen korkeakoulu', 'TaiK', 1),

('Turun kauppakorkeakoulu', 'TSE', 1),

('Teatterikorkeakoulu', 'TeaK', 0.25)

insert into module (name) values('eHops'), ('winOodi'), ('webOodi'), ('OpasOodi'), ('OpeOodi')

7.3 Ensimmäinen käyttäjätunnus

Ensimmäisen käyttäjän luominen tapahtuu käyttäjän lisäämiseen tarkoitetulla SQL-lauseella. SQL-lauseella luodun käyttäjän salasana ei toimi, koska salasanojen pitää olla tietokannassa tiivistettynä. Toimiva salasana luodaan järjestelmän salasanan resetointitoiminnon avulla, joka luo käyttäjälle uuden salasanan ja lähettää sen käyttäjälle määriteltyyn sähköpostiosoitteeseen.

Käyttäjän lisääminen

insert into user_account (first_name, last_name, password, password_salt, email, university, admin) values

(etunimi, sukunimi, tilapäinen_salasana, salasanansuola, sähköpostiosoite, yliopisto, ylläpitäjä/käyttäjä)

Jos järjestelmän käyttöönoton yhteydessä ei ole käytettävissä sopivaa sähköpostipalvelinta salasanan resetointia varten voidaan käyttää CreateUserCommand komentoa. Komentoon määritellään halutut käyttäjätiedot ja komento suoritetaan kutsumalla selmaille muppett/create_user.do komentoa. Komennolla voi lisätä järjestelmään vain ensimmäisen käyttäjän.

7.4 Tietokannan varmuuskopiointi

7.4.1 Tietokannan varmuuskopion luominen

save_dump.sh skriptin sisältö.

#!/bin/bash

DBNAME=muppett

PGPASSWORD=tietokannan_sa_salasana

PGPORT=tietokannan_tietoliikenneportti

PGDATA=/home/tkt_mupp/postgres/var

PGPATH=/usr/local/pgsql-8.2/bin

echo Saving database dump;

PATH=${PATH}:${PGPATH} PGPORT=${PGPORT} PGDATA=${PGDATA} PGPASSWORD=${PGPASSWORD} pg_dump ${DBNAME} --clean > ${DBNAME}_dump.sql

echo Database dump saved;

7.4.2 Tietokannan palauttaminen varmuuskopiosta

load_dump.sh skriptin sisältö.

```
#!/bin/bash


DBNAME=muppett

PGPASSWORD=tietokannan_sa_salasana

PGPORT=tietokannan_tietoliikenneportti

PGDATA=/home/tkt_mupp/postgres/var

PGPATH=/usr/local/pgsql-8.2/bin


echo Loading database dump;

PATH=${PATH}:${PGPATH} PGPORT=${PGPORT} PGDATA=${PGDATA} PGPASS-
WORD=${PGPASSWORD} psql ${DBNAME} --single-transaction <
${DBNAME}_dump.sql >/dev/null

echo Database dump loaded;
```