

# **Suunnitteludokumentti**

NJC2

Helsinki 18.3.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (6 ov)

**Projektiryhmä**

Eero Anttila

Olli Jokinen

Jesse Liukkonen

Jani Markkanen

Jere Salonen

Jouni Tuominen

**Asiakas**

Olli Lahti

**Johtoryhmä**

Juha Taina

**Kotisivu**

<http://www.cs.helsinki.fi/group/njc2/>

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
1.0	10.3.2004	Ensimmäinen versio
1.1	15.3.2004	Tietokantakyselyt luokkakuvausten yhteyteen
1.2	18.3.2004	Tietokantakaavion päivitys, luokkakuvausten päivitys

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Dokumentin tarkoitus . . . . .	1
1.2	Tuote ja ympäristö . . . . .	1
1.3	Määritelmät, merkintätavat ja lyhenteet . . . . .	1
1.4	Yleiskatsaus dokumenttiin . . . . .	2
<b>2</b>	<b>Järjestelmän yleiskuvaus</b>	<b>2</b>
2.1	Sovellusalueen kuvaus . . . . .	2
2.2	Järjestelmän liittyminen ympäristöönsä . . . . .	3
2.3	Laiteympäristö . . . . .	3
2.4	Ohjelmistoympäristö . . . . .	3
2.5	Toteutuksen keskeiset reunaehdot . . . . .	3
2.6	Sopimukset ja standardit . . . . .	3
<b>3</b>	<b>Arkkitehtuurin kuvaus</b>	<b>4</b>
3.1	Arkkitehtuurin yleiskuvaus . . . . .	4
3.2	Ohjelmiston toiminta esimerkkikäyttötapausten avulla . . . . .	4
3.2.1	Sisäänkirjautuminen . . . . .	4
3.2.2	Tietojen muokkaus . . . . .	5
<b>4</b>	<b>Virheen käsittely</b>	<b>5</b>
4.1	Käyttäjän virheiden käsittely . . . . .	5
4.2	Laitteistovirheiden käsittely . . . . .	6
<b>5</b>	<b>Luokat ja metodit</b>	<b>7</b>
5.1	ControllerServlet . . . . .	7
5.2	DAO . . . . .	7
5.3	SessionData . . . . .	9
5.4	Event . . . . .	10
5.5	HtmlTyokalu . . . . .	11
5.6	TiedostoServlet . . . . .	11
5.7	Tiedosto . . . . .	11
5.8	Postittaja . . . . .	12

5.9	Salaaja . . . . .	12
5.10	DBConnector . . . . .	12
5.11	Tulos . . . . .	13
5.12	Sähköposti . . . . .	14
5.13	Paivamaara . . . . .	14
5.14	Lausunto . . . . .	15
5.15	Asiantuntija . . . . .	16
5.16	Artikkeli . . . . .	16
5.17	Kayttaja . . . . .	18
5.18	Alat . . . . .	19
<b>6</b>	<b>JSP-sivut</b>	<b>19</b>
6.1	Kaikkien käyttäjien sivut . . . . .	19
6.1.1	kehys.jsp . . . . .	19
6.1.2	menu.jsp . . . . .	19
6.1.3	kirjautuminen.jsp . . . . .	20
6.1.4	profilinmuokkaus.jsp . . . . .	20
6.1.5	uloskirjautuminen.jsp . . . . .	20
6.2	Kirjoittajan sivut . . . . .	20
6.2.1	rekisteroityminen.jsp . . . . .	21
6.2.2	omatartikkelit.jsp . . . . .	21
6.3	Asiantuntijoiden sivut . . . . .	21
6.3.1	artikkelienarvostelu.jsp . . . . .	22
6.4	Toimituksen sivut . . . . .	22
6.4.1	artikkelienhallinta.jsp . . . . .	22
6.4.2	lehdentila.jsp . . . . .	23
6.4.3	kayttajahallinta.jsp . . . . .	23
6.4.4	raportit.jsp . . . . .	23
6.4.5	sahkoposti.jsp . . . . .	23
6.4.6	paatoimittajanakyma.jsp . . . . .	24
<b>7</b>	<b>Tietokantakuvaus</b>	<b>24</b>
7.1	Tietokantakyselyt ja transaktioiden sitoutuminen . . . . .	24
7.2	Käyttäjä . . . . .	26

7.3	Asiantuntija . . . . .	26
7.4	Osaamisala . . . . .	27
7.5	Ala . . . . .	27
7.6	Sähköposti . . . . .	27
7.7	Lehti . . . . .	27
7.8	Artikkeli . . . . .	28
7.9	Versio . . . . .	28
7.10	Lausuntopyyntö . . . . .	29
7.11	Lausunto . . . . .	29
7.12	Liite . . . . .	30
7.13	Päätös . . . . .	30
7.14	Loki . . . . .	30
7.15	Järjestelmään liittyvien tiedostojen tallennus . . . . .	31
<b>8</b>	<b>Käyttöliittymä</b>	<b>31</b>
8.1	Kaikki käyttäjäryhmät . . . . .	32
8.1.1	Kirjautuminen . . . . .	32
8.2	Kirjoittaja . . . . .	32
8.2.1	Rekisteröityminen . . . . .	32
8.2.2	Omat artikkelit . . . . .	33
8.2.3	Profilin muokkaus . . . . .	33
8.3	Asiantuntija . . . . .	34
8.3.1	Artikkelien arvostelu . . . . .	34
8.3.2	Profilin muokkaus . . . . .	35
8.4	Toimitus . . . . .	35
8.4.1	Artikkelien hallinta . . . . .	35
8.4.2	Lehden tila . . . . .	36
8.4.3	Käyttäjähallinta . . . . .	37
8.4.4	Raportit . . . . .	38
8.4.5	Profilin muokkaus . . . . .	38
8.4.6	Päätoimittajanäkymä . . . . .	38
<b>9</b>	<b>Järjestelmästä saatavat raportit</b>	<b>39</b>

<b>10 Erityiset tekniset ratkaisut</b>	<b>40</b>
10.1 Salasanan tallennus . . . . .	40
10.2 Sähköpostin lähetys . . . . .	40
10.3 Tiedostojen käsittely . . . . .	40

## **Liitteet**

### **1 Luokkakaavio**

# 1 Johdanto

Tämä dokumentti on osa tietojenkäsittelytieteen laitoksen NJC2-ryhmän ohjelmistotuotantoprojektia. Tarkoituksena on Nordic Journal of Computing -lehden toimeksiantamana tuottaa väline, jolla tehostetaan kyseisen lehden julkaisuprojektia.

Valmis tuote tulee Nordic Journal of Computing -lehden toimituksen käyttöön. Tuotteen oikeudet on luovutettu Helsingin yliopistolle, joka julkaisee tuotteen GNU General Public License tai GNU Lesser General Public Licensen alaisuudessa.

## 1.1 Dokumentin tarkoitus

Tämän suunnitteludokumentin tarkoituksena on tarkentaa määrittelydokumentin tiedot suunnitteluvaiheen vaatimaan muotoon. Valmiista dokumentista on käytävä ilmi, miten ohjelmiston määrittely ja vaatimukset tullaan toteuttamaan. Suunnitelmien on oltava niin tarkkoja, että toteutus onnistuu suoraviivaisesti niiden perusteella.

## 1.2 Tuote ja ympäristö

Projektin tarkoituksena on toteuttaa Nordic Journal of Computing -lehden toimitukselle väline, jolla nopeutetaan ja helpotetaan lehden julkaisuprosessia. Tuotteen on määrä yksikertaistaa NJC -lehden julkaisuprosessia, erityisesti kirjanpidon ja kommunikoinnin osalta.

NJC-lehden toimituksen lisäksi tuote yksinkertaistaa myös lehteen kirjoittavien ja kirjoitusten tarkastajina toimivien asiantuntijoiden työtä. Molemmille sidosryhmille tarjotaan HTML-lomakkeisiin perustuvat käyttöliittymät, joiden kautta he kommunikoivat toimituksen kanssa. Tuotteen käyttämiseen vaaditaan kehyksiä (frameset) tukeva WWW-selain ja Internet-yhteys.

## 1.3 Määritelmät, merkintätavat ja lyhenteet

Seuraavassa taulukossa mainitaan dokumentissa käytettävät vieraat lyhenteet, määritelmät ja merkintätavat.



Termi/lyhenne	Merkitys
CSS	Cascadin Style Sheets, tyylisivut HTML-kielen ulkoasun määrittelyyn
DAO	Data Access Object, suunnittelumalli (design pattern) alhaisen tason tiedon saannin logiikan erottamiseksi korkean tason logiikasta ja tässä ohjelmistossa käytettävän luokan nimi
GNU-lisenssi	General Public License, lisenssin alaiset ohjelmat ovat vapaasti levitettävissä ja muunneltavissa
JSP	JavaServer Pages, teknologia dynaamisten verkkosivujen tekoon
NJC	Nordic Journal of Computing -lehti, projektin asiakas
NJC2	Ryhmän nimi
PostgreSQL	Tietokantaohjelmisto joka tukee SQL-toimintoja
Servlet	Java-kielinen palvelimella suoritettava ohjelma
Tomcat	Servlet-alusta, servlettien hallitsemiseen

## 1.4 Yleiskatsaus dokumenttiin

Seuraavassa luvussa esitellään toteutettavan järjestelmän yleiskuvaus, lähtien projektin tarkoituksesta päätyen tarvittaviin ohjelmistoihin ja asiakkaan kanssa sovittuihin toteutuksen reunaehtoihin. Luvussa 3 keskitytään ohjelmiston arkkitehtuurin yleiskuvaukseen siten, että lukija saisi käsityksen koko ohjelmiston rakenteesta. Rakenteen esittelyn jälkeen luvussa 4 kerrotaan yleisiä virheenkäsittelysääntöjä ja luvussa 5 kerrotaan yksityiskohtaisesti luokkien ja metodien tarkoitukset sekä niiden kommunikointirajapinnat.

Luvussa 6 esitellään lyhyesti järjestelmään sisältyvät JavaServer Pages (JSP) -sivut. Tämän jälkeen luvussa 7 esitellään ohjelmiston tietokantarakenne ensin yleiskuvauksena ja sitten yksityiskohtaisesti tietokantataulu kerrallaan. Luku 8 keskittyy käyttöliittymiin. Luvussa 9 käydään läpi järjestelmästä saatavia raportteja ja luvussa 10 keskitytään erityisiin teknisiin ratkaisuihin.

## 2 Järjestelmän yleiskuvaus

Luvussa esitellään toteutettavan järjestelmän yleiskuvaus, johdatus asiakkaan ympäristöön ja sovellusalueeseen.

### 2.1 Sovellusalueen kuvaus

Projektin on tarkoitus toteuttaa Nordic Journal of Computing -lehden toimitukselle väline, jolla nopeutetaan ja helpotetaan lehden julkaisuprosessia. Tuotettavan ohjelmiston on tarkoitus automatisoida artikkelin tarkastuksen vaiheita ja helpottaa toimituksen kommunikointia sidosryhmien kanssa.

Projekti kuuluu Helsingin yliopiston tietojenkäsittelytieteen laitoksen Ohjelmistotuotantoprojektikurssiin. Järjestelmä tulee Nordic Journal of Computing -lehden toimituksen käyttöön.

## 2.2 Järjestelmän liittyminen ympäristöönsä

Järjestelmä toimii itsenäisesti olematta minkään muun järjestelmän osana. Järjestelmä käyttää apunaan lähtevän sähköpostin palvelinta.

## 2.3 Laiteympäristö

Ohjelmistoa suoritetaan tietojenkäsittelytieteen laitoksen alkokrunni-palvelimessa, johon on asennettu tarpeelliset palvelin ym. ohjelmistot. JSP- ja Servlet-sivuja suoritetaan Tomcat-ohjelmiston avulla, joka on myös asennettu valmiiksi alkokrunni-palvelimelle. Tietokantana käytetään PostgreSQL:ää.

## 2.4 Ohjelmistoympäristö

	Ohjelma	Versio
Käyttöjärjestelmä	Linux	CSL 2.113
Kääntäjä	Java	j2sdk 1.4.2
JSP ja JavaServlet ajoympäristö	Tomcat	4.1.18
WWW-palvelin	Apache	
Tietokantaohjelmisto	PostgreSQL	7.4
Lähtevän sähköpostin palvelin	Sendmail	
WWW-selain	Internet Explorer	5.0
	Mozilla	1.6
	Netscape	6
	Opera	6

## 2.5 Toteutuksen keskeiset reunaehdot

Tärkeimpiä vaatimuksia ovat:

- Sovelluksen oltava tietoturvallinen
- Käyttöliittymä tehdään englannin kielellä
- Projektin oikeudet ovat Helsingin yliopistolla
- Ohjelmisto julkaistaan GNU-lisenssin alaisuudessa

## 2.6 Sopimukset ja standardit

Koodausperiaatteet

- Koodi kirjoitetaan suomenkielellä

- Luokkien nimet kirjoitetaan isolla alkukirjaimella
- Metodit ja muuttujat kirjoitetaan pienelle alkukirjaimelle
- Sanat erotetaan isolla kirjaimella
- Final-tyyppiset muuttujat kirjoitetaan kokonaan isolla ja sanat erotetaan alaviivalla
- Kommentoinnit tehdään javadoc-tyylin mukaisesti ja suomenkielellä

Käytettävät pakkaukset

Kuvaus	Nimi	pakkaus
Tiedostojen Upload-toiminto	Apache FileUpload 1.0	commons-fileupload-1.0.jar
Yhteys PostgreSQL-kantaan	JDBC3	pg74jdbc3.jar
Testaustyökalu	JUnit 3.8.1	junit.jar
Sähköpostin lähetys	JavaMail 1.3.1	mail.jar

## 3 Arkkitehtuurin kuvaus

### 3.1 Arkkitehtuurin yleiskuvaus

Järjestelmä jaetaan kolmeen erilliseen osajärjestelmään: web-pohjaiseen käyttöliittymään, tietokannan ja käyttöliittymän välillä toimivaan Java-moduuliin sekä itse tietokantaan.

Käyttäjille näkyvä osa on selaimen kautta käytettävä käyttöliittymä, jonka avulla käyttäjät lähettävät artikkeleitaan toimitukselle, joka puolestaan lähettää ne eteenpäin asiantuntijoille. Sekä kirjoittajat, asiantuntijat että toimittajat käyttävät järjestelmää web-käyttöliittymän kautta. Osa käyttöliittymästä on kaikille käyttäjäryhmille yhteistä, mutta pääasiassa jokainen ryhmä käyttää järjestelmää oman käyttöliittymänsä kautta.

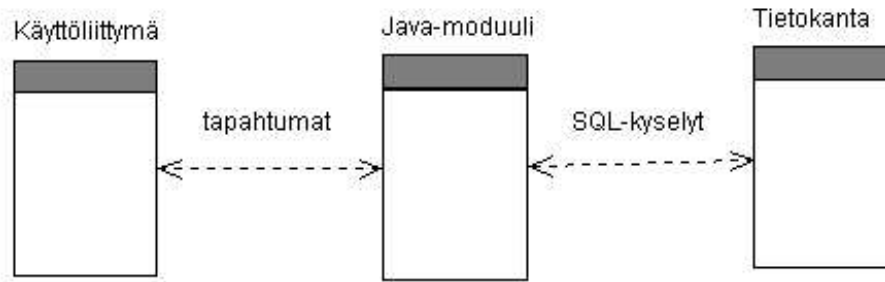
Java-moduuliin sijoitettu servlet kuuntelee käyttöliittymäsivuilta lähetettäviä tapahtumia (event), käsittelee ne ja hakee kyselyiden avulla tietokannasta tarvitsemansa tiedot. Tällaisia tapahtumia ovat esimerkiksi käyttäjän sisäänkirjautuminen ja rekisteröityminen sekä uuden artikkelin lähettäminen. Tietokantakyselyiden tulosten perusteella luodaan käyttäjille päivittynyt käyttöliittymäsivu. Järjestelmän Java-osio siis käsittelee käyttöliittymän ja tietokannan välillä kulkevaa tietoa.

### 3.2 Ohjelmiston toiminta esimerkkikäyttötapausten avulla

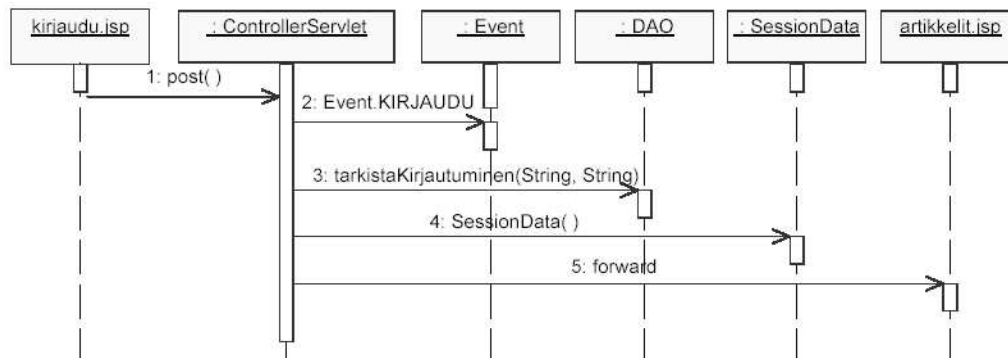
Tässä aliluvussa kuvataan luokkien välistä kommunikointia sekvenssikaavioiden avulla.

#### 3.2.1 Sisäänkirjautuminen

Sisäänkirjautuminen (kuva 2) tapahtuu kirjaudu.jsp -sivulta, joka lähettää käyttäjän syötämät tiedot ControllerServlet:ille. ControllerServlet ... ??



Kuva 1: Arkkitehtuurin yleiskuvaus



Kuva 2: Sekvenssikaavio, sisäänkirjautuminen.

### 3.2.2 Tietojen muokkaus

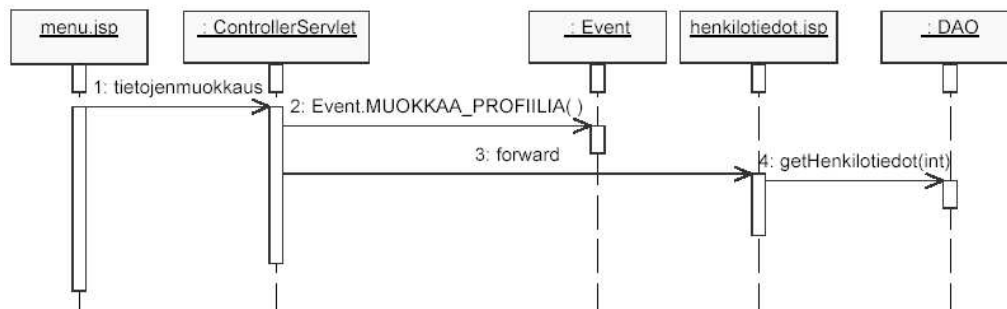
Kuvassa 3 tietojen muokkaus -sivulta menu.jsp lähettää eteenpäin ControllerServlet:ille...??

## 4 Virheen käsittely

Tässä luvussa käsitellään yleisiä virheenkäsittelysääntöjä. Käyttöliittymiin liittyvät virhetilanteet löytyvät luvun 8 käyttöliittymäkuvausten yhteydestä.

### 4.1 Käyttäjän virheiden käsittely

Käyttäjän syötteistä johtuvien virheiden käsittely jaetaan kahteen luokkaan: välttämättömään ja hyödylliseen virheenkäsittelyyn. Välttämättömätön virheenkäsittely kattaa kaikki virheet, jotka saattavat häiritä merkittävästi ohjelman toimintaa. Tällaisia ovat esimerkiksi syötteen väärä muoto tai tyhjä syöte silloin, kun se ei ole sallittu. Välttämätön virheenkäsittely toteutetaan Javalla, joka rajoittaa virheiden huomaamisajankohtaa. Javalla tehdyt virheet voidaan korjata vasta sitten, kun käyttäjä on lähettänyt virheet sisältävät tiedot



Kuva 3: Sekvenssikaavio, tietojen muokkaus.

palvelimelle. Virheen sattuessa virheen sisältävä sivu ladataan uudestaan näkyviin asiaankuuluvalla virheilmoituksella varustettuna, eikä mitään sivun sisältämiä muutoksia vielä tallenneta järjestelmään.

Välttämätön virheenkäsittely toteutetaan kokonaisuudessaan ensin, jonka jälkeen jatketaan hyödyllisen virheenkäsittelyn toteuttamisella. Hyödyllinen virheenkäsittely toteutetaan JavaScriptilla, jolla voidaan huomata virheet välittömästi. JavaScriptilla toteutetut tarkastukset vaativat Javalla tehdyn tarkastuksen alleen, koska käyttäjä voi kytkeä JavaScriptin pois selaimesta.

JavaScriptilla toteutetussa virheenkäsittelyssä on huomattava, että käyttäjän toimintoja ei saa rajoittaa liian nopeasti. Esimerkiksi jos syötteessä ei ole sallittu lainausmerkkiä (”), ei käyttäjää kannata estää kirjoittamasta sitä. Mieluummin käytetään jotain huomioväriä, jolloin käyttäjä näkee, ettei lainausmerkki ole sallittu. Virhetekstit luokitellaan CSS-komennolla (tag) `class="virhe"` ja sijoitetaan välittömästi virheen oikealle puolelle tai alle.

## 4.2 Laitteistovirheiden käsittely

Laitteistovirheet on pyritty ottamaan huomioon ensisijaisesti tietokantakäytäntöä suunniteltaessa. Järjestelmän kaatuessa laitteistovirheen johdosta käyttäjän syöttämää tietoa saattaa kadota, mutta tietokanta pysyy eheänä. Virheen sattuessa tai session päättyessä esimerkiksi selaimen sulkemisen tai uloskirjautumisen vuoksi mahdolliset väliaikatiedostot ja keskeneräiset työt menetetään. Tomcatin ja Postgresin käynnistettyä laitteistovirheen jälkeen järjestelmä on jälleen normaalisti käytettävissä.

Tietokantaoperaatioiden toteutuksesta ja sitoutumiskäytännöstä on tarkempi kuvaus luvussa 7.

## 5 Luokat ja metodit

### 5.1 ControllerServlet

Pääservlet, joka ohjaa pyynnöt oikeille JSP-sivuille.

*Konstruktorit*

*Metodit*

**public post()**

??

**public tietojenmuokkaus()**

??

### 5.2 DAO

DAO-luokan avulla päästään käsiksi järjestelmän tietosisältöön. Luokan avulla voidaan lukea tai kirjoittaa tietokantaan.

*Konstruktorit*

**public DAO()**

Luo DBConnector-olion ja luo yhteyden kantaan.

*Metodit*

**public Kayttaja getKayttajatiedot(int kayttajaID)**

Palauttaa Kayttaja-olion halutusta henkilöstä.

**public boolean submitKayttajatiedot(Kayttaja uusiKayttaja, String salasana)**

Päivittää tietokannassa olevaa käyttäjätietoriviä. Jos riviä ei löydy, sellainen luodaan. Jos salasana-String on alustettu, asetetaan se käyttäjän uudeksi salasanaksi. Jos salasana on null, siihen ei tehdä muutoksia.

**public List getArtikkelit()**

Palauttaa kaikki artikkelit listana, joka sisältää Artikkelio-olioita.

**public Artikkelit getArtikkeli(int artikkeliId, int versioNro)**

Palauttaa Artikkelio-olion halutusta artikkelista.

, **public List getKirjoittajanArtikkelit(int kayttajaId)**

Palauttaa listan Artikkelio-olioita halutun käyttäjän artikkeleista.

Metodi käyttää tietokantakyselyä, joka hakee rivejä tauluista Artikkelio ja Versio attribuut-

tin `kayt_id` perusteella.

**public List getAsiantuntijanArtikkelit(int kayttajaId)**

Palauttaa listan Artikkelio-olioista, joissa kyseinen käyttäjä on arvostelijana.

**public List getAlat()**

Palauttaa String-listana tietokantaan tallennetut artikkelien alaluokitukset.

Metodi käyttää tietokantakyselyä, joka hakee kaikki rivit taulusta Ala.

**public List getAsiantuntijat()**

Palauttaa Asiantuntija-listana kaikki asiantuntijat.

Metodi käyttää tietokantakyselyä, joka hakee rivejä taulusta Kayttaja attribuutin rooli perusteella.

**public List getAsiantuntijat(int artikkeliId, int versioNro)**

Palauttaa haluttuun artikkeliin kiinnitetyt asiantuntijat. Metodi käyttää tietokantakyselyä, joka liittää taulut Lausuntopyyntö ja Versio attribuuttien artik\_id ja versionro perusteella, sekä näin saadun tuloksen perusteella hakee rivejä taulusta Kayttaja.

**public List getLausunnot(int artikkeliID, int versioNro, boolean kirjoittaja)**

Palauttaa artikkeliin liittyvät lausunnot Lausunto-listana. Jos boolean-muuttujan kirjoittaja arvo on true, palautetaan kirjoittajalle tarkoitetut lausunnot. Jos arvo on false, palautetaan toimitukselle tarkoitetut lausunnot.

Metodi käyttää tietokantakyselyä, joka hakee rivejä taulusta Lausunto attribuuttien artik\_id ja versionro perusteella.

**public Lausunto getLausunto(int kayttajaId, int artikkeliId, int versioNro, boolean kirjoittaja)**

Palauttaa tietyn asiantuntijan (kayttajaId) tietylle artikkelille antaman lausunnon Lausunto-oliona.

**public boolean tarkistaKirjautuminen(String tunnus, String salasana)**

Tarkistaa käyttäjätunnuksen ja salasanan oikeellisuuden.

Metodi käyttää tietokantakyselyä, joka liittää taulut Kayttaja ja Sposti attribuutin kayt\_id perusteella.

**public boolean insertAla(String alanNimi)**

Lisää uuden alaluokituksen.

Metodi käyttää tietokantaoperaatiota, joka päivittää taulua Ala.

**public boolean submitLausunto(int kayttajaId, int artikkeliId, int versioNro, Lausunto kirjoittajaLausunto, Lausunto toimitusLausunto)**

Päivittää lausunnon muuttuneet tiedot. Jos lausuntoa ei löydy tietokannasta, lisätään se sinne.

Metodi käyttää tietokantaoperaatiota, joka päivittää taulua Lausunto.

**public boolean tarkistaSahkoposti(String sahkoposti)**

Palauttaa tiedon löytyykö parametrina saatu sähköposti jo järjestelmästä.

**public boolean submitArtikkeli(int kayttajaId, Artikkelik kirjoittajaArtikkeli)**

Päivittää artikkelin muuttuneet tiedot. Jos artikkelia ei löydy tietokannasta, lisätään se sinne.

Metodi käyttää tietokantaoperaatiota, joka päivittää tauluja Artikkelik ja Versio.

**public List getAsiantuntijat(String hakusanat)**

Palauttaa Asiantuntija-oliolistan kaikista asiantuntijoista, joiden nimi- tai osaamisala-kenttä sisältää hakusanan.

**public List getArtikkelitLehteen(int nro, int vuosikerta)**

Palauttaa tiettyyn painokseen valitut artikkelit Artikkelik-listana

**public void setLausuntopyyntoTila(int artik\_id, int versionro, int kayt\_id, int tila)**

Asettaa lausuntopyyntötilan artikkelille.

**public void insertAsiantuntijaArtikkelille(int artik\_id, int versionro, int kayt\_id)**

Lisää artikkelille asiantuntijan.

**public void removeAsiantuntijaArtikkelilta(int artik\_id, int versionro, int kayt\_id)**

Poistaa artikkelilta asiantuntijan.

### 5.3 SessionData

Jokaisen sisäänkirjautumisen yhteydessä luotava olio, joka sisältää käyttäjän istuntokohtaiset muuttujat. Luokan avulla voidaan varmistaa käyttöoikeudet eri operaatioita tehdessä. Lisäksi luokan apumuuttujien avulla voidaan vähentää tietokantahakuja. Olio on olemassa koko istunnon ajan sessiossa, mistä se saadaan käyttöön joka sivulle automaattisesti.

*Konstruktorit*



**public SessionData(String sessionId, int kayttajaId, int rooli)**

Luo ilmentymän SessionData-luokasta ja sijoittaa parametrit kenttien arvoksi.

*Metodit*

**public int getKasiteltavaSivu()**

Palauttaa kasiteltavaSivu-muuttujan arvon.

**public int getKasiteltavaArtikkeli()**

Palauttaa kasiteltavaArtikkeli-muuttujan arvon.

**public int getKayttajaId()**

Palauttaa kayttajaId-muuttujan arvon.

**public int getRooli()**

Palauttaa rooli-muuttujan arvon.

**public String getSessionId()**

Palauttaa sessionId-muuttujan arvon.

**public int getVersioNro()**

Palauttaa versioNro-muuttujan arvon.

**public void setKasiteltavaSivu(int kasiteltavaSivu)**

Asettaa kasiteltavaSivu-muuttujalle uuden arvon.

**public void setKasiteltavaArtikkeli(int kasiteltavaArtikkeli)**

Asettaa kasiteltavaArtikkeli-muuttujalle uuden arvon.

## 5.4 Event

*Luokkavakiot*

int LISAA\_ASIAANTUNTIJA = 1

int POISTA\_ASIAANTUNTIJA = 2

int KIRJAUDU = 3

int KIRJAUDU\_ULOS = 4

int REKISTEROIDY = 5

etc.

*Konstruktorit*

*Metodit*

## 5.5 HtmlTyokalu

Luokan metodien avulla saadaan usein toistuvista HTML-elementeistä (artikkelilistat, käyttäjälisat jne.) valmis HTML-esitys.

*Konstruktorit*

*Metodit*

### **public String getKirjoittajat()**

Palauttaa kirjoittajalistan HTML-esityksen.

Metodi käyttää tietokantakyselyä, joka hakee kaikki rivit taulusta Kayttaja.

### **public String getArtikkelit()**

Palauttaa artikkelilistan HTML-esityksen.

Metodi käyttää tietokantakyselyä, joka hakee kaikki rivit taulusta Artikkelit.

## 5.6 TiedostoServlet

Servlet-olio, jonka avulla sivulla voidaan näyttää tiedostoja, joihin käyttäjällä ei ole suoraa pääsyä. Lisäksi luokan avulla voidaan rajoittaa esimerkiksi tiedostoon kohdistuvia lukemiskertoja. Sivun saa parametriksi `kayt_id`, `artik_id`, `versionro`, `tiedostonimi` ja `liitenro`. Näillä parametreilla löydetään oikea tiedosto ja lisäksi tarkistetaan, että tiedoston nimi on annettu täsmälleen oikein. Servlet lopulta palauttaa binääriesityksen tiedostosta.

*Konstruktorit*

*Metodit*

## 5.7 Tiedosto

TiedostoServletin apuluokka, jonka avulla voidaan ladata tiedostoja ei-julkisista hakemistoista ohjelman käyttöön.

*Konstruktorit*

*Metodit*

### **public boolean poistaTiedosto(String polku)**

Poistaa halutun tiedoston.

**public boolean tallennaTiedosto(String nimi, String polku, HttpRequest request)**

Tallettaa HTML-lomakkeelta tulevan tiedoston haluttuun paikkaan.

**public BufferedOutputStream naytaTiedosto(String polku)**

Palauttaa tiedoston bittivirtana

## 5.8 Postittaja

Luokan avulla voidaan lähettää käyttäjille sähköpostia.

*Konstruktorit*

*Metodit*

**public static boolean laheta(Sahkoposti lahetettava)**

Lähettää viestin.

## 5.9 Salaja

Luokan avulla salataan salasanoja ja muuta salattavaa tietoa. Salasanojen salaamiseen käytetään Javan security-pakkauksesta löytyvää MessageDigest-luokkaa.

*Konstruktorit*

*Metodit*

**public String salaa(String teksti)**

Palauttaa parametrilla saadun tekstin MD5-algoritmillä salatussa muodossa.

## 5.10 DBConnector

Luokan avulla otetaan yhteyttä tietokantaan ja suoritetaan tietokantakyselyjä.

*Luokkavakiot*

String TUNNUS = tietokannan käyttäjätunnus

String SALASANA = tietokannan salasana

String AJURI = käytettävän tietokanta-ajurin tyyppi

String URL = tietokannan osoite

*Konstruktorit***public DBConnector()**

Ottaa yhteyden tietokantaan luokkavakioissa määritellyillä tunnuksilla.

*Metodit***public Tulos execute(String kysely)**

Tekee tietokantakyselyn ja palauttaa tuloksen Tulos-oliona.

**public boolean close()**

Sulkee yhteyden tietokantaan.

**public int getId(String sekvenssi)**

Palauttaa pyydetyn sekvenssin seuraavan arvon

## 5.11 Tulos

Oma tietorakenne tietokantakyselyiden tulosten tallentamiseen. Sisältää ResultSet-luokan metodeita vastaavat metodit, mutta luokan avulla voidaan käyttää aikaisempia tuloksia hyväksi tehtäessä uusia kyselyjä.

*Konstruktorit***public Tulos(ResultSet resultSet)**

Luo ResultSet-oliota vastaavan Tulos-olion.

*Metodit***public boolean hasNext()**

Kertoo onko Tulos-oliossa vielä rivejä.

**public void next()**

Siirtyy Tulos-olion seuraavalle riville.

**public String getString(String sarake)**

Palauttaa käsiteltävän rivin "sarake-nimisen sarakkeen arvon String-oliona.

**public int getInt(String sarake)**

Palauttaa käsiteltävän rivin "sarake-nimisen sarakkeen arvon int-arvona.

**public Date getDate(String sarake)**

Palauttaa käsiteltävän rivin "sarake-nimisen sarakkeen arvon Date-oliona.

**public int getRivit()**  
Palauttaa rivien lukumäärän.

## 5.12 Sähköposti

*Konstruktorit*

**Sähköposti()**  
Luo uuden sähköposti-olion.

**Sähköposti(String lahettaja, String vastaanottaja, String otsikko, String teksti, List liitetiedostot)**  
Luo uuden sähköposti-olion.

*Metodit*

**public String getLahettaja()**  
Palauttaa lahettaja-muuttujan arvon.

**public String getVastaanottaja()**  
Palauttaa vastaanottaja-muuttujan arvon.

**public String getOtsikko()**  
Palauttaa otsikko-muuttujan arvon.

**public String getTeksti()**  
Palauttaa teksti-muuttujan arvon.

**public List getLiitetiedostot()**  
Palauttaa liitetiedostot-muuttujan arvon.

## 5.13 Paivamaara

Luokka päivämäärien käsittelyyn.

*Konstruktorit*

**public Paivamaara()**  
Luo uuden päivämäärä-olion, jonka kenttien arvoksi tulee olion luontihetken päivämäärä.

**public Paivamaara(Date pvm)**

Luo Date-oliota pvm vastaavan Paivamaara-olion.

**public Paivamaara(String pvm)**

Luo String-oliota pvm vastaavan Paivamaara-olion.

**public Paivamaara(String paiva, String kuukausi, String vuosi)**

Luo Paivamaara-olion annetuilla arvoilla.

*Metodit***public String pvmKantaan()**

Tulostaa päivämäärän muodossa, joka voidaan tallentaa tietokantaan.

**public String pvmSivulle()**

Tulostaa päivämäärän muodossa, joka voidaan näyttää käyttöliittymässä.

**public String getPaiva()**

Palauttaa paiva-muuttujan arvon.

**public String getKuukausi()**

Palauttaa kuukausi-muuttujan arvon.

**public String getVuosi()**

Palauttaa vuosi-muuttujan arvon.

## 5.14 Lausunto

*Luokkavakiot*

int HYLATTY = 1 int HYVAKSYTTY\_SUURIN\_MUUTOKSIN = 2 int HYVAKSYTTY\_PNIENIN\_MUUTOKSIN = 3 int HYVAKSYTTY = 4

*Konstruktorit*

Konstruktorit käyttävät tietokantakyselyitä, jotka hakevat taulusta Kayttaja rivin attribuuttien kayt\_id perusteella, hakevat taulusta Lausunto rivin attribuuttien artik\_id, versionro ja kayt\_id perusteella sekä liittävät taulut Lausunto ja Liite attribuuttien artik\_id, versionro ja kayt\_id perusteella.

*Metodit*

**public String getAsiantuntija()**  
Palauttaa asiantuntija-muuttujan arvon.

**public List getTiedostot()**  
Palauttaa tiedostot-muuttujan arvon.

**public String getTeksti()**  
Palauttaa teksti-muuttujan arvon.

**public int getPaatos()**  
Palauttaa paatos-muuttujan arvon.

Set-metodit käyttävät tietokantaoperaatioita, jotka päivittävät tauluja Lausunto ja Liite attribuuttien artik\_id, versionro ja kayt\_id perusteella.

## 5.15 Asiantuntija

### *Konstruktorit*

Konstruktorit käyttävät tietokantakyselyitä, jotka hakevat taulusta Kayttaja rivin attribuutin kayt\_id perusteella ja liittävät taulut Asiantuntija ja Osaamisala attribuutin kayt\_id perusteella.

### *Metodit*

**public String[] getErikoisalat()**  
Palauttaa erikoisalat-muuttujan arvon.

**public Lausunto getLausunto()**  
Palauttaa lausunto-muuttujan arvon.

**public String getEmail()**  
Palauttaa email-muuttujan arvon.

Set-metodit käyttävät tietokantaoperaatioita, jotka päivittävät taulua Osaamisala attribuutin kayt\_id perusteella.

## 5.16 Artikkelit

### *Konstruktorit*

**public Artikkel**(int artikkeliId, String otsikko, Paivamaara pvm, String kirjoittaja, int tila, Url teksti, int versio, Asiantuntija[] asiantuntijat)

Luo uuden Artikkel-metodin ja alustaa sen kentät parametrien arvoilla.

Konstruktorit käyttävät tietokantakyselyä, joka liittää taulut Artikkel ja Versio attribuutin artik\_id perusteella.

*Metodit*

**public int getArtikkeliID()**

Palauttaa artikkeliID-muuttujan arvon.

**public String getOtsikko()**

Palauttaa otsikko-muuttujan arvon.

**public Paivamaara getPvm()**

Palauttaa pvm-muuttujan arvon.

**public String getKirjoittaja()**

Palauttaa kirjoittaja-muuttujan arvon.

**public int getTila()**

Palauttaa tila-muuttujan arvon.

**public String getTiivistelma()**

Palauttaa tiivistelma-muuttujan arvon.

**public Url getTeksti()**

Palauttaa teksti-muuttujan arvon.

**public int getVersio()**

Palauttaa versio-muuttujan arvon.

**public String getLausunto()**

Palauttaa lausunto-muuttujan arvon.

**public Asiantuntija[] getAsiantuntija()**

Palauttaa asiantuntija-muuttujan arvon.

Set-metodit käyttävät tietokantaoperaatioita, jotka päivittävät tauluja Artikkel ja Versio attribuuttien artik\_id ja versionro perusteella.



## 5.17 Kayttaja

### *Luokkavakiot*

int KAYTTAJA = 1 int ASIANTUNTIJA = 2 int TOIMITTAJA = 3 int PAATOIMITTAJA = 4

### *Konstruktorit*

**public Kayttaja(int kayttajaId, String etunimi, String sukunimi, String kaupunki, int postinumero, String maa, String sahkoposti, int rooli, List erikoisalajat)**

Luo uuden Kayttaja-olion ja alustaa sen kentät parametrien arvoilla.

Konstruktorit käyttävät tietokantakyselyä, joka hakee taulusta Kayttaja rivin attribuutin kayt\_id perusteella.

### *Metodit*

**public int getKayttajaID()**

Palauttaa kayttajaID-muuttujan arvon.

**public String getEtunimi()**

Palauttaa etunimi-muuttujan arvon.

**public String getSukunimi()**

Palauttaa sukunimi-muuttujan arvon.

**public String getKatuosoite()**

Palauttaa katuosoite-muuttujan arvon.

**public String getKaupunki()**

Palauttaa kaupunki-muuttujan arvon.

**public String getOsavaltio()**

Palauttaa osavaltio-muuttujan arvon.

**public int getPostinumero()**

Palauttaa postinumero-muuttujan arvon.

**public String getMaa()**

Palauttaa maa-muuttujan arvon.

**public String getEmail()**

Palauttaa email-muuttujan arvon.

**public int getRooli()**

Palauttaa rooli-muuttujan arvon.

**public List getErikoisala()**

Palauttaa erikoisala-muuttujan arvon.

Set-metodit käyttävät tietokantaoperaatioita, jotka päivittävät taulua Kayttaja attribuutin kayt\_id perusteella.

## 5.18 Alat

*Konstruktorit*

*Metodit*

**public int getAlaID()**

Palauttaa alaID-muuttujan arvon.

**public String getAlanNimi()**

Palauttaa alanNimi-muuttujan arvon.

## 6 JSP-sivut

### 6.1 Kaikkien käyttäjien sivut

#### 6.1.1 kehys.jsp

Pääsivu, joka määrittelee sivun kehyksien asettelut. Sisältää kaksi kehystä: sivun vasemman reunaan navigointivalikon (menu.jsp) ja pääkehysten. Pääkehykseen ladataan aluksi käyttäjän roolin mukaan omatartikkelit.jsp (kirjoittaja), artikkelienarvostelu.jsp (asiantuntija), artikkelienhallinta.jsp (toimittaja) tai paatoimittajanakyma.jsp (päätoimittaja).

#### 6.1.2 menu.jsp

Sivun vasemmassa laidassa oleva navigointivalikko. Valikon linkit johtavat suoraan eri jsp-sivuille ja se mukautuu käyttäjän roolin mukaan. Kirjoittajalle näytetään kirjoittajan jsp-sivut, asiantuntijalle kirjoittajan ja asiantuntijan sivut sekä toimitukselle kaikki.

### 6.1.3 kirjautuminen.jsp

Ensimmäisenä aukeava sivu. Sisältää e-mail- ja salasanakentät. Kenttien tiedot tarkistetaan ”Login”-napin painamisen jälkeen. Onnistuneen kirjautumisen jälkeen käyttäjä ohjataan oikealle JSP-sivulle käyttäjäryhmänsä mukaan.

### 6.1.4 profiilinmuokkaus.jsp

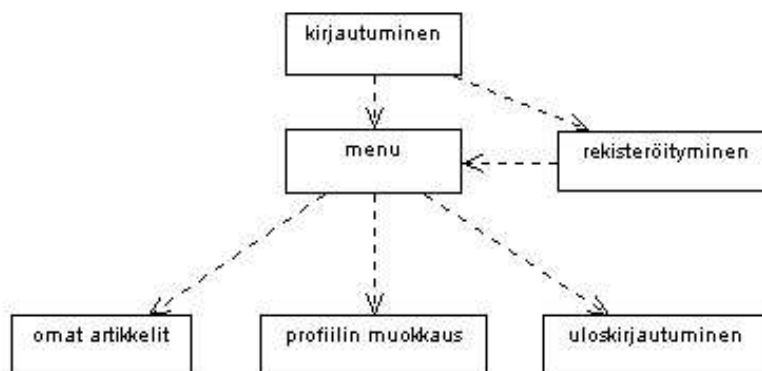
Sivulle ladataan lomake, joka on täytetty valmiiksi käyttäjän tiedoilla. Käyttäjän tehtyä muutokset ja painettua ”Update profile” -painiketta muuttuneet tiedot viedään tietokantaan ja käyttäjä saa ilmoituksen päivityksen onnistumisesta.

Lomakkeen ulkoasu muuttuu käyttäjän ryhmän mukaan, esimerkiksi asiantuntija voi muokata omia erikoisalojaan sivun kautta.

### 6.1.5 uloskirjautuminen.jsp

Suljetaan käyttäjän istunto ja poistetaan istuntoon liittyvät muuttujat muistista. Tulostetaan käyttäjälle hyvästelyviesti.

## 6.2 Kirjoittajan sivut



Kuva 4: Kirjoittajan navigointi.

### 6.2.1 rekisteroityminen.jsp

Lomake, jonka avulla käyttäjä voi rekisteröityä kirjoittajaksi järjestelmään. Lomakkeen täyttämisen jälkeen tarkistetaan kenttien oikeellisuus ja tallennetaan tiedot tietokantaan. Tämän jälkeen käyttäjä ohjataan sisäänkirjautumissivulle.

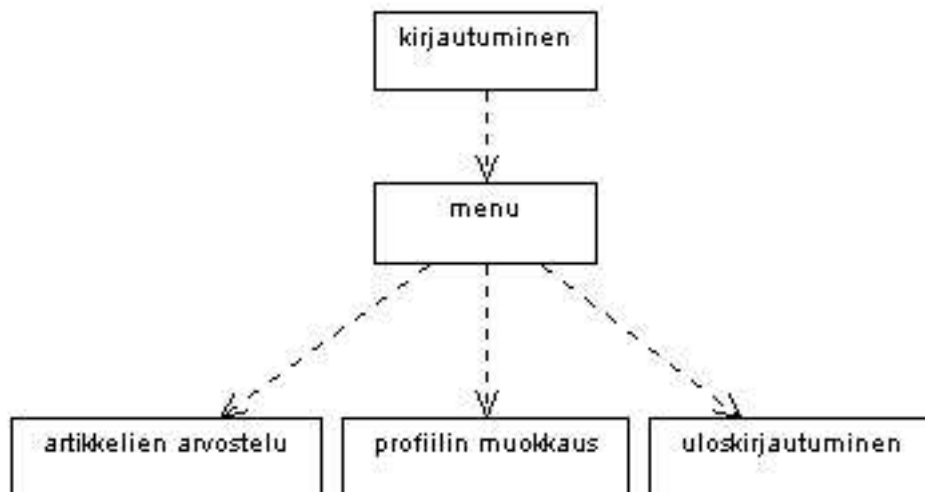
### 6.2.2 omatartikkelit.jsp

Omat artikkelit -sivu sisältää HTMLTyokalulla tuotetun artikkelilistan. Jos parametriä artik id ei ole määritetty, tulostetaan lomake, jolla voidaan lähettää uusi artikkeli. Mikäli artik id on määritetty, näytetään kyseisen artikkelin tiedot.

Uusi artikkeli pitää voida lähettää myös PS- tai PDF-muodossa. Tätä varten sivulla on tekstikenttä, jonka avulla voidaan lähettää binäärimuotoisia tiedostoja. Lähetettyään uuden artikkelin käyttäjä saa ilmoituksen toimenpiteen onnistumisesta, ja artikkeli vietään tietokantaan.

Kirjoittaja voi järjestää artikkelit haluamaansa järjestykseen painamalla listan otsikkorivillä olevia linkkejä (otsikko, aihepiiri, lähetyspäivä).

## 6.3 Asiantuntijoiden sivut



Kuva 5: Asiantuntijan navigointi.

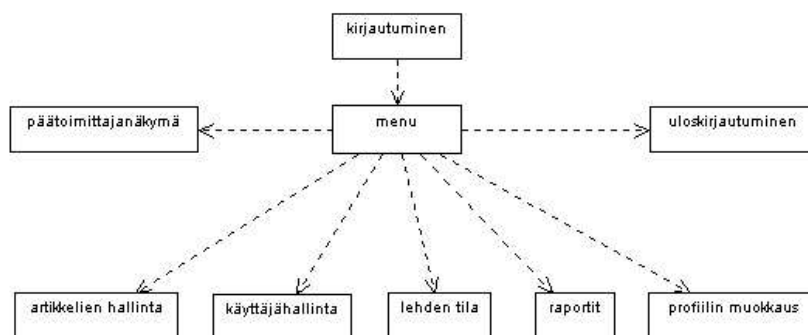
### 6.3.1 artikkelienarvostelu.jsp

Asiantuntija näkee listan artikkeleista, joihin hänet on merkattu arvostelijaksi. Klikkaamalla artikkelin otsikkoa, avautuu artikkelin informaatio sivu alakehykseen. Jos artikkeli on uusi, kysytään asiantuntijalta haluaako hän toimia kyseisen artikkelin arvostelijana. Kun asiantuntija painaa ”Yes”-painiketta, jää siitä jälki tietokantaan, ja artikkeli siirtyy tilaan, jossa asiantuntijalta odotetaan palautetta.

Jos asiantuntija on hyväksynyt artikkelin arvosteltavaksi ja painaa linkkiä kyseisen artikkelin kohdalta, saa hän arvostelulomakkeen täytettäväksi. Painettuaan ”send review”-painiketta lausunto siirtyy tietokantaan ja artikkeli muuttuu arvostelluksi asiantuntijan osalta. Painaessaan linkkiä arvostellun artikkelin kohdalta asiantuntija näkee artikkelin tiedon ja antamansa palautteen.

Asiantuntija voi järjestää artikkelit haluamaansa järjestykseen painamalla listan otsikkorivillä olevia linkkejä (otsikko, aihepiiri, lähetyspäivä).

## 6.4 Toimituksen sivut



Kuva 6: Toimituksen navigointi.

### 6.4.1 artikkelienhallinta.jsp

Sivulla näkyy oletuksena luettelo artikkeleista, jotka odottavat toimituksen toimenpiteitä. Luettelon yhteydessä on myös linkki, jota painamalla saadaan luettelon kaikista artikkeleista.

Kun artikkelin linkkiä painetaan, avautuu alempaan kehykseen artikkelin tiedot. Sivu sisältää artikkelin perustiedot (sekä linkin varsinaiseen artikkeliin), asiantuntijoiden valintalomakkeen sekä viestilaatikon päätoimittajalle.

Toimittaja voi valita artikkelille asiantuntijat valintalistassa. Kun toimittaja valitsee vasemmalla puolella olevasta luettelosta haluamansa asiantuntijan artikkelille ja painaa ”add referee to article” -linkkiä, siirtyy hänen valitsemansa asiantuntija oikeanpuoleen luetteloon, jossa on lueteltu artikkelille valitut asiantuntijat. Vastaavasti toimittaja voi poistaa artikkelin asiantuntijoita artikkelista ”remove referee from article” -linkistä. Jos asiantuntija puuttuu luettelosta, voidaan uusi lisätä samalta sivulta löytyvällä lomakkeella. Toimittajan täydennettyä asiantuntijan tiedot ja painettua ”add referee” -nappia, arvot viedään tietokantaan ja asiantuntija ilmestyy asiantuntijaluetteloon valittavaksi.

Jos toimittaja haluaa tiedustella päätoimittajalta artikkeliin liittyvää asiaa, voi hän kirjoittaa viestin sille varattuun laatikkoon.

Jos artikkeli on käynyt arvostelukierroksen läpi, avautuu alempaan kehykseen asiantuntijoiden palautteet. Toimittaja näkee sivulta tulleet palautteet ja hän voi tarvittaessa muokata niitä. Lisäksi toimittaja voi hallita liitetiedostoja sivun avulla. Painaessaan ”save and send information to writer” -nappia ohjataan kirjoittajalle osoitetut palautteet ja liitetiedostot Sähköposti-oliona sähköposti.jsp-sivulle.

#### **6.4.2 lehdentila.jsp**

Lehden tila -sivulla toimittajalle aukeaa luettelo artikkeleista, joita ei ole vielä lisätty mihinkään lehteen, mutta joihin on saatu asiantuntijoilta arviot. Sivulla on pudotusvalikko lehden numeroa varten. Kun käyttäjä valitsee jonkun numeron, näytetään kyseiseen lehteen valitut artikkelit sivun alalaidassa olevassa luettelossa.

Sivun kautta voidaan lähettää myös muistutuksia arvostelijoille Painamalla ”send reminder” -painiketta. Kun muistutusviesti on lähetetty, tallennetaan tietokantaan aikaleima muistutuksen lähetyksestä.

#### **6.4.3 kayttajahallinta.jsp**

Sivun yläreunaan avautuu järjestettävissä oleva luettelo (joka saadaan HTMLTyokalu-luokalta) käyttäjistä. Painamalla käyttäjän nimeä avautuu alakehykseen käyttäjän tiedoilla täydennetty lomake, johon toimittaja voi tehdä muutoksia. ”Update fields” -painiketta painamalla tiedot siirtyy tietokantaan.

#### **6.4.4 raportit.jsp**

Raportit-sivu on staattinen sivu, johon on kerätty erilaisia tietokantakyselyitä.

#### **6.4.5 sähköposti.jsp**

Sähköposti-sivulla toimittaja voi lähettää sähköpostia järjestelmän sidosryhmille. Oletuksena toimittajalle aukeaa tyhjä sivu, johon hän voi täyttää tarvittavat tiedot. Painaessaan

”send”-painiketta kentän tiedot siirtyvät Sähköposti-oliona Postittaja-luokalle, joka lähettää viestin eteenpäin. Sivua voidaan kutsua myös Sähköposti-muotoisella parametrilla, jolloin lomake täytetään kyseisen olion sisältämällä arvoilla.

#### 6.4.6 päätoimittajanakyma.jsp

Päätoimittajan näkymä vastaa muun toimituksen näkymää artikkeleiden hallintaan, mutta päätoimittajalle lähetettävän viestikentän sijaan päätoimittajalla on mahdollisuus vastata hänelle lähetettyihin viesteihin. Päätoimittajan artikkeliluettelossa näytetään oletuksena vain ne artikkelit, jotka odottavat päätoimittajan kommentointia ja asiantuntijaehdotuksia.

## 7 Tietokantakuvaus

Tässä luvussa esitellään tietokannan suunnitteluun liittyvän yleisen kuvauksen lisäksi tietokantataulujen keskinäiset suhteet (kuva 7) ja luokkien yksityiskohtaiset luokkakuvaukset.

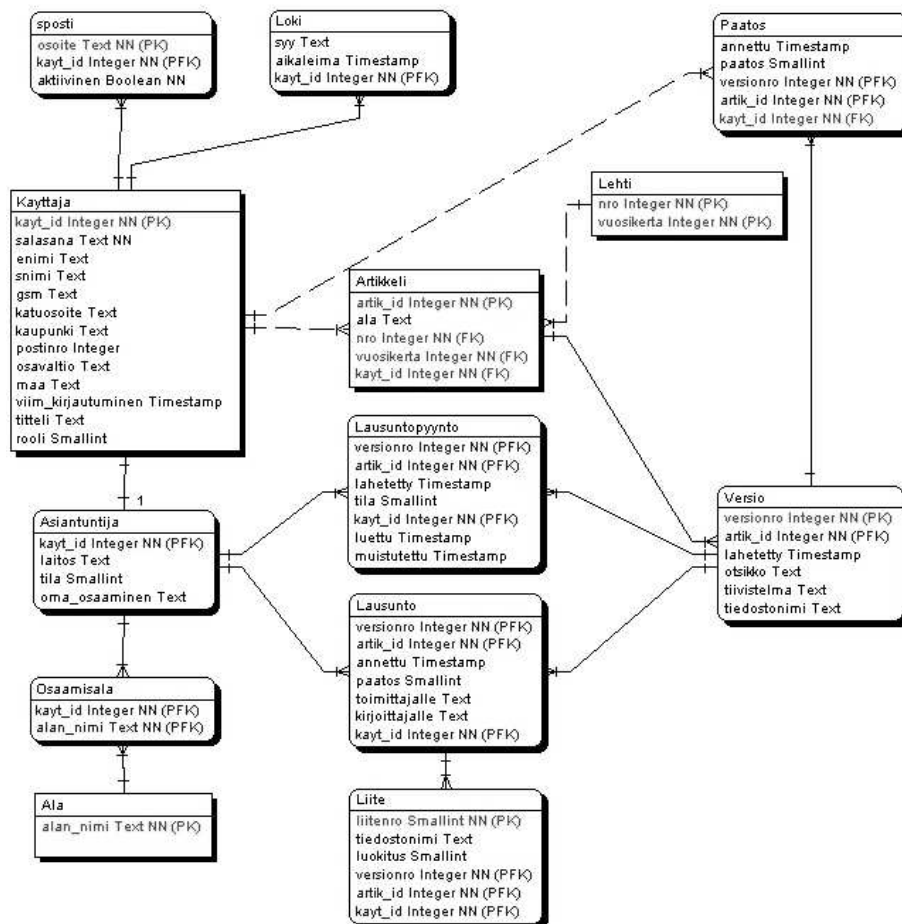
### 7.1 Tietokantakyselyt ja transaktioiden sitoutuminen

Tietokantaa ja sen käyttöä suunniteltaessa lähtökohtana on ollut tietokannan eheyden säilyttäminen virhetilanteissa. Järjestelmän kaatuessa esimerkiksi laitteistovirheen vuoksi tietoa saatetaan menettää, sillä tietokantaan kohdistuvat päivitysoperaatiot pidetään mahdollisimman atomisina, ts. ne joko suoritetaan kokonaan tai ei ollenkaan.

Käyttäjän tekemät päivitykset tehdään tietokantaan vasta siinä vaiheessa, kun kaikki tarvittavat tiedot on saatu. Käytännössä siis käyttäjän käyttöliittymäsivuilla syöttämät tiedot tallennetaan tietokantaan syötteiden oikeellisuuden tarkistamisen jälkeen, kun käyttäjä on lähettänyt lomakkeen tiedot järjestelmälle. Virheellisten syötteiden tapauksessa käyttöliittymäsivu asianmukaisine virheilmoituksineen näytetään käyttäjälle uudestaan, mutta päivityksen tuottava transaktio ei ala eikä sitoudu ennen korjauksia. Transaktiot pidetään siis mahdollisimman lyhyinä, jolloin suurin osa likaisista luvuista ja kirjoituksista saadaan estettyä.

Joissakin tapauksissa anomaliaita saattaa kuitenkin esiintyä. Käyttöliittymäsivujen generoimiseen voidaan nimittäin käyttää kahden eri tietokantakyselyn tulosjoukkoa, jolloin kyselyiden välissä tapahtuvat muutokset eivät näy jälkimmäistä kyselyä suoritettaessa; operointi tapahtuu siis vanhentuneilla monikoiden arvoilla. Tämä mahdollistaa eräänlaiset toistokelvottomat luvut, mutta tietokanta pysyy silti eheässä tilassa. Esimerkiksi kirjoittajan artikkeleita haettaessa artikkelilista luodaan sisäkkäisten kyselyiden avulla: ulommaisessa silmukassa haetaan tiettyyn kirjoittajaan liittyvät artikkelit ja sisemmässä kyseiseen artikkeliin liittyvät lausunnot.

Koska yhdellä käyttäjällä käytössä olevien tietojen lukitseminen muilta käyttäjiltä ei pidea vartenotettavana ratkaisuna, on mahdollista että tietoa häviää päivitysten yhteydessä.



Kuva 7: Tietokantakaavio.

Kirjoittaja ja toimittaja saattavat esimerkiksi samaan aikaan päivittää kirjoittajan tietoja, jolloin vain myöhemmin sitoutuneen transaktion aiheuttamat muutokset jäävät voimaan.

Selainpohjaisen käyttöliittymän vuoksi tällainen vanhentuneen tiedon käyttäminen päivityksiin on väistämättä mahdollista, sillä näytöllä näkyvän tiedon ajankohtaisuus on kiinni käyttäjistä itsestään. Yllämainitut seikat on kuitenkin pyritty ottamaan suunnittelussa huomioon tietokannan eheyden takaamiseksi.



## 7.2 Käyttäjä

Attribuutti	Arvojoukko	Kuvaus
kayt id	luku	käyttäjän tunniste
salasana	teksti	käyttäjän salasana
enimi	teksti	etunimi
snimi	teksti	sukunimi
gsm	teksti	puhelinnumero
katuosoite	teksti	katuosoite
kaupunki	teksti	postitoimipaikka
postinro	luku	postinumero
maa	teksti	maa
osavaltio	teksti	osavaltio
arvo	teksti	arvonimi tai titteli
rooli	luku	kirjoittaja, (pää)toimittaja tai asiantuntija

Kirjoittaja-taulu sisältää järjestelmään rekisteröityneiden ja toimituksen lisäämien käyttäjien tiedot. Kaikilla rekisteröidyillä käyttäjillä on siis oikeus lähettää artikkeleita toimituksen ja asiantuntijoiden arvioitavaksi. Yhdellä kirjoittajalla voi olla yksi tai useampia sähköpostiosoitteita käytössään. Käyttäjä identifioidaan yksikäsitteisen käyttäjätunnisteen avulla.

## 7.3 Asiantuntija

Attribuutti	Arvojoukko	Kuvaus
kayt id	viite	käyttäjän tunniste
laitos	teksti	työpaikka (yliopisto tai muu oppilaitos)
oma osaaminen	teksti	vapaamuotoinen kuvaus osaamisaloista
tila	boolean	aktiivinen?

Asiantuntija on tavallisen käyttäjän, eli kirjoittajan erikoistapaus. Toimituksen lisäämällä asiantuntijalla on jokaiseen kirjoittajaan liittyvien tietojen ja identifioivan tunnuksen lisäksi laitos, jossa asiantuntija tällä hetkellä toimii, sekä rajoittamaton joukko osaamisaloja. Asiantuntijalla on hänelle arvioitavaksi lähetettyihin artikkeleihin liittyen joukko lausuntopyyntöjä niihin edelleen liittyviä lausuntoja. Asiantuntija voi valita, ettei halua toistaiseksi ottaa vastaan lausuntopyyntöjä, jolloin asiantuntijan tila muuttuu aktiivisesta poissaolevaksi.

## 7.4 Osaamisala

Attribuutti	Arvojoukko	Kuvaus
kayt id	viite	käyttäjän tunniste
alan nimi	viite	osaamisalan nimi

Osaamisala-taulu sisältää asiantuntijoiden osaamisalat.

## 7.5 Ala

Attribuutti	Arvojoukko	Kuvaus
alan nimi	teksti	tieteenhaaran nimi

Ala-taulu sisältää kaikki tällä hetkellä valittavana olevat tieteenalat, joista asiantuntijat valitsevat omaa osaamistaan parhaiten kuvaavat alat.

## 7.6 Sähköposti

Attribuutti	Arvojoukko	Kuvaus
kayt id	viite	käyttäjän tunniste
osoite	teksti	sähköpostiosoite
aktiivinen	boolean	aktiivinen tai passiivinen

Käyttäjällä voi olla useita sähköpostiosoitteita, joista käyttäjän kulloinkin valitsema toimii senhetkisenä aktiivisena osoitteena. Käyttäjää koskevat tiedotukset käsittelyn etene- misestä, uusista lausuntopyyntöistä ym. lähetetään aktiiviseen osoitteeseen.

## 7.7 Lehti

Attribuutti	Arvojoukko	Kuvaus
nro	luku	lehden numero
vuosikerta	luku	vuosikerta

Lehti koostuu määrittelemättömän monesta artikkelista ja se identifioidaan volyymin, eli lehden numeron ja vuosikerran, avulla.

## 7.8 Artikkel

Attribuutti	Arvojoukko	Kuvaus
artik id	luku	artikkelin tunniste
kayt id	luku	kirjoittajan tunniste
ala	teksti	artikkeliin liittyvä tieteenhaara
nro	viite	lehden numero
vuosikerta	viite	lehden vuosikerta

Uudesta artikkelista luodaan rivi Artikkel- ja Versio-tauluihin myöhempää versionhallintaa varten, ja se identifioidaan yksikäsitteisellä artikkelitunnuksella. Artikkelin liittyvät yleensä useita uusintakierrosten ja korjausten perusteella syntyneitä versioita. Yhteistä kaikille artikkelin versioille on niihin liittyvä tieteenala.

## 7.9 Versio

Attribuutti	Arvojoukko	Kuvaus
artik id	viite	artikkelin tunniste
versionro	luku	version numero
lahetetty	aikaleima	lähettämishetki
otsikko	teksti	otsikko
tiivistelmä	teksti	raakamuotoinen tiivistelmä
tiedostonimi	teksti	viittaus artikkelitiedostoon

Jokaisella artikkelin versiolla on oma versionumeronsa, jonka avulla versio identifioidaan. Uudelleenlähetyksen yhteydessä sisällön lisäksi artikkelin otsikko ja raakatekstimuotoinen tiivistelmä saattavat muuttua. Yhden artikkelin tiettyyn versioon liittyy omat lausuntopyyntönsä ja lausuntonsa, ja eri saman artikkelin eri versioita voi olla arvioimassa eri asiantuntijoita.

## 7.10 Lausuntopyyntö

Attribuutti	Arvojoukko	Kuvaus
artik id	viite	artikkelin tunniste
kirj id	viite	asiantuntijan tunniste
versionro	viite	version numero
lahetetty	aikaleima	pyynnön lähettämishetki
tila	teksti	asiantuntija valittu (<0), odottaa vastausta (0), hylätty (1), hyväksytty (2)
luettu	aikaleima	pyynnön lukemishetki
muistutettu	aikaleima	muistutuspyynnön lähettämishetki

Lausuntopyyntö koskee tiettyä artikkelin versiota, ja se identifioidaan artikkelin ja asiantuntijan tunnuksilla sekä versionumerolla. Lausuntopyyntöllä on kolme mahdollista tilaa: hyväksytty, hylätty tai odottaa vastausta. Asiantuntijan lukiessa hänelle lähetetyn lausuntopyyntönsä ensimmäistä kertaa lukemishetkestä otetaan aikaleima tilastointia ja käsittelytilanteen seuraamista varten.

## 7.11 Lausunto

Attribuutti	Arvojoukko	Kuvaus
kirj id	viite	asiantuntijan tunniste
artik id	viite	artikkelin tunniste
versionro	viite	version numero
annettu	aikaleima	lausunnon antamishetki
toimitukselle	teksti	toimitukselle menevä lausunto
kirjoittajalle	teksti	kirjoittajalle menevä lausunto
paatos	numero	hyväksytty (0), hyväksytty suurin muutoksin (1), hyväksytty pienin muutoksin (2), hylätty (3)

Lausunto koskee tiettyä artikkelin versiota, ja se identifioidaan lausuntopyyntönsä tavoin. Lausunnolla on neljä mahdollista tilaa: hyväksytty, hyväksytty pienin muutoksin, hyväksytty suurin muutoksin tai hylätty. Lausuntoon liittyy määrittelemättömän monta liitettä, jotka voivat olla joko kuvia tai tekstiä. Lausunnon antamishetkestä otetaan aikaleima tilastointia ja käsittelyn seuraamista varten.

## 7.12 Liite

Attribuutti	Arvojoukko	Kuvaus
kirj id	viite	asiantuntijan tunniste
artik id	viite	artikkelin tunniste
versionro	viite	version numero
liitenro	viite	liitteen numero
tiedostonimi	teksti	viittaus liitetiedostoon
luokitus	boolean	sekä kirjoittajalle että toimitukselle?

Liite liittyy yhteen lausuntoon, ja se identifioidaan artikkelin ja asiantuntijan tunnusten sekä versio- ja liitenumeron avulla. Asiantuntija määrittelee jokaiselle liitteelle näkyvyysluokituksen; osa liitteistä on tarkoitettu vain toimituksen käyttöön.

## 7.13 Päätös

Attribuutti	Arvojoukko	Kuvaus
artik id	viite	artikkelin tunniste
kayt id	viite	toimittajan tunniste
versionro	viite	versionumero
annettu	aikaleima	päätöksen antamishetki
paatos	numero	hyväksytty (0), hyväksytty suurin muutoksin (1), hyväksytty pienin muutoksin (2), hylätty (3)

## 7.14 Loki

Attribuutti	Arvojoukko	Kuvaus
kirj id	viite	käyttäjän tunniste
aikaleima	aikaleima	kirjattavan toiminnon tapahtumishetki
syy	teksti	kirjattava toiminto

Loki-tauluun kirjataan sekä kirjoittajien että asiantuntijoiden toimintoja. Raporteissa useasti käytetyt aikaleimat pidetään kyselyiden helpottamiseksi niihin liittyvissä tauluissa, ja lokiin kirjataan kyseisten aikaleimojen lisäksi kaikki toimitusta mahdollisesti kiinnostavat tapahtumat. Tällaisia ovat esimerkiksi asiantuntijoiden sisäänkirjautumiset, lausunto-pyyntöjen selailut ja pyyntöihin vastaamiset.

## 7.15 Järjestelmään liittyvien tiedostojen tallennus

Järjestelmässä tallennetaan tietokannan lisäksi dataa myös tiedostoihin. Jokaiselle järjestelmän asiantuntijalle luodaan hakemistoon *liitteet* alihakemisto, jonka nimi on *ID(kayt id)*.

Asiantuntijoiden lausuntoihin liittyvät liitetiedostot tallennetaan asiantuntijoiden henkilökohtaisiin hakemistoihin seuraavasti:

A(artik id)V(versionro)L(liitenro).(tiedostopääte)

Siis esimerkiksi hakemistossa *liitteet/ID84/* olevasta tiedostosta *A110V3L2.pdf* voitaisiin päätellä, että se on asiantuntijan numero 84 PDF-muotoinen liite numero 2 ja se liittyy artikkelin numero 110 kolmannen version lausuntoon. Liitetiedoston muoto ei ole rajattu, ja se voi PDF-muodon lisäksi olla esimerkiksi JPEG, PS tai GIF.

Artikkelin eri versiot tallennetaan tiedostoon *versiot/kayt id/* seuraavassa muodossa:

A(artik id)V(versionro).(tiedostopääte)

Esimerkiksi *versiot/ID56/A2341V4.ps* tarkoittaa artikkelin numero 2341 neljättä PS-muotoista versiota, jonka on kirjoittanut käyttäjä numero 56.

Kun artikkelien versio tai liitetiedosto kopioidaan palvelimeen, tiedostot tallennetaan väliaikaisesti hakemistoon *temp* edellä mainitun nimisinä. Tiedostojen siirtäminen oikeisiin hakemistoihin tapahtuu lähetyispainikkeen painamisen yhteydessä. Jos lähetyispainiketta ei paineta, tyhjennetään temp-hakemisto uloskirjautumisen yhteydessä.

Erillinen lähetyispainike tarvitaan, jotta asiantuntija voi ladata useita liitetiedostoja. Lähetyispainike on myös tarpeellinen silloin, kun hitaan Internet-yhteyden takia tiedoston lataaminen palvelimelle kestää kauan.

## 8 Käyttöliittymä

Järjestelmällä on kolme käyttäjäryhmää: kirjoittajat, asiantuntijat ja toimitus, ja siten myös kolme erillistä käyttöliittymää.

## 8.1 Kaikki käyttäjäryhmät

### 8.1.1 Kirjautuminen

Järjestelmään rekisteröitynyt käyttäjä syöttää sähköpostiosoitteensa ja salasansa niille varattuihin kenttiin. Painamalla ”Login” -painiketta käyttäjä kirjautuu sisään järjestelmään. Käyttäjän ollessa 1) kirjoittaja, hän siirtyy ”omat artikkelit” -näkyseen, 2) asiantuntija, hän siirtyy ”artikkelien arvostelu” -näkyseen, 3) toimittaja, hän siirtyy ”artikkelien hallinta” -näkyseen tai 4) päätoimittaja, hän siirtyy ”päätoimittaja”-näkyseen. Näkymän vasemmassa reunassa on kaikilla käyttäjillä valikko, jonka avulla siirrytään käyttäjän eri näkymien välillä.

Uusi käyttäjä voi rekisteröityä järjestelmän kirjoittajaksi ”Register new user” -painiketta painamalla, jolloin hän siirtyy ”rekisteröityminen” -näkyseen. Uusien asiantuntijoiden ja toimittajien rekisteröinnin hoitaa lehden toimitus.

#### Virheenkäsittely

- Jos sähköpostiosoitetta ei löydy tietokannasta, tai jos salasana ei vastaa sähköpostiosoitteen salasanaa, annetaan virheilmoitus ”Login failed: wrong email or password.”

## 8.2 Kirjoittaja

Kirjoittajan näkymän vasemman reunan valikossa on linkit näkyisiin ”omat artikkelit” ja ”profiilin muokkaus” sekä linkki ”uloskirjautuminen”.

### 8.2.1 Rekisteröityminen

Käyttäjä syöttää kenttiin etunimen, sukunimen, tittelin, katuosoitteen, kaupungin, postinumeron, osavaltion, maan, puhelinnumeron, sähköpostiosoitteen sekä salasanan kahteen kertaan niille varattuihin kenttiin. Painamalla ”Create profile” -painiketta hän rekisteröityy kirjoittajaksi ja sisäänkirjautuu automaattisesti.

#### Virheenkäsittely

- Jos etunimi, sukunimi, sähköpostiosoite tai salasana on tyhjä, annetaan virheilmoitus ”You must specify first name / last name / e-mail / password”.
- Jos sähköpostiosoite ei ole oikeaa muotoa (@-merkki puuttuu), annetaan virheilmoitus ”Must have @-symbol in e-mail address”.
- Jos sähköpostiosoite löytyy jo tietokannasta, annetaan virheilmoitus ”E-mail address already registered”.

- Jos syötetty salasana ei ole tarpeeksi pitkä, annetaan virheilmoitus ”Password must be at least 5 characters”.
- Jos syötetty salasana ei vastaa uudelleensyötettyä salasanaa, annetaan virheilmoitus ”Passwords don’t match”.

### 8.2.2 Omat artikkelit

Näkymän yläreunassa on lueteltu kaikki kyseisen kirjoittajan järjestelmään lähettämät artikkelit. Luettelossa näkyy artikkelin otsikko, aihe, saapumispäivämäärä ja tila.

Artikkeliluettelon alapuolella on lomake, jolla kirjoittaja voi lähettää uuden artikkelin. Kirjoittaja syöttää artikkelin otsikon, aiheen, tiivistelmän ja artikkelin PS/PDF-muodossa niille varattuihin kenttiin ja painamalla ”Send article” -painiketta lähettää artikkelin. Uusi artikkeli siirtyy näkymän artikkeliluetteloon, ja käyttäjä saa kuittauksen lähetyksen onnistumisesta.

Lausuntokierrokselta valmistuneen artikkelin otsikkoa painamalla kirjoittaja voi katsoa millaista palautetta artikkeli on saanut. Palautteisiin kuuluvat tekstit ja liitetiedostot luettellaan asiantuntijoin (anonyymisti), ja tiedoston nimeä painamalla kirjoittaja voi avata kyseisen tiedoston. Jos artikkeli on hyväksytty pienin tai suurin muutoksin ja siitä on pyydetty uusi versio, kirjoittaja voi lähettää korjatun version artikkelista palautteen alapuolella olevalla lomakkeella. Kirjoittaja voi muokata artikkelin otsikkoa ja tiivistelmää niille varatuissa kentissä sekä syöttää artikkelin korjatun version PS/PDF-muodossa. Kirjoittajan painettua ”Send article” -painiketta korjattu versio siirtyy näkymän artikkeliluetteloon, ja kirjoittaja saa kuittauksen lähetyksen onnistumisesta. Jos artikkeli on hyväksytty ja siitä on pyydetty julkaistava versio, kirjoittaja voi syöttää artikkelin  $\text{\LaTeX}$ -muodossa sille varattuun kenttään ja lähettää sen ”Send article” -painiketta painamalla. Artikkelin tila muuttuu näkymän artikkeliluettelossa, ja kirjoittaja saa kuittauksen lähetyksen onnistumisesta.

### Virheenkäsittely

- Jos tiedostonlataamiskenttään on syötetty virheellinen polku, annetaan virheilmoitus ”Unknown path”.

### 8.2.3 Profilin muokkaus

Kirjoittaja voi muokata seuraavia tietojaan: etunimeä, sukunimeä, titteliä, katuosoitetta, kaupunkia, postinumeroa, osavaltiota, maata, puhelinnumeroa, sähköpostiosoitteita, aktiivista sähköpostiosoitetta ja salasanaansa niitä vastaavilla kentillä. ”Update profile” -painiketta painamalla muutokset tulevat voimaan, ja kirjoittaja siirtyy takaisin ”omat artikkelit” -näkykseen.

### Virheenkäsittely



- Jos etunimi, sukunimi, sähköpostiosoite tai salasana on tyhjä, annetaan virheilmoitus ”You must specify first name / last name / e-mail address / password”.
- Jos lisättävä sähköpostiosoite on väärää muotoa (@-merkki puuttuu), annetaan virheilmoitus ”Must have @-symbol in e-mail address”.
- Jos lisättävä sähköpostiosoite löytyy jo tietokannasta, annetaan virheilmoitus ”E-mail address already registered”.
- Jos käyttäjän ainoa sähköpostiosoite yritetään poistaa, annetaan virheilmoitus ”You must have at least one e-mail address”.
- Jos syötetty salasana ei ole tarpeeksi pitkä, annetaan virheilmoitus ”Password must be at least 5 characters”.
- Jos syötetty salasana ei vastaa uudelleensyötettyä salasanaa, annetaan virheilmoitus ”Passwords don’t match”.

### 8.3 Asiantuntija

Asiantuntijan näkymän vasemman reunan valikossa on linkit näkymiin ”artikkelien arvostelu” ja ”profiilin muokkaus” sekä linkki ”uloskirjautuminen”.

#### 8.3.1 Artikkelien arvostelu

Näkymän yläreunassa on lueteltu kaikki artikkelit, joihin kyseinen asiantuntija on valittu lausunnonantajaksi. Oletuksena luettelosta on valittu usin saapunut artikkeli. Luettelossa näkyy artikkelien otsikko, kirjoittaja, saapumispäivämäärä, aihe sekä tila. Artikkelin otsikkoa painamalla asiantuntija voi tutkia artikkelin tiivistelmää ja itse artikkelia.

Asiantuntija voi hyväksyä uuden lausuntopyynnön ensin valittuaan kyseisen artikkelin (artikkelin tila: ”New article”) ja sitten painamalla ”Yes”-painiketta, jolloin artikkelin tila muuttuu (”Not yet reviewed”), tai hylätä pyynnön painamalla ”No”-painiketta, jolloin artikkeli poistuu asiantuntijan näkymän artikkeliluettelosta.

Asiantuntija voi arvostella artikkelin ensin valittuaan kyseisen artikkelin ja sitten valitsemalla sopivan vaihtoehdoista ”Accepted”, ”Accepted with minor revision”, ”Accepted with major revision” tai ”Rejected” sekä lisäämällä toimitukselle ja kirjoittajalle omat palautteensa niille varattuihin tekstilaatikoihin sekä palautteen liitteet tiedostoina niille varattuihin kenttiin ja painamalla ”Send review” -painiketta. Artikkelin tila päivittyy näkymän artikkeliluetteloon, ja asiantuntija saa kiittauksen palautteen lähetyksen onnistumisesta. Asiantuntija voi muokata jo antamaansa palautetta, kunnes toimitus on antanut kyseisestä artikkelista lopullisen päätöksen.

#### Virheenkäsittely

- Jos tiedostonlataamiskenttään on syötetty virheellinen polku, annetaan virheilmoitus ”Unknown path”.

### 8.3.2 Profilin muokkaus

Asiantuntija voi muokata seuraavia tietojaan: etunimeä, sukunimeä, titteliä, katuosoitetta, kaupunkia, postinumeroa, osavaltiota, maata, laitosta, puhelinnumeroa, sähköpostiosoitteita, aktiivista sähköpostiosoitetta, erikoisaloja, vapaamuotoista kuvausta itsestään ja salasanaansa niitä vastaavilla kentillä. ”Update profile” -painiketta painamalla muutokset tulevat voimaan, ja asiantuntija siirtyy takaisin ”artikkelien arvostelu” -näkyeseen.

#### Virheenkäsittely

- Jos etunimi, sukunimi, sähköpostiosoite tai salasana on tyhjä, annetaan virheilmoitus ”You must specify first name / last name / e-mail address / password”.
- Jos lisättävä sähköpostiosoite on väärää muotoa (@-merkki puuttuu), annetaan virheilmoitus ”Must have @-symbol in e-mail address”.
- Jos lisättävä sähköpostiosoite löytyy jo tietokannasta, annetaan virheilmoitus ”E-mail address already registered”.
- Jos käyttäjän ainoa sähköpostiosoite yritetään poistaa, annetaan virheilmoitus ”You must have at least one e-mail address”.
- Jos syötetty salasana ei ole tarpeeksi pitkä, annetaan virheilmoitus ”Password must be at least 5 characters”.
- Jos syötetty salasana ei vastaa uudelleensyötettyä salasanaa, annetaan virheilmoitus ”Passwords don’t match”.

## 8.4 Toimitus

Toimittajan näkymän vasemman reunan valikossa on linkit näkyviin ”artikkelien hallinta”, ”lehden tila”, ”käyttäjähallinta”, ”raportit” ja ”profiilin muokkaus” sekä linkki ”uloskirjautuminen”. Päätoimittajalla on näiden lisäksi linkki näkyeseen ”päätoimittajanäkymä”.

### 8.4.1 Artikkelien hallinta

Näkymän yläreunassa on lueteltu kaikki järjestelmän artikkelit, jotka vaativat toimituksen toimia. Oletuksena luettelosta on valittu uusin saapunut artikkeli. Luettelossa näkyy artikkelien otsikko, kirjoittaja, ala, saapumispäivämäärä sekä tila asiantuntijoittain. Luettelon alapuolella olevan valintaruudun rastittamalla toimittaja saa artikkeliluetteloon kaikki järjestelmän artikkelit. Artikkelin otsikkoa artikkeliluettelosta painamalla toimittaja pääsee

käsittelyyn kyseistä artikkelia. Toimittaja voi tutustua artikkelin tekstilaatikossa näkyvään tiivistelmään ja koko artikkeliin ”Show article” painiketta painamalla, poistaa artikkelin järjestelmästä ”Remove article” -painikkeella sekä siirtää artikkelin ”lehden tila” -näkyvään ”Move to next magazine” -painikkeella.

Toimittaja voi valita uudelle artikkelille asiantuntijoita arvostelijoiksi. Artikkeliluettelon alapuolella on luettelo, johon on lueteltu kaikki järjestelmässä olevat asiantuntijat sekä heidän erikoisalansa ja sähköpostiosoitteensa. Tästä luettelosta toimittaja voi lisätä asiantuntijoita arvostelemaan valittua artikkelia ”Add referee to article”-painiketta haluttujen asiantuntijoiden kohdalla painamalla, jolloin valitut asiantuntijat siirtyvät kyseisen artikkelin arvostelijoiden luetteloon. Arvostelijoita voi poistaa kyseiseltä artikkelilta ”Remove referee from article”-painiketta painamalla. Toimittaja voi hakea toimittajia erikoisalan perusteella kirjoittamalla hakusanan ”Search referee by expertise” -kenttään ja painamalla ”Search”-painiketta. Toimittaja voi lisätä järjestelmään uuden asiantuntijan täyttämällä asiantuntijan etunimen, sukunimen, erikoisalat ja sähköpostiosoitteen niille varattuihin kenttiin ja painamalla ”Add referee” -painiketta. ”Send to referees” -painiketta painamalla artikkelin lausuntopyynnöt lähtevät valituille asiantuntijoille ja artikkelin tila päivittyy näkyvän artikkeliluetteloon. Toimittaja voi lähettää viestin päätoimittajalle valittua artikkelia koskien kirjoittamalla viestin sille varattuun kenttään ja painamalla ”Send message to editor in chief” -painiketta.

Lausuntokierrokselta valmistuneen artikkelin saamat palautteet ja niiden liitetiedostot on lueteltu asiantuntijoittain ryhmiteltynä. Toimittaja voi muokata palautteita, avata liitetiedostoja tiedostonimiä painamalla sekä valitsemalla valintaruuduilla mitkä liitetiedostot lähetetään kirjoittajalle. Toimittaja voi asiantuntijoiden palautteiden perusteella valita sopivan vaihtoehdoista ”Accepted”, ”Accepted with minor revision”, ”Accepted with major revision” tai ”Rejected”. ”Send information to writer” -painiketta painamalla artikkelin tila päivittyy näkyvän artikkeliluetteloon ja toimittaja saa kuittauksen lähetyksen onnistumisesta.

### **Virheen käsittely**

- Jos lisättävän asiantuntijan etunimi, sukunimi tai sähköpostiosoite on tyhjä, annetaan virheilmoitus ”You must specify referee’s first name / last name / e-mail address”.
- Jos lisättävän asiantuntija sähköpostiosoite on väärää muotoa (@-merkki puuttuu), annetaan virheilmoitus ”Must have @-symbol in e-mail address”.
- Jos lisättävän asiantuntijan sähköpostiosoite löytyy jo tietokannasta, annetaan virheilmoitus ”E-mail address already registered”.

#### **8.4.2 Lehden tila**

Näkymän yläreunassa on lueteltu järjestelmän artikkelit, joita ei ole vielä sijoitettu mihinkään lehteen, ja alareunassa lehden seuraavaan numeroon valitut artikkelit. Toimittaja

voi siirtyä lehden muihin numeroihin artikkeliluettelon alapuolella olevasta lehtiluettelosta halutun lehden numeroa painamalla. Artikkeleista näkyy luettelossa otsikko, saapumispäivämäärä, kirjoittaja sekä tila. Artikkelin otsikkoa artikkeliluettelossa painettuun toimittaja voi tutustua artikkelin tiivistelmään ”Show abstract” -painiketta painamalla ja koko artikkeliin ”Show article” -painikkeella tai lähettää artikkelin lausunnonantajille muistutuksen ”Send reminder” -painiketta painamalla halutun asiantuntijan nimen kohdalla. ”Send reminder” -painikkeiden oikealla puolella on kenttä, jossa näkyy viimeisin ajankohta, jolloin kyseiselle asiantuntijalle on lähetetty muistutus kyseisen artikkelin arvostelusta.

### 8.4.3 Käyttäjähallinta

Näkymän yläreunassa on lueteltu kaikki järjestelmän käyttäjät. Luettelossa näkyy käyttäjän nimi, lisäysoikeus, rooli (kirjoittaja/asiantuntija/toimittaja/päätoimittaja) ja sähköpostiosoite. Käyttäjän nimeä painamalla toimittaja pääsee muokkaamaan kyseisen käyttäjän tietoja käyttäjäluehduellon alapuolella olevalla lomakkeella. Lomakkeessa on kentät etunimelle, sukunimelle, tittelille, katuosoiteelle, kaupungille, postinumerolle, osavaltiolle, maalle, laitokselle, puhelinnumerolle, sähköpostiosoitteelle, roolille sekä erikoisaloille valitun käyttäjän roolin mukaan. ”Update fields” -painiketta painamalla kyseisen käyttäjän päivitettyt tiedot astuvat voimaan ja päivittyvät käyttäjäluehduellon. Toimittaja saa kuittauksen tietojen päivituksen onnistumisesta. Toimittaja voi lähettää käyttäjälle uuden järjestelmän generoiman salasanan painamalla ”Generate and send new password” -painiketta.

Toimittaja voi lisätä järjestelmään uusia toimittajia näkymän alareunan lomakkeella täyttämällä uuden toimittajan tiedot vastaaviin kenttiin ja painamalla ”Add editor” -painiketta.

Toimittaja voi muokata järjestelmän erikoisaloja näkymän alareunassa. Erikoisalat on lueteltu aakkosjärjestyksessä ja niitä voi poistaa sekä muokata vastaavilla painikkeilla. Uusia erikoisaloja voi lisätä ”Add new expertise” -painikkeella.

### Virheenkäsittely

- Jos muokattu etunimi tai sukunimi on tyhjä, annetaan virheilmoitus ”You must specify user’s first name / last name”.
- Jos lisättävä sähköpostiosoite on väärää muotoa (@-merkki puuttuu), annetaan virheilmoitus ”Must have @-symbol in e-mail address”.
- Jos lisättävä sähköpostiosoite löytyy jo tietokannasta, annetaan virheilmoitus ”E-mail address already registered”.
- Jos käyttäjän ainoa sähköpostiosoite yritetään poistaa, annetaan virheilmoitus ”User must have at least one e-mail address”.
- Jos yritetään antaa päätoimittaja-status useammalle kuin yhdelle toimittajalle, annetaan virheilmoitus ”Editor in chief already defined: name”.

- Jos lisättävän toimittajan etunimi, sukunimi tai sähköpostiosoite on tyhjä, annetaan virheilmoitus ”You must specify editor’s first name / last name / e-mail address”.
- Jos lisättävän toimittajan sähköpostiosoite löytyy jo tietokannasta, annetaan virheilmoitus ”E-mail address already registered”.

#### **8.4.4 Raportit**

Näkymässä näkyy järjestelmän tilastotietoja, joihin liittyviä attribuutteja on esitelty luvussa 9.

#### **8.4.5 Profilin muokkaus**

Toimittaja voi muokata seuraavia tietojaan: etunimeä, sukunimeä, titteliä, katuosoitetta, kaupunkia, postinumeroa, osavaltiota, maata, puhelinnumeroa, sähköpostiosoitetta ja salasanaanasi niitä vastaavilla kentillä. ”Update profile” -painiketta painamalla muutokset tulevat voimaan, ja toimittaja siirtyy takaisin näkymään, jossa oli ennen profilin muokkausta.

#### **Virheenkäsittely**

- Jos etunimi, sukunimi, sähköpostiosoite tai salasana on tyhjä, annetaan virheilmoitus ”You must specify first name / last name / e-mail address / password”.
- Jos muokattu sähköpostiosoite on väärää muotoa (@-merkki puuttuu), annetaan virheilmoitus ”Must have @-symbol in e-mail address”.
- Jos muokattu sähköpostiosoite löytyy jo tietokannasta, annetaan virheilmoitus ”E-mail address already registered”.
- Jos syötetty salasana ei ole tarpeeksi pitkä, annetaan virheilmoitus ”Password must be at least 5 characters”.
- Jos syötetty salasana ei vastaa uudelleensyötettyä salasanaa, annetaan virheilmoitus ”Passwords don’t match”.

#### **8.4.6 Päätoimittajanäkymä**

Päätoimittajanäkymän yläreunassa on lueteltu artikkelit, joita koskien muu toimitus on lähettänyt viestejä päätoimittajalle. Luettelon alapuolella olevan valintaruudun rastittamalla päätoimittaja saa artikkeliluetteloon kaikki järjestelmän artikkelit. Artikkelin otsikkoa artikkeliluettelosta painamalla päätoimittaja näkee mahdollisen toimitukselta tulleen viestin artikkeleita koskien sekä käsittelemään artikkeleita samanlaisen näkymän kautta kuin muutkin toimittajat.

Päätoimittajanäkymän vasemman reunan valikko on identtinen tavallisen toimittajan valikon kanssa ”päätoimittajanäkymä”-linkkiä lukuunottamatta, joten päätoimittajalle tarjotaan samat toiminnallisuudet kuin muillekin toimittajille.

## 9 Järjestelmästä saatavat raportit

Lehden toimituksella on mahdollisuus tulostaa erilaisia raportteja artikkeleihin ja asiantuntijoihin liittyen. Seuraavassa luetellaan raporttien kannalta oleelliset attribuutit ja niiden merkitykset sekä tieto jonka mukaan kyseinen attribuutti voidaan järjestää. Asiantuntijoiden tietoja voidaan tulostaa seuraavasti:

Attribuutti	Merkitys	Järjestäminen
Sukunimi	Asiantuntijan sukunimi	Aakkosjärjestys
Etunimi	Asiantuntijan etunimi	(Aakkosjärjestys)
Ala	Asiantuntijan erikoistumisalat	Aakkosjärjestys
Arvioitavaa	Asiantuntijalla on yhden tai useamman artikkelin arviointi kesken	kesken/ei kesken
Kieltäytymisiä	Kuinka monta kieltäytymistä artikkelin tarkastuksesta/tarkastuspyynnöt	Kieltäytymisten lkm
Lausunnon kesto	Kuinka kauan lausunnon antaminen on keskimäärin kestänyt	Keston mukaan
Arviot	Kuinka monta arviota kussakin ryhmässä: hyväksytyt / hyväksytyt pienin muutoksin / hyväksytyt suurin muutoksin / hylätyt /	Ei voida järjestää

Lisäksi seuraavista attribuuteista on ”yhteensä”-kenttä: arvioitavaa, kieltäytymisiä, lausunnon kesto ja arviot. Artikkelien tietoja voidaan rajata tietyille aikavälille ja tulostaa seuraavasti:

Attribuutti	Merkitys	Järjestäminen
Nimi	Artikkelin nimi	Aakkosjärjestys
Tila	Artikkelin tila: hylätty, vastaanotettu, 1. kierros, 2. kierros, ..., hyväksytyt muutoksin, hyväksytyt, julkaistu. Jokaisessa päivämäärä.	Tilan ja tilan sisällä ajan mukaisessa järjestyksessä

Lisäksi seuraavista tiloista on ”yhteensä”-kenttä: hylätty, vastaanotettu, 1.kierros, 2.kierros, ..., hyväksytyt, julkaistu.

## 10 Erityiset tekniset ratkaisut

### 10.1 Salasanan tallennus

Kaikki salasanat ovat tietokannassa Salaaja-luokan avulla salatussa muodossa. Turvallisuussyistä salasanoja käsitellään selväkielisenä vain, kun salasana lähetetään kirjautumissivulta järjestelmään. Kirjautumissivulta saatu salasana salataan Salaaja-luokan avulla ja verrataan salattua versiota tunnusta vastaavaan salattuun arvoon tietokannassa. Jos salasana oli oikein, luodaan käyttäjää varten istuntotunnus (sessionID), joka tallennetaan istuntomuuttujiin. Käyttöoikeuksia tarkistettaessa verrataan tunnus+istuntotunnus-yhdistelmää vastaavaan tauluun tietokannassa.

### 10.2 Sähköpostin lähetys

Sähköpostin lähetykseen käytämme Javan standardilaajennusta JavaMail API 1.3.1.<sup>1</sup> JavaMail-kirjasto käyttää hyväkseen JavaBeans Activation Framework -kirjastoa. Molemmat löytyvät Sunin sivuilta<sup>2</sup> ja tulevat myös JBuilder 9:n mukana. JBuilderissa nämä kaksi kirjastoa saadaan käyttöön projektin luonnin yhteydessä valitsemalla ne tarvittaviksi kirjastoiksi (Required Libraries). Sunin sivuilta löytyy myös artikkeli komentokirjaston (Tag Library) käytöstä JavaMailin kanssa<sup>3</sup>.

### 10.3 Tiedostojen käsittely

Binääritiedostot, joille halutaan rajattu näkyvyys, säilytetään aina hakemistossa, johon on oikeus vain njc2-tunnuksen haltijalla. Näin varmistetaan, etteivät ulkopuoliset pääse lukemaan tiedostoja esimerkiksi URL-osoitetta arvaamalla.

Tiedostot näytetään käyttäjälle TiedostoServlet-luokan avulla. Näin voidaan pitää huoli siitä, että käyttäjällä on oikeus lukea kyseinen tiedosto, ja että näyttökertoja voidaan tarvittaessa rajata.

Myös järjestelmään lähetetyt tiedostot tallennetaan hakemistoon, johon ei ole suoraa pääsyä Internetistä.

---

<sup>1</sup><http://java.sun.com/products/javamail/javadocs/index.html>

<sup>2</sup><http://java.sun.com/products/javamail/>

<sup>3</sup><http://java.sun.com/developer/technicalArticles/javaserverpages/emailapps/>

## **Liite 1. Luokkakaavio**