

# **Suunnitteludokumentti**

AssariXP

Helsinki 13.11.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (6 ov)

**Projektiryhmä**

Mikko Hakila  
Maruan Khoury  
Ilkka Manner  
Pirjo Tervonen  
Kirsi Yläne  
Tuija Åkerblom

**Asiakas**

Harri Laine

**Johtoryhmä**

Sini Ruohomaa (ohjaaja)  
Juha Taina  
Turjo Tuohiniemi

**Kotisivu**

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

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
1.0	13.11.2004	Jäädetytty dokumentti

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Dokumentin tarkoitus . . . . .	1
1.2	Dokumentin rakenne . . . . .	1
<b>2</b>	<b>Järjestelmän kokonaiskuvaus</b>	<b>1</b>
<b>3</b>	<b>Tietokannan rakenne</b>	<b>3</b>
3.1	Tietokantaan tehtävät muutokset . . . . .	4
3.1.1	Ryhmän tiedot . . . . .	4
3.1.2	Pakkauksen näkyvyyden tiedot . . . . .	4
3.1.3	Pakkauksen tiedot . . . . .	5
3.1.4	Kurssin tiedot . . . . .	5
3.1.5	Palautteen tiedot . . . . .	6
<b>4</b>	<b>MainServlet-komponentti</b>	<b>6</b>
4.1	MainServlet-luokan kuvaus . . . . .	6
4.1.1	MainServlet-luokan metodien kuvaus . . . . .	7
<b>5</b>	<b>Tietokantakyselyiden suorituskomponentti</b>	<b>9</b>
5.1	DatabaseAccess-luokan kuvaus . . . . .	9
5.1.1	DatabaseAccess-luokan metodien kuvaus . . . . .	10
<b>6</b>	<b>Navigointi- ja käyttäjän syötteen validointikomponentti</b>	<b>11</b>
6.1	NavBar-luokan kuvaus . . . . .	11
6.1.1	NavBar-luokan metodien kuvaus . . . . .	12
6.2	HTMLFilter-luokan kuvaus . . . . .	12
6.2.1	HTMLFilter-luokan metodin kuvaus . . . . .	12
<b>7</b>	<b>Ryhmäkomponentti</b>	<b>13</b>
7.1	GroupHTML-luokan kuvaus . . . . .	14
7.1.1	GroupHTML-luokan metodien kuvaus . . . . .	14
7.2	Group-luokan kuvaus . . . . .	16
7.2.1	Group-luokan metodien kuvaus . . . . .	16

<b>8 Pakkauskomponentti</b>	<b>17</b>
8.1 ModuleHTML-luokan kuvaus . . . . .	18
8.1.1 ModuleHTML-luokan metodien kuvaus . . . . .	18
8.2 Module-luokan kuvaus . . . . .	19
8.2.1 Module-luokan metodien kuvaus . . . . .	20
<b>9 Palautelomakekomponentti</b>	<b>22</b>
9.1 FeedbackFormServlet-luokan kuvaus . . . . .	23
9.1.1 FeedbackFormServlet-luokan metodien kuvaus . . . . .	23
<b>10 Palauteyhteenvetokomponentti</b>	<b>25</b>
10.1 FeedbackSummary-luokan kuvaus . . . . .	26
10.1.1 FeedbackSummary-luokan metodien kuvaukset . . . . .	27
<b>11 QTI-muunnoskomponentti</b>	<b>28</b>
11.1 QtiHTML-luokan kuvaus . . . . .	28
11.1.1 QtiHTML-luokan metodien kuvaus . . . . .	29
11.2 Qti-luokan kuvaus . . . . .	30
11.2.1 Qti-luokan metodien kuvaus . . . . .	30
<b>Lähteet</b>	<b>32</b>
<b>Liitteet</b>	
<b>1 Selvitys QTI-muunnoksesta</b>	
<b>2 Testaussuunnitelma</b>	
<b>A Johdanto</b>	<b>1</b>
<b>B Testauksessa käytettävät välineet</b>	<b>1</b>
<b>C Testausstrategia</b>	<b>1</b>
C.1 Yleistä . . . . .	1
C.2 Yksikkötestaus . . . . .	3
C.3 Integraatiotestaus . . . . .	3
C.4 Järjestelmättestaus . . . . .	3

C.5	Hyväksymistestaus . . . . .	3
<b>D</b>	<b>Testauksen raportointi</b>	<b>3</b>
<b>E</b>	<b>Suoritettavat testitapaukset</b>	<b>4</b>
E.1	MainServlet-komponentti . . . . .	4
E.2	Navigointikomponentti . . . . .	4
E.2.1	Navigointipalkki . . . . .	4
E.3	Käyttäjän syötteen validointikomponentti . . . . .	5
E.4	Tietokantakyselyiden suorituskomponentti . . . . .	5
E.5	Ryhmäkomponentti . . . . .	5
E.5.1	Uuden ryhmän luominen . . . . .	5
E.5.2	Käyttäjän liittäminen ryhmään . . . . .	5
E.5.3	Käyttäjän poistaminen ryhmästä . . . . .	6
E.6	Pakkauksikomponentti . . . . .	6
E.6.1	Uuden pakkauksen luominen . . . . .	6
E.6.2	Tehtävän lisääminen pakkaukseen . . . . .	7
E.6.3	Tehtävän poistaminen pakkauksesta . . . . .	7
E.6.4	Tehtävien järjestäminen pakkauksessa . . . . .	7
E.6.5	Pakkauksen käyttöajan määrittäminen . . . . .	8
E.6.6	Pakkauksen näkyvyyden määrittäminen . . . . .	9
E.6.7	Pakkauksen tyypin määrittäminen . . . . .	9
E.6.8	Palautemoodin määrittäminen . . . . .	9
E.7	Palautelomakekomponentti . . . . .	10
E.7.1	Palautteen antaminen . . . . .	10
E.8	Palauteyhteenvetokomponentti . . . . .	10
E.9	QTI-muunnoskomponentti . . . . .	11
E.9.1	Valittujen tehtävien muuntaminen QTI-muotoon . . . . .	11

### 3 Käyttöliittymäkuvia

#### 4 Palautelomake

# 1 Johdanto

AssariXP on Helsingin yliopiston tietojenkäsittelytieteen laitoksen Ohjelmistotuotanto-projekti-kurssin projekti. Projektissa on tarkoitus laajentaa projektin asiakkaan Harri Lai-  
neen kehittämää eAssari-opetusohjelmistoa [Lai03] uusilla ominaisuuksilla.

## 1.1 Dokumentin tarkoitus

Tämän dokumentin tarkoituksena on kuvata projektissa toteutettava järjestelmä niin tarkasti, että järjestelmän toteuttaminen on suoraviivaista. Järjestelmän vaatimukset on esitetty Vaatimusmäärittely-dokumentissa, ja tässä dokumentissa esitetään, miten vaatimukset toteutetaan teknisesti.

## 1.2 Dokumentin rakenne

Luvussa 2 esitetään järjestelmän kokonaiskuvaus ja luvussa 3 kuvataan tietokannan rakenne ja nykyiseen tietokantaan projektissa tehtävät muutokset. Luvussa 4 kuvataan MainServlet-komponentti, jonka avulla luodaan käyttäjille näytettävät HTML-sivut. Luvussa 5 esitellään tietokantakyselyiden suorituskomponentti, jonka kautta tehdään kaikki tietokantakyselyt. Luvussa 6 esitellään navigointi- ja käyttäjän syötteen validointikomponentit, joita käytetään apukomponentteina. Luvussa 7 kuvataan ryhmäkomponentti, luvussa 8 pakkauskomponentti, luvussa 9 palautelomakekomponentti ja luvussa 10 palautteiden yhteenvetokomponentti. Luvussa 11 esitellään QTI-muunnoskomponentti.

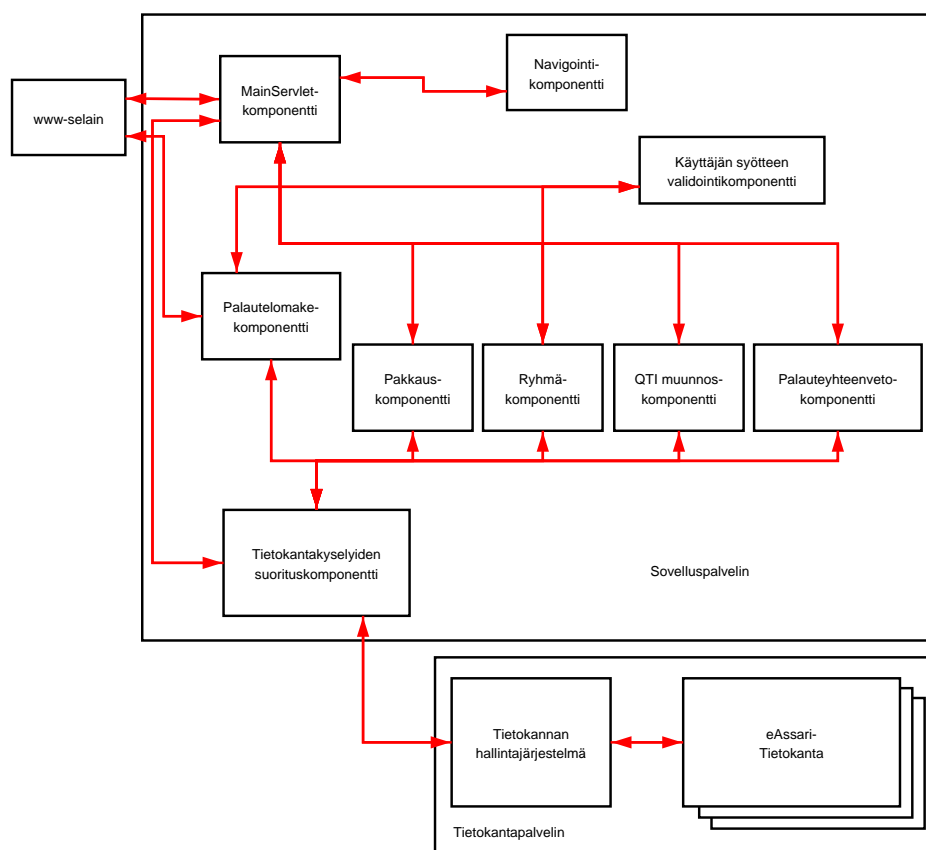
Liite 1 sisältää QTI-selvityksen, liite 2 testaussuunnitelman, liite 3 käyttöliittymäkuvia ja liite 4 palautelomakkeen kuvan.

# 2 Järjestelmän kokonaiskuvaus

Tämä luku kertoo järjestelmän eri komponenttien suhteesta toisiinsa ja olemassa olevaan järjestelmään. Luvussa kuvataan järjestelmän arkkitehtuuri eli sen komponenttien toiminta yleisellä tasolla. Kuvassa 1 sivulla 2 on esitetty järjestelmän kokonaiskuva.

Sovelluspalvelimessa ovat seuraavat osakomponentit, joita muut komponentit käyttävät yhteisesti:

**MainServlet-komponentti** huolehtii käyttäjän syötteiden vastaanotosta ja niiden välittämisestä edelleen. Komponentti validoi saamansa syötteet käyttäjän syötteen validointikomponentin avulla. Komponentti huolehtii myös käyttöoikeuksien hallinnasta. Paluuviestit eri komponenteilta käyttäjälle kulkevat tämän komponentin kautta. Paluuviestiin, joka on HTML-sivu, liitetään navigointikomponentilta pyydetty navigointipalkin HTML-koodi.



Kuva 1: Järjestelmän kokonaiskuva

**Navigointikomponentti** tarjoaa kaikkiin käyttäjälle näytettäviin sivuihin navigointipalkin.

**Käyttäjän syötteen validointikomponentti** validoi käyttäjältä saadut syötteet, jotka kierretään komponentin kautta ennen kuin syötteitä käytetään muissa komponenteissa. Komponentti muuttaa tietyt merkit sellaisiksi, että ne näkyvät HTML-sivulla oikein. Tällaisia merkkejä ovat esimerkiksi '<' ja '>'.

**Tietokantakyselyiden suorituskomponentti** vähentää virheitä järjestelmässä, kun kaikki tietokantaan tehtävät haut, poistot ja lisäykset tehdään tämän komponentin kautta. Muut komponentit lähettävät komponentille SQL-standardin mukaisia lauseita, joiden suorittamisesta järjestelmän tietokannassa tämä komponentti huolehtii.

Seuraavat komponentit suorittavat jonkun tietyn toimintokokonaisuuden:

**Ryhmäkomponentti** tarjoaa ryhmälle asetettujen vaatimusten toteuttamisen eli ryhmän perustamisen ja ryhmään liittymisen. Komponentti tuottaa myös käyttäjälle näytettävän HTML-sivun pohjan.

**Pakkauskomponentti** tarjoaa pakkausten hallinnointiin liittyvät toiminnot. Komponentti tuottaa myös käyttäjälle näytettävän HTML-sivun pohjan.

**Palautelomakekomponentti** luo käyttäjälle näytettävän palautelomakkeen ja käsittelee palautelomakkeen tiedot. Komponentti tuottaa myös käyttäjälle näytettävän HTML-sivun pohjan.

**Palauteyhteenvetokomponentti** tuottaa yhteenvetotietoja annetusta palautteesta. Komponentti tuottaa myös käyttäjälle näytettävän HTML-sivun pohjan.

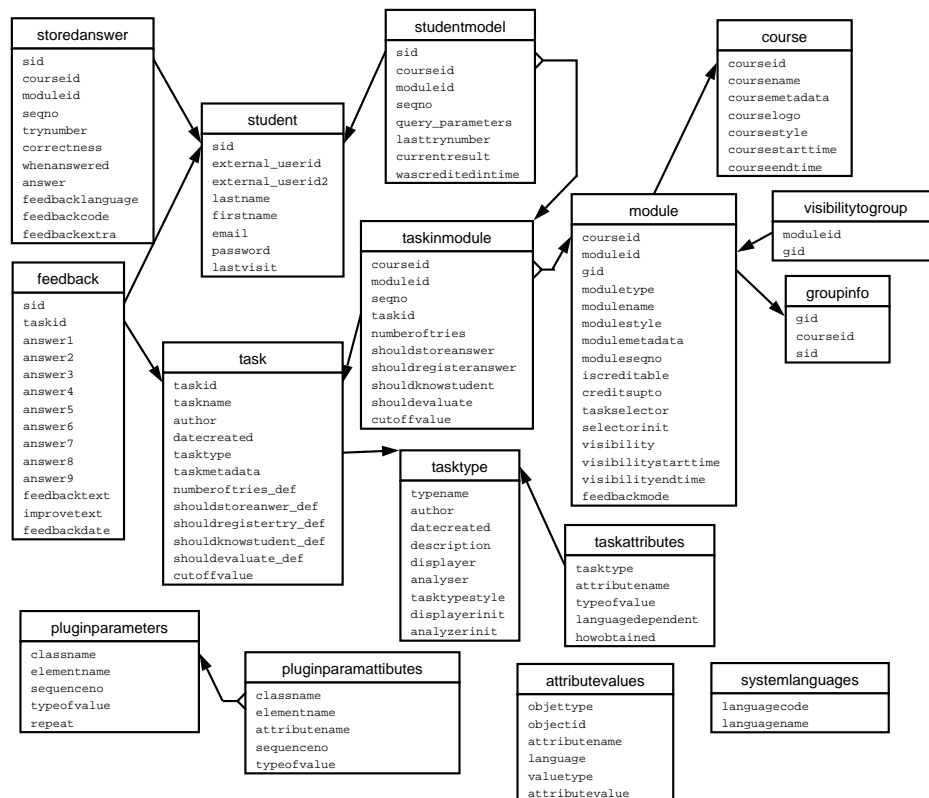
**QTI-muunnoskomponentti** suorittaa QTI-muunnoksen. Komponentti tuottaa myös käyttäjälle näytettävän HTML-sivun pohjan.

Näiden komponenttien lisäksi järjestelmässä käytetään tietokannan hallintajärjestelmää, joka tekee varsinaiset kyselyt tietokannan tauluihin.

Kaikki projektissa toteutettavat luokat kuuluvat pakkaukseen fi.hy.eassari.xp.

### 3 Tietokannan rakenne

Kuvassa 2 on kuvattu järjestelmän tietokanta. Kuvassa näkyvät tietokannassa jo olevat taulut sekä siihen tehtävät uudet taulut ja sarakkeet.



Kuva 2: Tietokantakaavio



### 3.1 Tietokantaan tehtävät muutokset

Alla on kuvattu tietokantaan lisättävät uudet taulut ja jo olemassa oleviin tauluihin lisättävät uudet sarakkeet. Uusia tauluja ovat `groupinfo`, jossa säilytetään ryhmien tiedot, ja `visibilitytogroup`, jossa ovat tiedot pakkauksen näkyvyyttä varten. Vanhoihin tauluihin `module` ja `course` lisätään uusia sarakkeita. Uusi taulu on myös `feedback`, johon tallennetaan palautelomakkeella annettavat palautteet.

Alla olevissa taulukoissa kohta Sarake kertoo taulun sarakkeiden arvot ja tyypit, ja kohta Kuvaus kertoo, millaista tietoa sarake sisältää ja miten sarakkeen arvo tuotetaan kyseiseen tauluun.

Ryhmän tunniste on käyttäjän ryhmälle antama yksiselitteinen nimi.

#### 3.1.1 Ryhmän tiedot

Taulu <code>groupinfo</code>	
Sarake	Kuvaus
<code>gid: varchar(20)</code>	Ryhmän tunniste. Arvon tuottaa käyttäjä.
<code>courseid: varchar(20)</code>	Kurssin tunniste. Arvon tuottaa järjestelmä.
<code>sid: varchar(20)</code>	Opiskelijan tunniste. Arvon tuottaa järjestelmä.

#### 3.1.2 Pakkauksen näkyvyyden tiedot

Taulu <code>visibilitytogroup</code>	
Sarake	Kuvaus
<code>moduleid: varchar(20)</code>	Pakkauksen tunniste. Arvon tuottaa järjestelmä.
<code>gid: varchar(20)</code>	Ryhmän tunniste. Arvon tuottaa käyttäjä.

### 3.1.3 Pakkauksen tiedot

<b>Taulu</b> module	
<b>Sarake</b>	<b>Kuvaus</b>
gid: varchar(20)	Ryhmän tunniste. Arvon tuottaa käyttäjä.
visibility: integer	Pakkauksen näkyvyys: arvo on 0, jos pakkaus on suunnittelutilassa, ja arvo on 1, jos pakkaus on julkistetussa tilassa. Arvon tuottaa pakkauksen omistaja.
visibilitystarttime: date	Pakkauksen julkistamisaika. Arvon tuottaa pakkauksen omistaja.
visibilityendtime: date	Pakkauksen julkistetun tilan loppumisaika. Arvon tuottaa pakkauksen omistaja.
feedbackmode: integer	Pakkauksen tila: arvo on 1, jos palaute-moodi on päällä, ja arvo on 0, jos palaute-moodi ei ole päällä. Arvon tuottaa opettaja.

### 3.1.4 Kurssin tiedot

<b>Taulu</b> course	
coursestarttime: date	Kurssin alkamisaika. Arvon tuottaa kurssin opettaja.
courseendtime: date	Kurssin loppumisaika. Arvon tuottaa kurssin opettaja.

### 3.1.5 Palautteen tiedot

Palautelomakkeen kuva on liitteessä 4. Kuvassa näkyvät palautelomakkeen kysymykset 1-9.

<b>Taulu</b> feedback	
<b>Sarake</b>	<b>Kuvaus</b>
sid: varchar(20)	Opiskelijan tunniste. Arvon tuottaa järjestelmä.
taskid: varchar(20)	Tehtävän tunniste. Arvon tuottaa järjestelmä.
answer1: integer	Palautelomakkeen kysymyksen 1 palaute. Arvon tuottaa palautteen antaja.
answer2: integer	Palautelomakkeen kysymyksen 2 palaute. Arvon tuottaa palautteen antaja.
answer3: integer	Palautelomakkeen kysymyksen 3 palaute. Arvon tuottaa palautteen antaja.
answer4: integer	Palautelomakkeen kysymyksen 4 palaute. Arvon tuottaa palautteen antaja.
answer5: integer	Palautelomakkeen kysymyksen 5 palaute. Arvon tuottaa palautteen antaja.
answer6: integer	Palautelomakkeen kysymyksen 6 palaute. Arvon tuottaa palautteen antaja.
answer7: integer	Palautelomakkeen kysymyksen 7 palaute. Arvon tuottaa palautteen antaja.
answer8: integer	Palautelomakkeen kysymyksen 8 palaute. Arvon tuottaa palautteen antaja.
answer9: integer	Palautelomakkeen kysymyksen 9 palaute. Arvon tuottaa palautteen antaja.
feedbacktext: varchar(2000)	Tehtävän sanallinen palaute. Arvon tuottaa palautteen antaja.
improvetext: varchar(2000)	Tehtävän parannusehdotuksia -palaute. Arvon tuottaa palautteen antaja.
feedbackdate: date	Palautteen antamisen päivämäärä. Arvon tuottaa järjestelmä.

## 4 MainServlet-komponentti

MainServlet-komponentti koostuu MainServlet-luokasta.

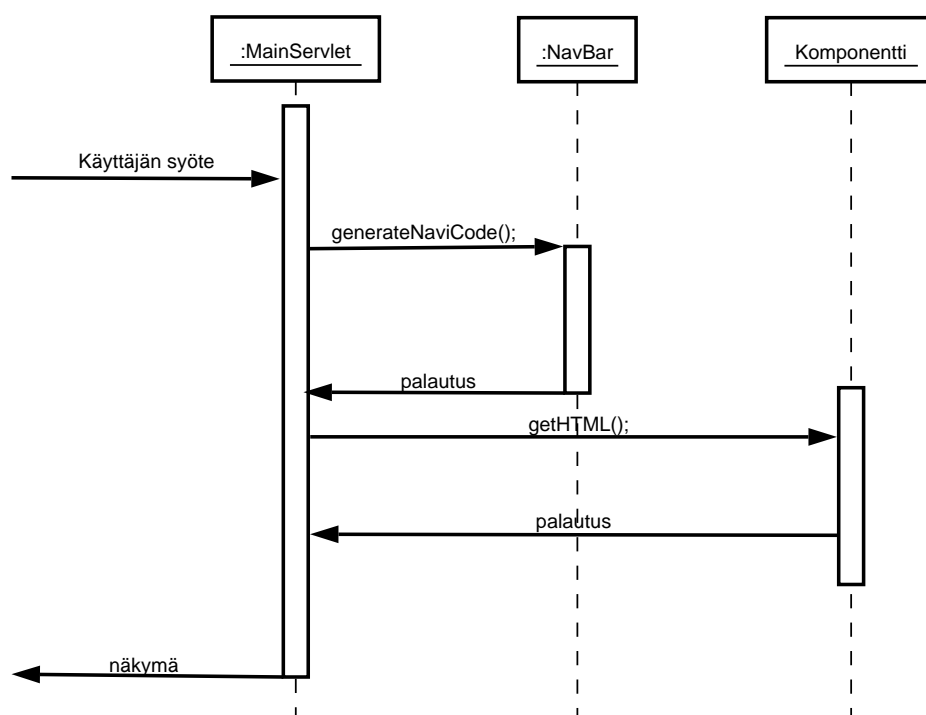
### 4.1 MainServlet-luokan kuvaus

MainServlet-luokka toimii järjestelmän koordinaattorina. Käyttäjän pyytämä näkymä tuotetaan MainServlet-luokan service-metodilla. Metodi kutsuu ensiksi NavBar-luokkaa, jol-

ta se saa näkymään liittyvän navigointipalkin HTML-koodin. Sen jälkeen kutsutaan käyttäjän pyytämän näkymän tuottavaa komponenttia, jolle MainServlet ohjaa käyttäjän syötteen. Kun jokin komponentti on käsitellyt syötteen, se palauttaa MainServletille näkymän HTML-koodina, jonka MainServlet siirtää selaimelle. Kuvassa 3 on esitetty MainServletin sekvenssikaavio ja kuvassa 4 sivulla 8 on MainServletin luokkakaavio.

MainServlet-luokka tarjoaa seuraavat palvelut:

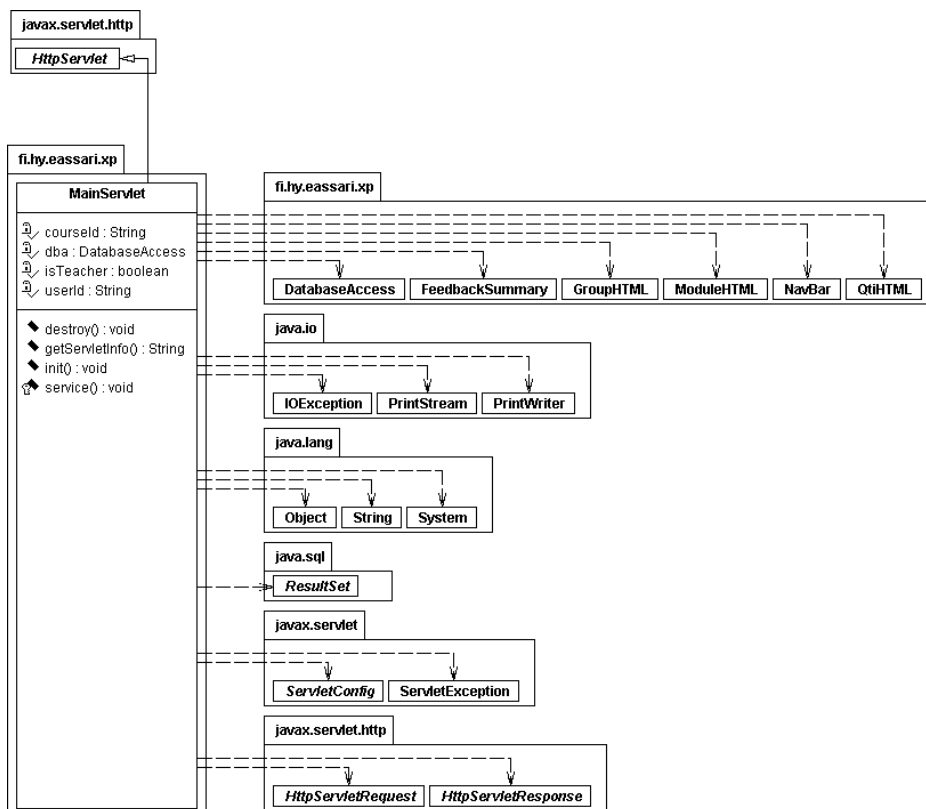
- Käyttäjän syötteen ohjaaminen oikealle komponentille.
- Käyttäjän session lopettaminen.
- Näkymien koonti HTML-sivuksi.



Kuva 3: MainServletin sekvenssikaavio

#### 4.1.1 MainServlet-luokan metodien kuvaus

Tässä luvussa kuvataan MainServlet-luokan metodit parametreineen ja paluuarvoineen.



Kuva 4: MainServlet-luokan luokkakaavio

<b>public void init(ServletConfig config)</b>		
Metodin kuvaus	Servletin alustusmetodi	
Parametrit	config	Servletin alustusolio
Heittää virheet	ServletException	Yleinen servletin virhe: heitetään, kun kohdataan vaikeuksia.

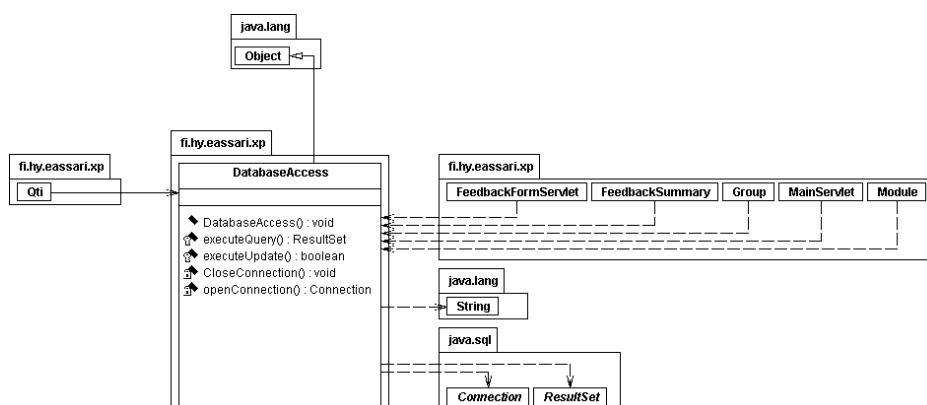
<b>public void destroy()</b>	
Metodin kuvaus	Tuhoaa servletin

<b>protected void service(HttpServletRequest request, HttpServletResponse response)</b>		
Metodin kuvaus	Käsittelee doGet- ja doPost-metodit	
Parametrit	request  response	Tiedot selaimelta saadusta pyynnöstä Kenttä, jolla voidaan tuottaa sisältöä vastaukseen
Heittää virheet	ServletException  IOException	Yleinen servletin virhe: heitetään, kun kohdataan vaikeuksia. Heitetään, kun kohdataan jonkinlainen I/O-virhe.

<b>public String getServletInfo()</b>	
Metodin kuvaus	Palauttaa lyhyen kuvauksen servletistä
Paluuarvo	Lyhyt kuvaus servletistä

## 5 Tietokantakyselyiden suorituskomponentti

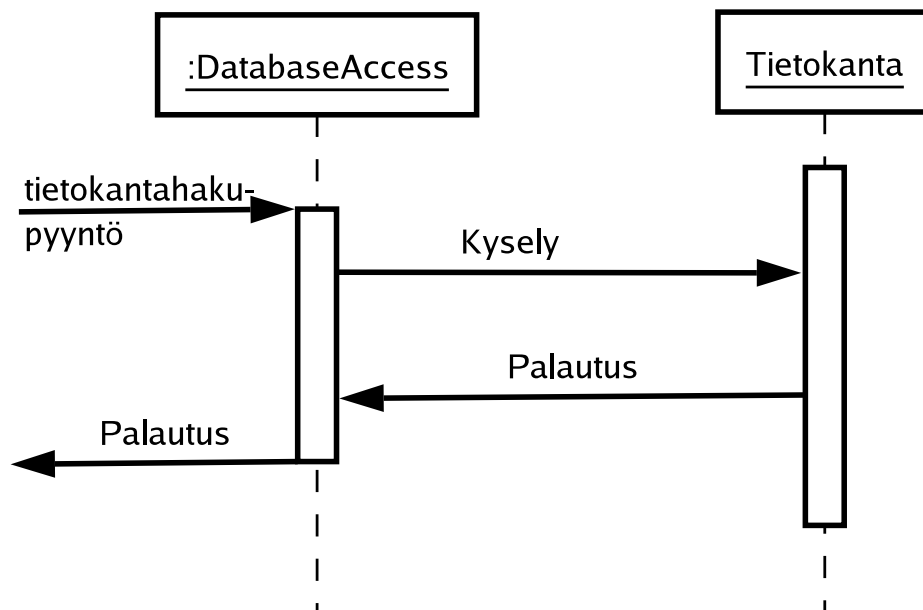
Tietokantakyselyiden suorituskomponentti koostuu DatabaseAccess-luokasta. Kuvassa 5 on esitetty luokan luokkakaavio.



Kuva 5: DatabaseAccess-luokan luokkakaavio

### 5.1 DatabaseAccess-luokan kuvaus

DatabaseAccess-luokkaa käytetään kaikkiin tietokantahakuihin ja päivityksiin. Muut komponentit kutsuvat DatabaseAccessin metodeita ja antavat parametreina SQL-lauseet, joiden avulla ne haluavat hakea tietokannasta tietoja tai päivittää sinne tietoja. DatabaseAccess suorittaa kyselyt suoraan näitä lauseita käyttäen. Mahdollisia kyselyiden tarkistuksia ei



Kuva 6: DatabaseAccess-luokan sekvenssikaavio

suoriteta tässä luokassa, vaan tarkistukset tehdään ennen kuin DatabaseAccessia kutsutaan. Kuvassa 6 on esitetty tietokantakyselyiden suorituskomponentin sekvenssikaavio.

### 5.1.1 DatabaseAccess-luokan metodien kuvaus

DatabaseAccess-luokassa on konstruktori ja kaksi metodia, joita voidaan kutsua luokan ulkopuolelta:

<b>public DatabaseAccess( )</b>	
Metodin kuvaus	Luokan konstruktori

<b>protected ResultSet executeQuery(String SQLQuery)</b>	
Metodin kuvaus	Tätä metodia kutsutaan aina, kun halutaan hakea jotain tietoa tietokannasta. Metodi hakee tietokannasta ne tiedot, joita parametrina annetulla String-muotoisella SQL-lauseella pyydetään.
Parametrit	SQLQuery      Haluttu tietokantakysely
Paluuarvo	Palauttaa ResultSet-oliossa haun vastauksen

<b>protected int executeUpdate(String SQLQuery)</b>	
Metodin kuvaus	Tätä metodia kutsutaan aina, kun halutaan muuttaa tietokannassa olevia tietoja. Metodi suorittaa kaikki Insert-, Delete- ja Update-lauseet, joita muut komponentit haluavat suorittaa.
Parametrit	SQLQuery Haluttu tietokantapäivitys
Paluuarvo	Palauttaa kutsujalleen joko arvon 1 (jos muutos onnistui) tai arvon 0 (jos muutos ei onnistunut).

Alla olevat metodit ovat vain luokan omaa käyttöä varten. Niitä ei voida kutsua luokan ulkopuolelta.

<b>private Connection openConnection()</b>	
Metodin kuvaus	Metodilla yritetään avata tietokantayhteys. Metodi hakee tarvittavat tiedot yhteyden avaamiseen tiedostosta.
Paluuarvo	Metodi palauttaa takaisin Connection-olion.

<b>private void closeConnection(Connection conn)</b>	
Metodin kuvaus	Metodilla suljetaan tietokantayhteys aina, kun tarvittavat haut tai päivitykset on suoritettu.
Parametrit	conn Yhteysolio, joka suljetaan
Paluuarvo	Metodi ei palauta mitään.

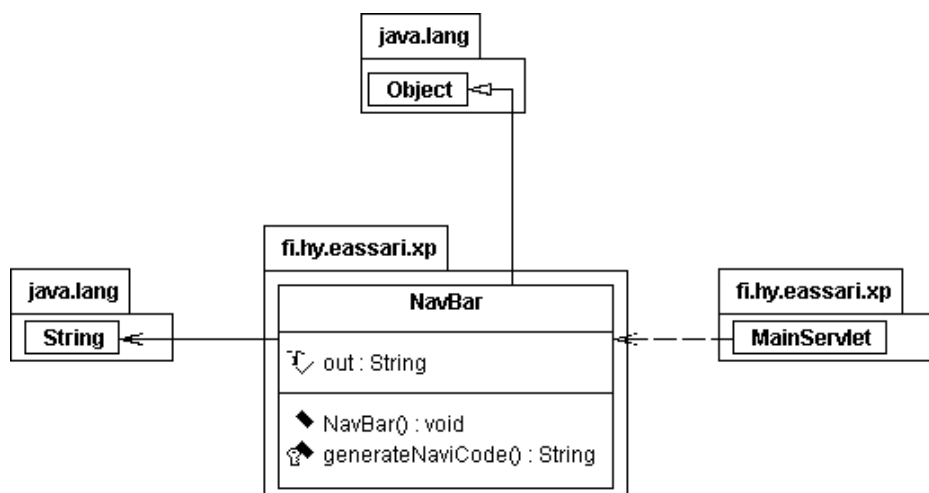
## 6 Navigointi- ja käyttäjän syötteen validointikomponentti

Tässä luvussa esitellään järjestelmän käyttämiä apukomponentteja. Apukomponentit ovat muiden komponenttien käytettävissä. Ne eivät itse käytä muita luokkia palveluidensa tuottamiseen. Navigointi-komponentti koostuu NavBar-luokasta ja käyttäjän syötteen validointikomponentti koostuu HTMLFilter-luokasta. Alaluvuissa ovat näiden luokkien kuvaukset.

### 6.1 NavBar-luokan kuvaus

NavBar-luokka generoi navigointipalkin HTML-koodin sitä pyydettyäessä. Navigointipalkki voi olla erilainen riippuen käyttäjän statuksesta. Luokan luokkakaavio on esitetty kuvassa 7 sivulla 12.





Kuva 7: NavBar-luokan luokkakaavio

### 6.1.1 NavBar-luokan metodien kuvaus

Tässä luvussa kuvataan NavBar-luokan metodit parametreineen ja paluuarvoineen.

<b>public NavBar()</b>	
Metodin kuvaus	Luokan konstruktori

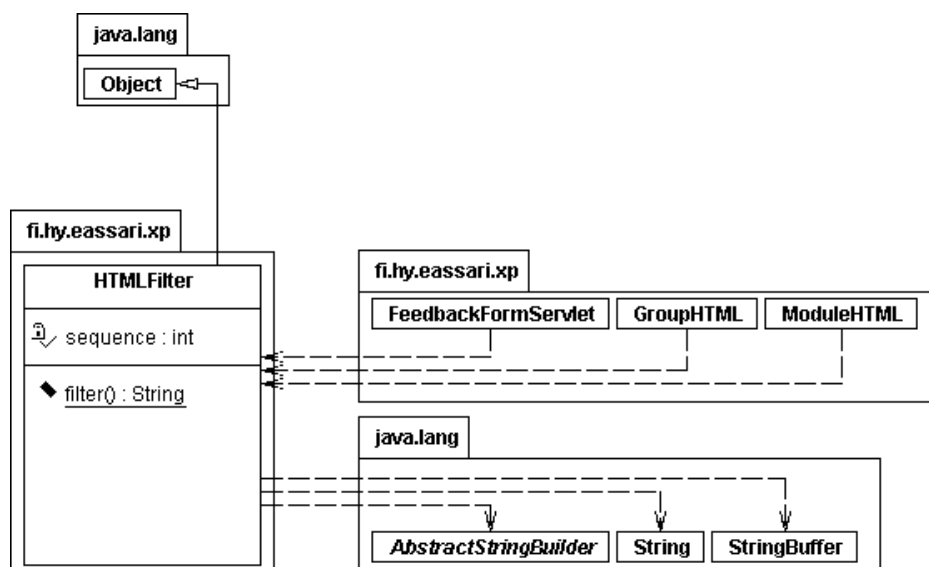
<b>protected String generateNaviCode(boolean isTeacher)</b>		
Metodin kuvaus	Metodi, jonka avulla MainServlet-luokka näyttää navigointipalkin	
Parametrit	isTeacher	Käyttäjän status
Paluuarvo	Navigointipalkin HTML-koodi	

## 6.2 HTMLFilter-luokan kuvaus

Käyttäjän syötteen validointikomponentti koostuu HTMLFilter-luokasta. HTMLFilter-luokkaa käyttävät komponentit, jotka saavat käyttäjän muokkaamia syötteitä. HTMLFilter-luokka tarkistaa käyttäjän antamat syötteet ja poistaa niistä merkit, joita käyttämällä järjestelmää vastaan voitaisiin tehdä esimerkiksi JavaScript-hyökkäys. Kuvassa 8 sivulla 13 on esitetty luokan luokkakaavio.

### 6.2.1 HTMLFilter-luokan metodin kuvaus

Tässä luvussa kuvataan HTMLFilter-luokan metodi parametreineen ja paluuarvoineen.



Kuva 8: HTMLFilter-luokan luokkakaavio

<b>public static String filter(String message)</b>	
Metodin kuvaus	Käyttäjän syötteet tarkastava metodi
Parametrit	message      Tarkistettava syöte
Paluuarvo	Tarkastettu ja muokattu syöte

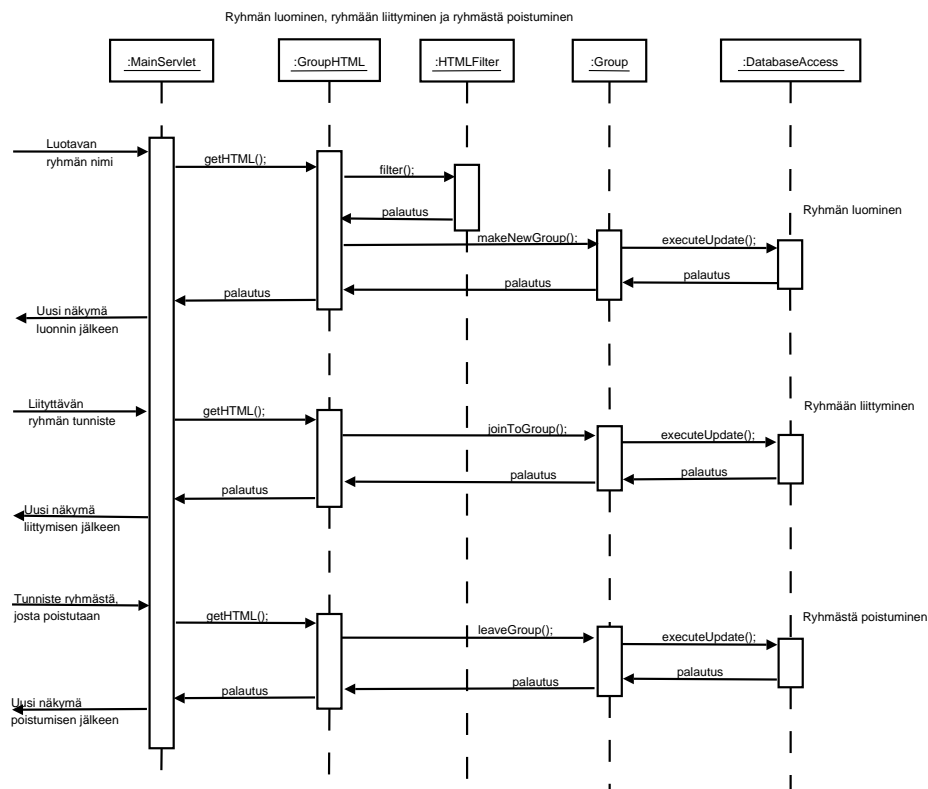
## 7 Ryhmäkomponentti

Ryhmäkomponentti tuottaa sovelluksessa käytettävien ryhmien hallintaan liittyvät näkymät. Komponentti huolehtii myös tietokantaan tehtävien muutosten ohjaamisesta tietokantakyselyiden suorituskomponentille. Ryhmäkomponentti tarjoaa seuraavat palvelut:

- Uuden ryhmän luominen
- Käyttäjän liittäminen ryhmään
- Käyttäjän poistaminen ryhmästä
- Käyttäjän omistamien ryhmien nimien näyttäminen
- Ryhmän omistajien näkyminen

Ryhmäkomponentin sekvenssikaavio on esitetty kuvassa 9 sivulla 14.

Ryhmäkomponentti koostuu kahdesta luokasta, joista GroupHTML-luokka tuottaa HTML-koodin Group-luokan palveluiden avulla.



Kuva 9: Ryhmäkomponentin sekvenssikaavio

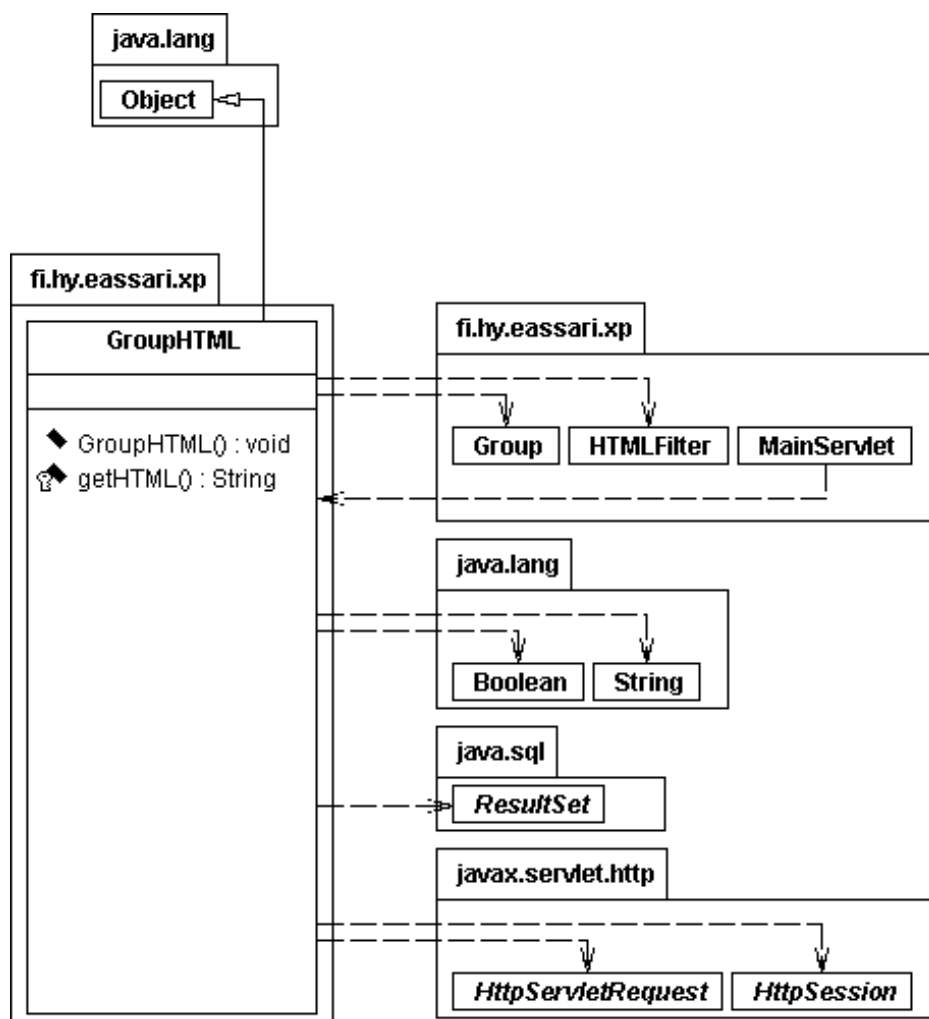
## 7.1 GroupHTML-luokan kuvaus

GroupHTML-luokka tuottaa MainServletin sille osoittaman pyynnön vastauksen. GroupHTML-luokan metodi getHTML tuottaa String-muotoista HTML-koodia, jonka perusteella seuraava näkymä tuotetaan. Luokka kutsuu getHTML-metodissaan Group-luokkaa, joka tekee pakkaukseen liittyvät tietokantakyselyt ja -lisäykset. GroupHTML:n luokkakaavio on esitetty kuvassa 10 sivulla 15.

### 7.1.1 GroupHTML-luokan metodien kuvaus

Tässä luvussa kuvataan GroupHTML-luokan metodit parametreineen ja paluuarvoineen.

<b>public GroupHTML()</b>	
Metodin kuvaus	Luokan konstruktori

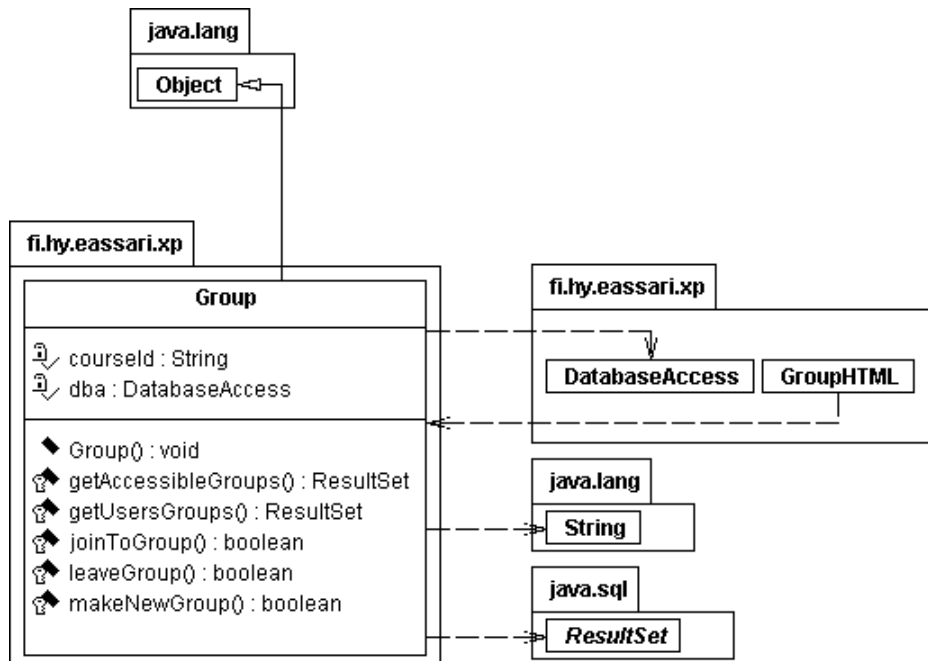


Kuva 10: GroupHTML-luokan luokkakaavio

<b>protected String getHTML(HttpServletRequest request, String userId, String courseId)</b>		
Metodin kuvaus	Metodi, jota MainServlet kutsuu, kun sen pyyntö kohdistuu ryhmäkomponentille. Metodi tuottaa Group-luokan avulla HTML-koodin, jonka MainServlet-luokka tarvitsee näkymän tuottamiseen.	
Parametrit	request userId courseId	Tiedot selaimelta saadusta pyynnöstä. Käyttäjän tunniste. Kurssin tunniste.
Paluuarvo	HTML-koodia, jonka perusteella näkymä luodaan selaimelle.	

## 7.2 Group-luokan kuvaus

Group-luokka toimii ModuleHTML-luokan apuluokkana. GroupHTML saa luokalta tarvitsemansa tiedot ja ohjaa sille pyydytyt päivitykset. Group-luokka käyttää tietokantaky-selyihinsä DatabaseAccess-luokkaa. Kuvassa 11 on esitetty luokan luokkakaavio.



Kuva 11: Group-luokan luokkakaavio

### 7.2.1 Group-luokan metodien kuvaus

Tässä luvussa kuvataan Group-luokan metodit parametreineen ja paluuarvoineen.

<b>public Group()</b>	
Metodin kuvaus	Luokan konstruktori

<b>protected ResultSet getAccessibleGroups(String userId, boolean isTeacher)</b>		
Metodin kuvaus	Metodi hakee parametrien perusteella kaikki ne ryhmät, joihin käyttäjä voi liittyä, ja palauttaa vastauksena ryhmien tunnisteet.	
Parametrit	userId isTeacher	Käyttäjän tunniste. Tieto siitä, onko käyttäjä opettaja, saako hän nähdä kaikki ryhmät.
Paluuarvo	Kaikkien niiden ryhmien tunnisteiden rivit, joihin käyttäjä voi liittyä.	

<b>protected ResultSet getUsersGroups(String userId, boolean isTeacher)</b>		
Metodin kuvaus	Metodi hakee kaikkien niiden ryhmien tunnisteet eli nimet, jotka käyttäjä omistaa.	
Parametrit	userId isTeacher	Käyttäjän tunniste. Tieto siitä, onko käyttäjä opettaja.
Paluuarvo	Käyttäjän omistamien ryhmien tunnisteiden rivit.	

<b>protected boolean makeNewGroup(String userId, String courseId, String groupId)</b>		
Metodin kuvaus	Metodi lisää uuden ryhmän tietokantaan ja palauttaa tiedon siitä, onnistuiko lisääminen. Samalla uuden ryhmän luoja liitetään tietokantaan.	
Parametrit	userId courseId groupId	Käyttäjän tunniste. Kurssin tunniste. Ryhmän tunniste.
Paluuarvo	Totuusarvo, joka kertoo, onnistuiko ryhmän luominen.	

<b>protected boolean joinToGroup(String userId, String courseId, String groupId)</b>		
Metodin kuvaus	Metodi liittää käyttäjän valittuun ryhmään tietokantaan ja palauttaa tiedon siitä, onnistuiko liittäminen.	
Parametrit	userId courseId groupId	Käyttäjän tunniste. Kurssin tunniste. Ryhmän tunniste.
Paluuarvo	Totuusarvo, joka kertoo, onnistuiko liittäminen.	

<b>protected boolean leaveGroup(String userId, String courseId, String groupId)</b>		
Metodin kuvaus	Metodi poistaa käyttäjän valitusta ryhmästä tietokannasta ja palauttaa tiedon siitä, onnistuiko poistaminen. Jos poistuja on ryhmän viimeinen jäsen, poistetaan myös ryhmä.	
Parametrit	userId courseId groupId	Käyttäjän tunniste. Kurssin tunniste. Ryhmän tunniste.
Paluuarvo	Totuusarvo, joka kertoo, onnistuiko poistaminen.	

## 8 Pakkauskomponentti

Pakkauskomponentti tuottaa sovelluksessa käytettävien pakkausten hallintaan liittyvät näkymät. Komponentti huolehtii myös tietokantaan tehtävien muutosten ohjaamisesta Tieto-

kantakyselyiden suorituskomponentille. Pakkauskomponentti tarjoaa seuraavat palvelut:

- Uuden pakkauksen luominen
- Tehtävän lisääminen pakkaukseen
- Tehtävän poistaminen pakkauksesta
- Tehtävien järjestäminen pakkauksessa
- Pakkauksen näyttäminen tekijälle
- Pakkauksen käyttöajan määrittäminen
- Pakkauksen näkyvyyden määrittäminen
- Pakkauksen tyypin määrittäminen
- Palautemoodin määrittäminen
- Pakkauksen tekijöiden näkyminen

Pakkauskomponentin sekvenssikaavio on esitetty kuvassa 12 sivulla 19.

Pakkauskomponentti koostuu kahdesta luokasta, joista `ModuleHTML`-luokka tuottaa HTML-koodin `Module`-luokan palveluiden avulla.

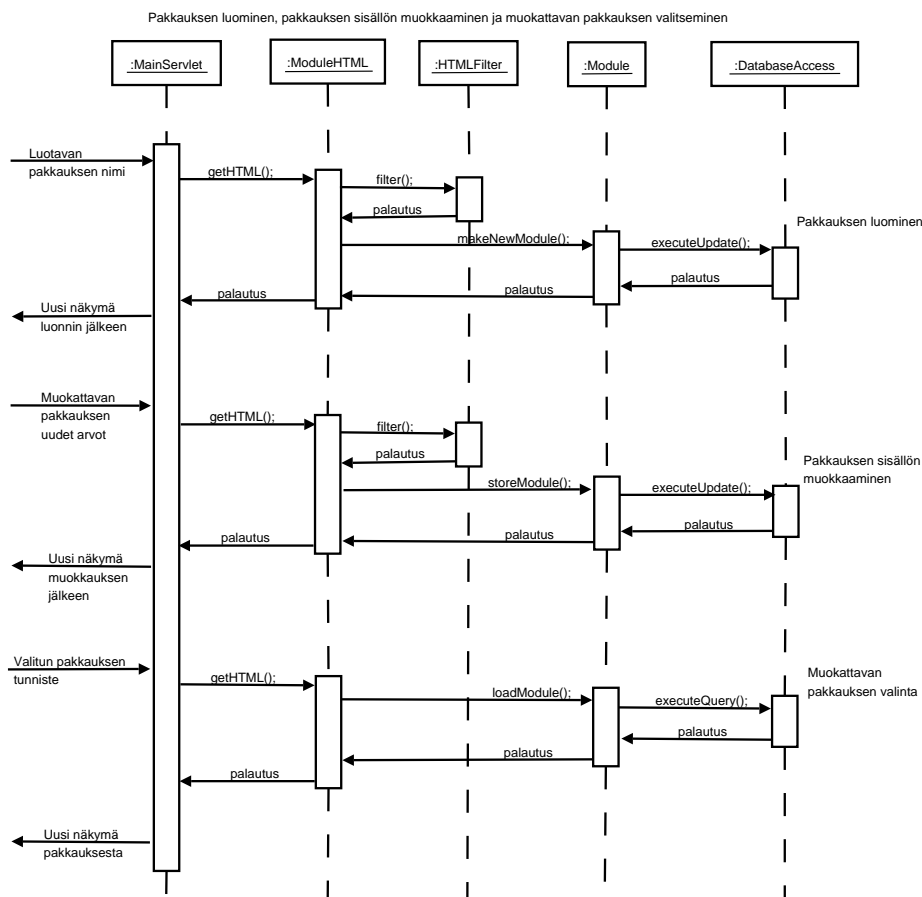
## 8.1 `ModuleHTML`-luokan kuvaus

`ModuleHTML`-luokka tuottaa `MainServlet`in sille osoittaman pyynnön vastauksen. `ModuleHTML`-luokan metodi `getHTML` tuottaa `String`-muotoista HTML-koodia, jonka perusteella seuraava näkymä tuotetaan. Luokka kutsuu `getHTML`-metodissaan `Module`-luokkaa, joka tekee pakkaukseen liittyvät tietokantakyselyt ja lisäykset. Kuvassa 13 sivulla 20 on esitetty luokan luokkakaavio.

### 8.1.1 `ModuleHTML`-luokan metodien kuvaus

Tässä luvussa kuvataan `ModuleHTML`-luokan metodit parametreineen ja paluuarvoineen.

<b>public ModuleHTML()</b>	
Metodin kuvaus	Luokan konstruktori



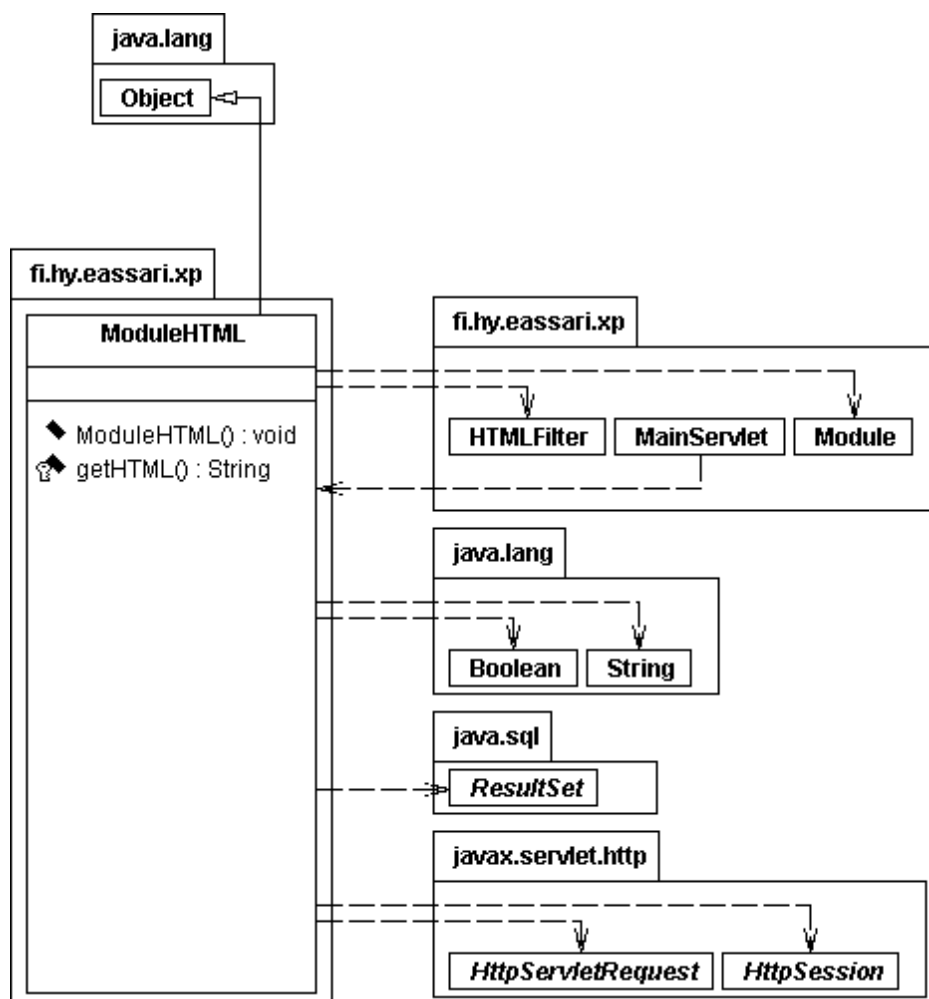
Kuva 12: Pakkauskomponentin sekvenssikaavio

<b>protected String getHTML(HttpServletRequest request, String userId)</b>		
Metodin kuvaus	Metodi tuottaa Module-luokan avulla HTML-koodin, jonka MainServlet-luokka tarvitsee näkymän tuottamiseen.	
Parametrit	request userId	Tiedot selaimelta saadusta pyynnöstä Käyttäjän tunniste
Paluuarvo	HTML-koodia, jonka perusteella näkymä luodaan selaimelle.	

## 8.2 Module-luokan kuvaus

Module-luokka toimii ModuleHTML-luokan apuluokkana. ModuleHTML saa luokalta tarvitsemansa tiedot ja ohjaa sille pyydyt päivitykset. Module-luokka käyttää tietokantakyselyihinsä DatabaseAccess-luokkaa. Kuvassa 14 sivulla 21 on esitetty Module-luokan





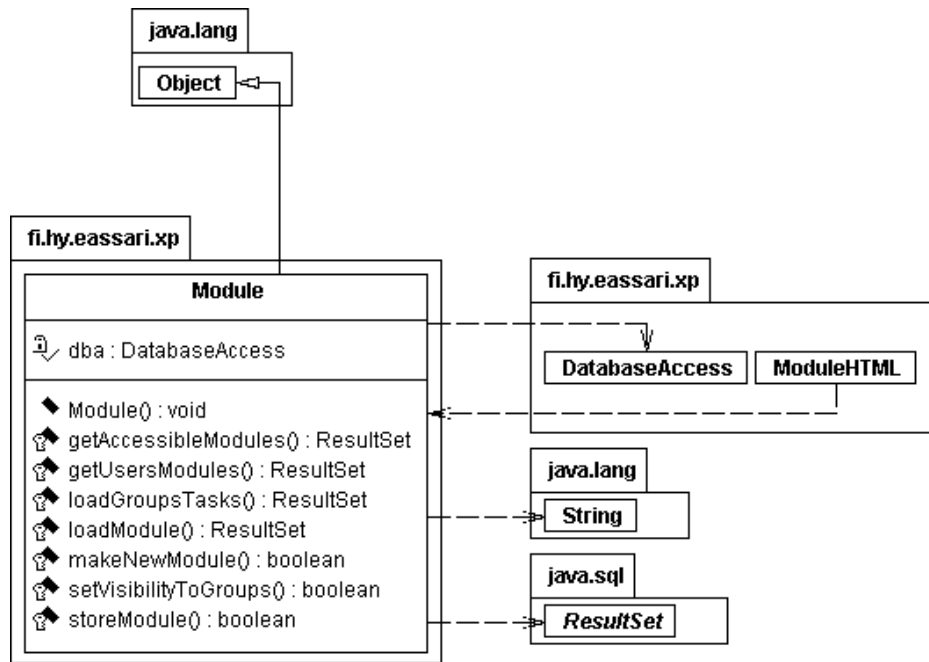
Kuva 13: ModuleHTML-luokan luokkakaavio

luokkakaavio.

### 8.2.1 Module-luokan metodien kuvaus

Tässä luvussa kuvataan Module-luokan metodit parametreineen ja paluuarvoineen.

<b>public Module()</b>	
Metodin kuvaus	Luokan konstruktori



Kuva 14: Module-luokan luokkakaavio

<b>protected boolean makeNewModule(String moduleId, String groupId, int moduleType)</b>		
Metodin kuvaus	Metodi luo uuden pakkauksen saamiensa parametrien perusteella ja palauttaa vastauksen siitä, onnistuiko pakkauksen luonti vai ei.	
Parametrit	moduleId groupId moduleType	Pakkauksen tunniste Pakkauksen omistavan ryhmän tunniste Pakkauksen tyyppi
Paluuarvo	Totuusarvo, joka kertoo, onnistuiko uuden pakkauksen luonti	

<b>protected ResultSet getUsersModules(String userId, boolean isTeacher)</b>		
Metodin kuvaus	Metodi pyytää DatabaseAccess-luokalta listan kaikista pakkauksista, jotka käyttäjä omistaa.	
Parametrit	userId isTeacher	Käyttäjän tunniste Tieto siitä, onko käyttäjä opettaja
Paluuarvo	Käyttäjän omistamien pakkausten rivit	

<b>protected ResultSet getAccessibleModules(String userId, boolean isTeacher)</b>		
Metodin kuvaus	Metodi pyytää DatabaseAccess-luokalta listan kaikista pakkauksista, jotka käyttäjä voi nähdä.	
Parametrit	userId isTeacher	Käyttäjän tunniste Tieto siitä, onko käyttäjä opettaja
Paluuarvo	Niiden pakkausten rivit, jotka käyttäjä voi nähdä	

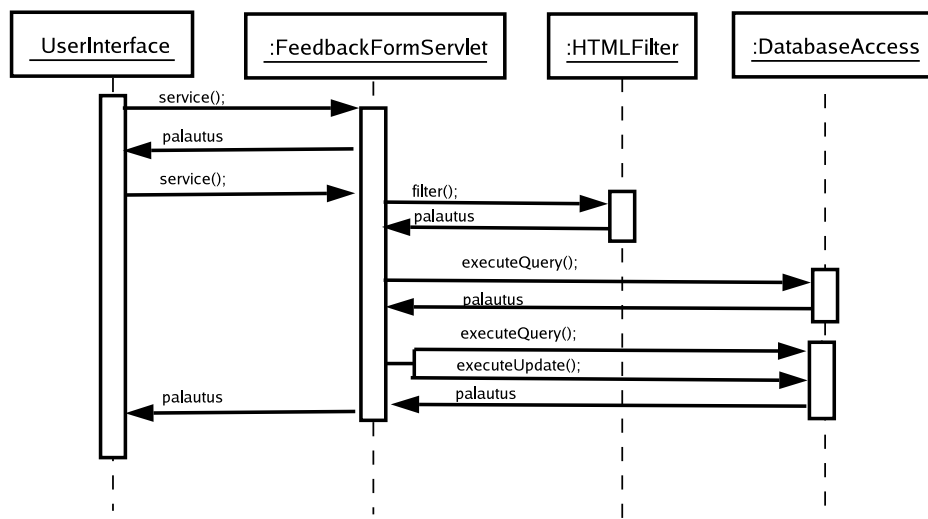
<b>protected ResultSet loadModule(String moduleId)</b>		
Metodin kuvaus	Metodi pyytää DatabaseAccess-luokalta kaikki tietyn pakkauksen tiedot ja palauttaa nämä.	
Parametrit	moduleId	Pakkauksen tunniste
Paluuarvo	Tietokannan rivi, joka kuvaa pyydettyä pakkausta	

<b>protected ResultSet loadGroupsTasks(String moduleId)</b>		
Metodin kuvaus	Metodi pyytää DatabaseAccess-luokalta kaikkia pakkauksen omistavan ryhmän tekemiä tehtäviä ja palauttaa nämä listana.	
Parametrit	moduleId	Pakkauksen tunniste
Paluuarvo	Tietokannan rivit kaikista tehtävistä, jotka pakkauksen omistavan ryhmän jäsenet ovat tehneet	

<b>protected boolean storeModule(String moduleId, String[] moduleData)</b>		
Metodin kuvaus	Metodi pyytää DatabaseAccess-luokkaa tallettamaan muokatun pakkauksen tietokantaan.	
Parametrit	moduleId moduleData	Pakkauksen tunniste Päivitettävän pakkauksen tiedot
Paluuarvo	Totuusarvo, joka kertoo, onnistuiko pakkauksen päivitys	

## 9 Palautelomakekomponentti

Palautelomakkeella kerätään tietoa tehtävien eri ominaisuuksista. Käyttäjä antaa palautetta täyttämällä HTML-lomakkeen, jossa kutakin ominaisuutta arvioidaan viisiportaisella asteikolla. Lomakkeessa on myös tekstikentät vapaamuotoista palautetta ja parannusehdotuksia varten. Palautelomakkeen kuva on liitteessä 4. Täytetyn lomakkeen tiedot tallennetaan tietokantaan feedback-tauluun. Jos sama käyttäjä antaa uudelleen palautetta tehtävästä, josta hän on jo antanut palautetta, uusi palaute tallennetaan vanhan päälle. Palautelomake esitetään joko automaattisesti tai kun käyttäjä ilmaisee haluavansa antaa



Kuva 15: Palautelomakkeen sekvenssikaavio

palautetta. Tieto ratkaistun tehtävän taskid:stä ja käyttäjän userid:stä välittyy FeedbackFormServlet-luokalle request-parametrin sisällä, kun luokkaa kutsutaan. Tässä projektissa toteutetaan palautelomakkeen näyttäminen ja annetun palautteen tallentaminen. Projektin tehtäviin ei kuulu FeedbackFormServlet-luokan kutsumisen toteuttaminen. Palautelomakekomponentti tarjoaa seuraavat palvelut:

- Palautelomakkeen näyttäminen
- Täytetyn lomakkeen vastaanottaminen
- Annetun palautteen tallentaminen tietokantaan

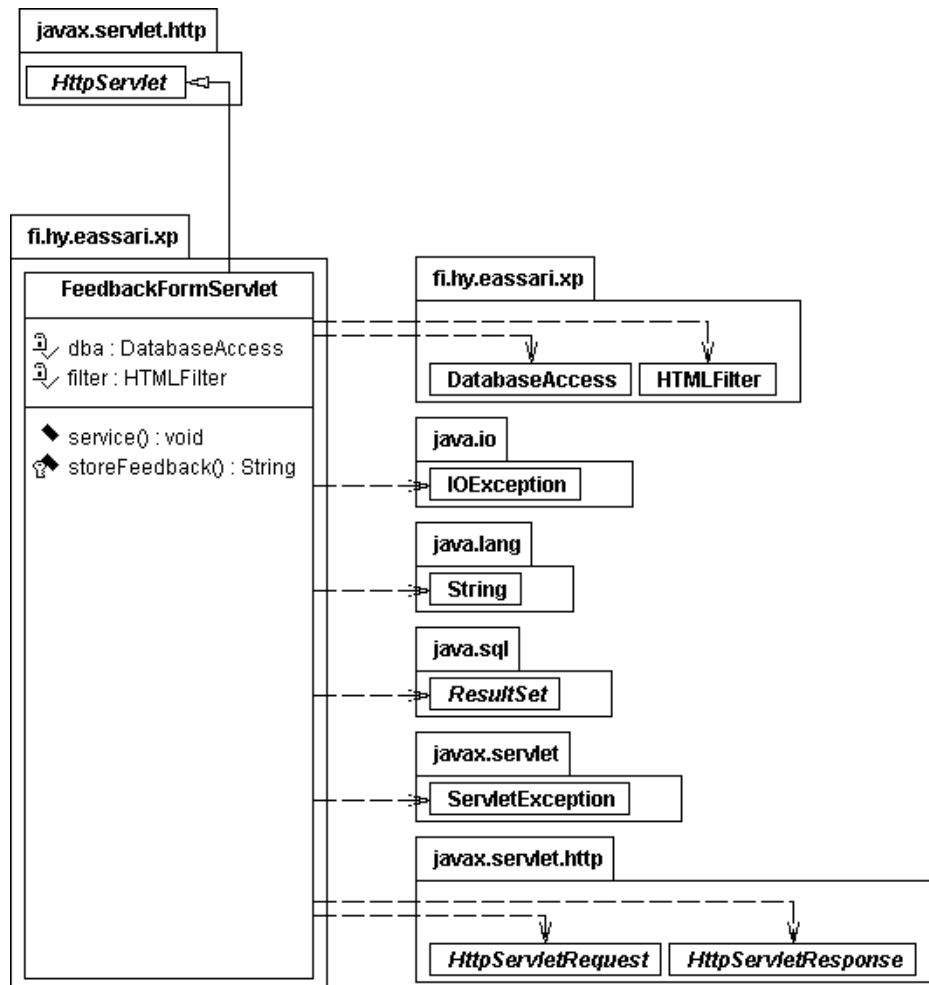
Palautelomakkeen sekvenssikaavio on esitetty kuvassa 15.

## 9.1 FeedbackFormServlet-luokan kuvaus

FeedbackFormServlet-luokka sisältää service- ja storeFeedback-metodit. Service-metodi näyttää palautelomakkeen ja vastaanottaa täytetyn lomakkeen. StoreFeedback-metodi tallentaa lomakkeen tiedot tietokantaan käyttäen DatabaseAccess-luokkaa. Kuvassa 16 sivulla 24 on esitetty luokan luokkakaavio.

### 9.1.1 FeedbackFormServlet-luokan metodien kuvaus

Tässä luvussa kuvataan FeedbackFormServlet-luokan metodit parametreineen ja paluuarvoineen.



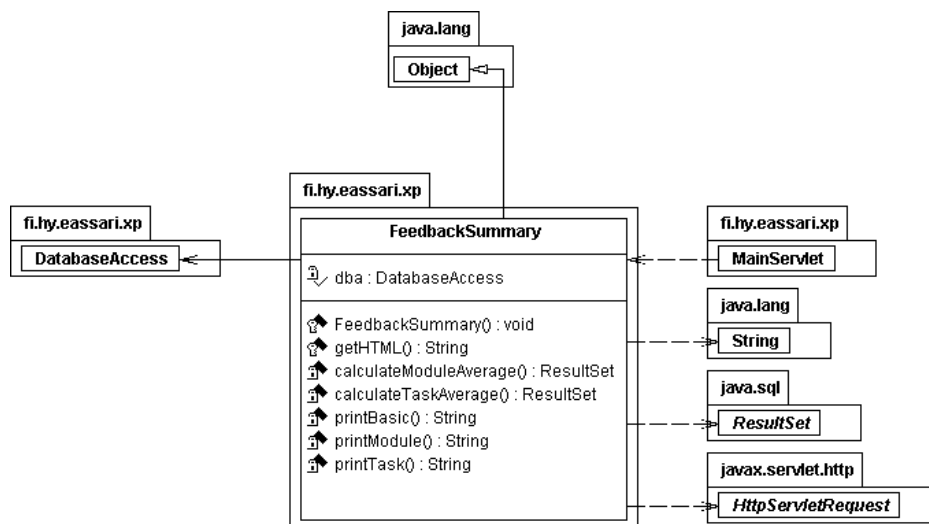
Kuva 16: FeedbackFormServlet-luokan luokkakaavio

<b>public void service(HttpServletRequest request, HttpServletResponse response)</b>		
Metodin kuvaus	Metodi esittää palautelomakkeen, vastaanottaa täytetyn lomakkeen ja välittää lomakkeen tiedot storeFeedback-metodille.	
Parametrit	request  response	Tiedot selaimelta saadusta pyynnöstä Kenttä, jolla voidaan tuottaa sisältöä vastaukseen

<b>protected boolean storeFeedback(String taskId, String userId, HttpServletRequest request)</b>		
Metodin kuvaus	Metodi pyytää DatabaseAccess-luokalta tiedon siitä, onko tietokannassa jo tallennettu palautevastaukset ko. taskId:llä ja userId:llä. Jos on, metodi pyytää DatabaseAccess-luokkaa tekemään muutokset olemassa oleviin tietoihin. Jos tallennuksia ei ole, metodi pyytää DatabaseAccess-luokkaa tallettamaan annetun palautteen tietokantaan.	
Parametrit	taskId userId  request	Tehtävä, johon palautetta annetaan Palautetta antavan henkilön tunniste, vastaa tietokannan sid-attribuuttia Täytetyn palautelomakkeen tiedot
Paluuarvo	Totuusarvo, joka kertoo, onnistuiko tietojen tallennus tai muutos	

## 10 Palauteyhteenvetokomponentti

Tässä luvussa kuvataan palauteyhteenvetokomponentin toteutus ja toiminta. Palauteyhteenvetojen on tarkoitus antaa tarkempaa tietoa siitä, mitä mieltä tehtävien ratkojat ovat olleet tehtävistä. Tarkemmat kuvaukset kysymyksistä löytyvät dokumentin liitteestä 4. Kuvassa 17 on esitetty luokan luokkakaavio.



Kuva 17: FeedbackSummary-luokan luokkakaavio

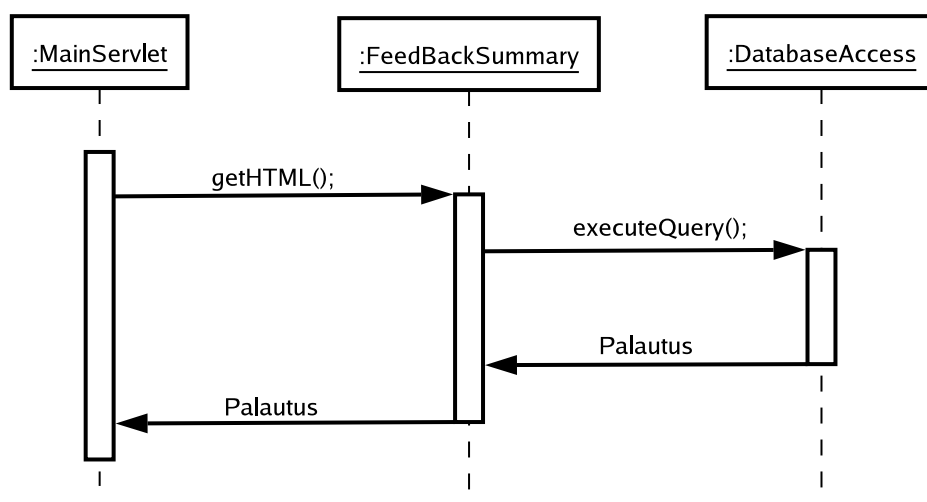
## 10.1 FeedbackSummary-luokan kuvaus

Palauteyhteenvetojen muodostus tapahtuu FeedbackSummary-luokassa. MainServlet kutsuu FeedbackSummary-luokan getHTML-metodia ja antaa tälle parametrina itse saamansa parametrit. Tämän jälkeen FeedbackSummary-luokka purkaa saadut parametrit ja niistä saatujen tietojen perusteella tulostaa yhden kolmesta mahdollisesta näkymästä.

FeedbackSummary-luokka tarjoaa seuraavanlaiset näkymät:

- Perusnäkö. Näkö, jossa näkyy kaikki kyseisen käyttäjän oikeuksien mahdollistamat pakkaukset. Mikäli puretuissa parametreissa ei ole mainittu tarkennettu näkö, niin getHTML-metodi kutsuu printBasic-metodia, joka tulostaa tämän näkö.
- Pakkausnäkö. Tässä näkössä nähdään jonkin valitun pakkauksen sisältö, eli pakkauksen tehtävät ja näiden saamien palautteiden keskiarvot. Tämä näkö saadaan, mikäli puretuissa parametreissa on mukana module-parametri, jonka arvo on jonkin pakkauksen tunniste. Tällöin getHTML-metodi kutsuu omaa printModule-metodia, jolla tulostetaan halutun pakkauksen sisältö.
- Tehtävänäkö. Tässä näkössä on nähtävillä yhden tehtävän saamat palautteet. Näkössä on nähtävillä kaikki saadut kirjalliset palautteet. Mikäli parametreissa on module-parametri, jolla on jonkin pakkauksen nimi, niin katsotaan, onko saaduissa parametreissa jonkin tehtävän nimi task-parametrissa. Jos molemmat löytyvät, niin getHTML-metodi kutsuu printTask-metodia, jolla tulostetaan halutun tehtävän saamat palautteet yksityiskohtaisemmin.

Kuvassa 18 on esitetty palauteyhteenvetokomponentin sekvenssikaavio.



Kuva 18: Palauteyhteenvetokomponentin sekvenssikaavio

### 10.1.1 FeedbackSummary-luokan metodien kuvaukset

FeedbackSummary-luokka sisältää seuraavat metodit:

<b>protected FeedbackSummary()</b>	
Metodin kuvaus	Luokan konstruktori

<b>protected String getHTML(HttpServletRequest request)</b>		
Metodin kuvaus	MainServlet kutsuu tätä metodia silloin, kun se on saanut pyynnön esittää jonkin palauteyhteen- vetonäkymän.	
Parametrit	request	Tarvittavat parametrit, joiden avulla metodi osaa tulostaa oikean näkymän
Paluuarvo	Metodi palauttaa pyydetyn palauteyhteen- vetonäkymän String-muotoisena.	

Seuraavat metodit ovat vain luokan omaan käyttöön. Niitä ei voida kutsua muista luokista suoraan.

<b>private String printBasic()</b>	
Metodin kuvaus	Metodia kutsutaan silloin, kun käyttäjä haluaa nähdä kaikki paketit palautesivulla.
Paluuarvo	Metodi palauttaa pyydetyn palauteyhteen- vetonäkymän String-muotoisena.

<b>private String printModule()</b>	
Metodin kuvaus	Metodia kutsutaan silloin, kun käyttäjä haluaa nähdä jonkin tietyn paketin sisältämien tehtävien palautteet.
Paluuarvo	Metodi palauttaa pyydetyn palauteyhteen- vetonäkymän String-muotoisena.

<b>private String printTask()</b>	
Metodin kuvaus	Metodia kutsutaan silloin, kun käyttäjä haluaa nähdä jonkin tietyn paketin sisältämän tehtävän saamat palautteet.
Paluuarvo	Metodi palauttaa pyydetyn palauteyhteen- vetonäkymän String-muotoisena.



<b>private ResultSet calculateModuleAverage(String moduleId)</b>		
Metodin kuvaus	Apumetodi, jolla lasketaan halutun pakkauksen palautekysymysten keskiarvo	
Parametrit	moduleId	Pakkauksen tunniste, jonka tehtävien keskiarvot lasketaan
Paluuarvo	Metodi palauttaa ResultSet-oliossa laskemansa keskiarvot.	

<b>private ResultSet calculateTaskAverage(String moduleId, String taskId)</b>		
Metodin kuvaus	Apumetodi, jolla lasketaan halutun tehtävän palautekysymysten keskiarvo	
Parametrit	moduleId taskId	Pakkauksen tunniste jossa tehtävä on Tehtävä jonka keskiarvot lasketaan
Paluuarvo	Metodi palauttaa ResultSet-oliossa laskemansa keskiarvot.	

## 11 QTI-muunnoskomponentti

Tässä luvussa esitetään, miten projektissa toteutetaan QTI-muuntamisen suorittava työkalu. Liitteessä 1 on tarkempi selostus siitä, miten tehtävät on tallennettu eAssari-järjestelmään ja mitkä eAssarin tehtävätyypit on mahdollista esittää QTI-muodossa. Tässä projektissa toteutetaan QTI-muunnos eAssarin valintatehtäville eli sellaisille tehtäville, joissa tehtävän ratkaisija valitsee vaihtoehdoista yhden tai useamman oikean vaihtoehdon. Projektissa ei toteuteta QTI-muodossa olevien tehtävien muunnosta eAssari-järjestelmän muotoon.

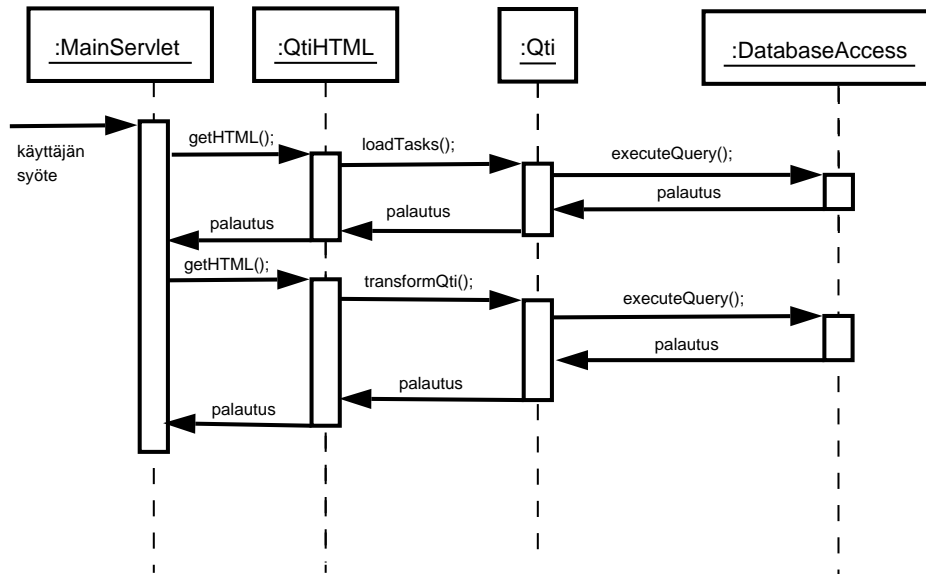
Kuvassa 19 sivulla 29 on esitetty muunnoksen kulku sekvenssikaaviona.

QTI-muunnoskomponentti koostuu kahdesta luokasta: QtiHTML- ja Qti-luokista. Näistä luokista edellinen tuottaa HTML-koodia jälkimmäisen palveluiden avulla.

### 11.1 QtiHTML-luokan kuvaus

QtiHTML-luokka luo käyttäjälle näytettävien HTML-sivujen pohjan MainServletin pyynnöstä: MainServlet kutsuu luokan getHTML-metodia, joka tuottaa string-tyyppisen HTML-merkkijonon. getHTML-metodi puolestaan kutsuu Qti-luokan metodeja, jotka tekevät tarvittavat tietokantakyselyt ja muuntavat valitut tehtävät QTI-muotoon.

MainServlet luo getHTML-metodin avulla kaksi käyttäjälle näytettävää HTML-sivua: toisella sivulla on lista käyttäjän tehtävistä, joista valitaan muunnettavat tehtävät, ja toisella on linkki tiedostoon, jossa muunnetut QTI-muotoiset tehtävät ovat.



Kuva 19: QTI-muunnoksen sekvenssikaavio

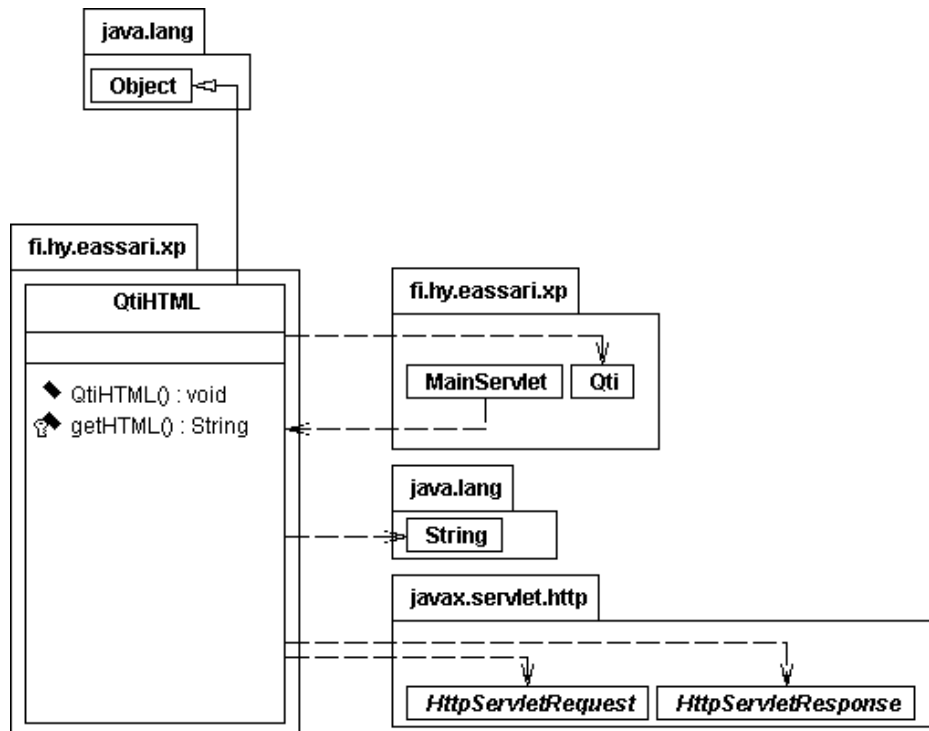
Kuvassa 20 sivulla 30 on esitetty luokan luokkakaavio.

### 11.1.1 QtiHTML-luokan metodien kuvaus

Tässä luvussa kuvataan QtiHTML-luokan metodit, niiden parametrit ja paluuarvot.

<b>public QtiHTML()</b>	
Metodin kuvaus	Luokan konstruktori

<b>protected String getHTML(HttpServletRequest request, HttpServletResponse response, String userId)</b>							
Metodin kuvaus	Metodi tuottaa Qti-luokan metodien avulla HTML-koodin, jonka MainServlet-luokka käyttää näkymän tuottamiseen.						
Parametrit	<table border="1"> <tr> <td>request</td> <td>Tiedot selaimelta saadusta pyynnöstä</td> </tr> <tr> <td>response</td> <td>Kenttä, jolla voidaan tuottaa sisältöä vastaukseen</td> </tr> <tr> <td>userId</td> <td>Käyttäjän tunniste</td> </tr> </table>	request	Tiedot selaimelta saadusta pyynnöstä	response	Kenttä, jolla voidaan tuottaa sisältöä vastaukseen	userId	Käyttäjän tunniste
request	Tiedot selaimelta saadusta pyynnöstä						
response	Kenttä, jolla voidaan tuottaa sisältöä vastaukseen						
userId	Käyttäjän tunniste						
Paluuarvo	HTML-koodia, jonka perusteella näkymä luodaan selaimelle.						



Kuva 20: QtiHTML-luokan luokkakaavio

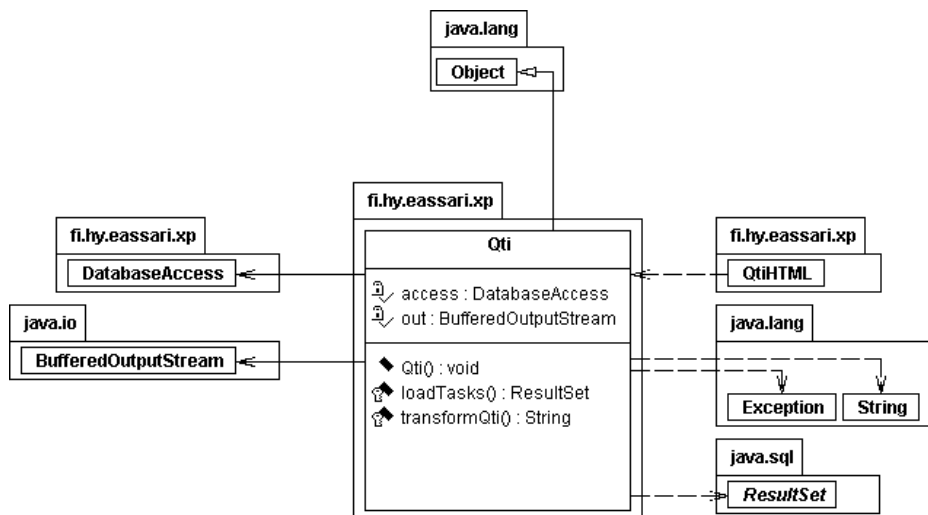
## 11.2 Qti-luokan kuvaus

Qti-luokka on QtiHTML-luokan apuluokka, joka tekee tietokantakyselyt ja muuntaa valitut tehtävät QTI-muotoon. QtiHTML-luokan metodi getHTML kutsuu Qti-luokan metodia loadTasks, joka palauttaa tiedot opettajalle kuuluvista tehtävistä. getHTML kutsuu myös transformQti-metodia, joka tekee apumetodiensa avulla tietokantakyselyt ja muunnettaviksi valittujen tehtävien QTI-muunnoksen. Qti-luokka käyttää tietokantakyselyihin DatabaseAccess-luokan metodia executeQuery. Kuvassa 21 sivulla 31 on esitetty luokan luokkakaavio.

### 11.2.1 Qti-luokan metodien kuvaus

Tässä luvussa kuvataan Qti-luokan metodit, niiden parametrit ja paluuarvot.

<b>public Qti()</b>	
Metodin kuvaus	Luokan konstruktori



Kuva 21: Qti-luokan luokkakaavio

<b>protected ResultSet loadTasks(String courseId)</b>	
Metodin kuvaus	Metodi pyytää DatabaseAccess-luokalta kaikkia kurssin opettajalle kuuluvia tehtäviä ja palauttaa näiden tiedot.
Parametrit	courseId      Kurssin tunniste
Paluuarvo	ResultSet-olio, joka sisältää tiedot kurssin opettajalle kuuluvista tehtävistä

<b>protected String transformQti(String[] tasknames)</b>	
Metodin kuvaus	Metodi muuntaa tehtävät, joiden nimet se saa parametrinään.
Parametrit	tasknames      Lista, jossa on muunnettavien tehtävien nimet
Paluuarvo	Merkkijono, joka kertoo sen tiedoston nimen, jossa QTI-muunnetut tehtävät ovat. Tiedoston nimi on ns. polkunimi.

Alla olevat metodit ovat luokan yksityisiä metodeja, joita ei voi kutsua luokan ulkopuolelta.

<b>private ResultSet getTaskdata(String[] tasknames)</b>		
Metodin kuvaus	Metodi hakee DatabaseAccess-luokan avulla kaikki muuntamisessa tarvittavat tehtävän tiedot.	
Parametrit	tasknames	Lista, jossa on muunnettavien tehtävien nimet
Paluuarvo	ResultSet-olio, joka sisältää muunnettavien tehtävien ne tiedot, joita muuntamisessa tarvitaan.	

<b>private String makeQti(ResultSet taskdata)</b>		
Metodin kuvaus	Metodi hakee DatabaseAccess-luokan avulla kaikki muuntamisessa tarvittavat tehtävän tiedot.	
Parametrit	taskdata	ResultSet-olio, joka sisältää muunnettavien tehtävien ne tiedot, joita muuntamisessa tarvitaan
Paluuarvo	Merkkijono, jossa on muunnettu tehtävä QTI-muodossa.	

<b>private String makeFile(String[] qtiTasks)</b>		
Metodin kuvaus	Metodi luo tiedoston ja kirjoittaa tiedostoon kaikki muunnetut QTI-muotoiset tehtävät.	
Parametrit	qtiTasks	Lista, jossa ovat muunnetut QTI-muotoiset tehtävät.
Paluuarvo	Merkkijono, joka kertoo sen tiedoston nimen, jossa QTI-muunnetut tehtävät ovat. Tiedoston nimi on ns. polkunimi.	

## Lähteet

- IMS      IMS Global Learning Consortium, *IMS Question & Test Interoperability Specification*. <http://www.imsglobal.org/question/index.cfm>. [17.9.2004]
- Lai03    Laine, H., eassari - practicing, assessment and exam engine, 2003. <http://www.cs.helsinki.fi/u/laine/eassari.pdf>. [17.9.2004]

# Liite 1. Selvitys QTI-muunnoksesta

## Johdanto

Tämän kirjoituksen tarkoituksena on selvittää, voidaanko eAssarin tehtävätyyppejä muuntaa IMS-organisaation QTI (Question and Test Interoperability) -muotoon. Aluksi esitellään, millaisia tehtävätyyppejä eAssarissa tällä hetkellä on ja miten tehtävät on tallennettu järjestelmässä. (Nämä tiedot on saatu keväällä 2004 tehdystä Assari-ohjelmistotuotantoprojektin Suunnitteludokumentista<sup>1</sup> sekä suoraan bodbacka.cs.helsinki.fi-palvelimella olevasta eAssarin testitietokannasta.) Sitten tarkastellaan, onko QTI:n avulla mahdollista esittää samantyyppisiä tehtäviä kuin eAssarissa on ja onko QTI-muotoisia tehtäviä mahdollista muuntaa eAssariin. Lopuksi esitellään, miten muunnos voidaan toteuttaa.

## eAssarin tehtävätyypit

Tämänhetkinen eAssari-järjestelmä sisältää seuraavat tehtävätyypit:

- monivalintatehtävä (vaihtoehtoista valitaan yksi)
- valikkotehtävä (vaihtoehtoista voidaan valita yksi tai useampi, myös nolla tai kaikki vaihtoehdot käyvät)
- järjestystehtävä (annetut vaihtoehdot järjestetään)
- aukkojentäyttötehtävä (yksi tai useampi aukko, johon syötetään vapaa vastaus)

Näistä kaksi ensin mainittua on tallennettu nykyiseen järjestelmään samantyyppisinä valintatehtävinä. Erona on se, että monivalintatehtävässä oikeita vaihtoehtoja on vain yksi, valikkotehtävässä oikeita vaihtoehtoja voi olla useampia tai ei yhtään.

Tehtävätyyppien yleistiedot määritellään tietokannassa `tasktype`-taulussa ja tehtävätyyppeihin liittyvien attribuuttien tiedot esitetään `taskattributes`-taulussa. Taulussa `tasktype` on sarake `typename`, joka ilmoittaa tehtävätyypin nimen. Tällä hetkellä järjestelmässä ovat seuraavat tehtävätyypit: `optiontask`, `blankfilltask` ja `orderingtask`.

Yksittäisten tehtävien yleiset tiedot tallennetaan `task`-tauluun. Sarake `tasktype` ilmoittaa tehtävän tehtävätyypin nimen, `taskid` tehtävän tunnuksen ja `taskname` tehtävän nimen. Tehtäväkohtaiset tiedot, kuten tehtävän anto ja oikeat vastaukset, tallennetaan `attributevalues`-tauluun, jossa sarake `objectid` ilmoittaa kohteen (eli `task`-taulussa `taskid`), johon attribuutti liittyy. Tehtävänanto eli itse kysymys tallennetaan `attributevalues`-tauluun sarakkeeseen `attributevalue`, jolloin saman rivin `attributename`-sarakkeen arvo on 'task'. Taulussa `attributevalues` sarakkeella `objecttype` on arvo 'T', kun rivi liittyy tehtävään.

---

<sup>1</sup>Assari-ohjelmistotuotantoprojekti, Suunnitteludokumentti, 11.5.2004, versio 1.1, [http://www.cs.helsinki.fi/group/assari/dokumentit/suunnitteludokumentti/Suunnitteludokumentti\\_v1.1.pdf](http://www.cs.helsinki.fi/group/assari/dokumentit/suunnitteludokumentti/Suunnitteludokumentti_v1.1.pdf)

Esimerkiksi valintatehtävien vaihtoehdot ovat `attributevalues`-taulussa sarakkeessa `attributevalue` sellaisella rivillä, jossa sarakkeen `attributename` arvo on `'option#'`. (Tässä # on 1, 2, 3, jne.). Vastaukset, 'Y' tai 'N', ilmoitetaan samassa taulussa sarakkeessa `attributevalue` sellaisella rivillä, jossa sarakkeen `attributename` arvo on `'isselected#'`. Aukkojentäyttötehtävissä puolestaan täytettävä teksti on `attributevalue` arvona rivillä, jolla `attributename` arvo on `'text'`. Aukkoihin tulevat sanat ovat riveillä, joilla `attributename` on `'gap#'`. Järjestystehtävissä järjestettävät oliot esitetään riveillä, joilla `attributename` on `'object#'`, ja olioiden oikeat paikat kerrotaan riveillä, joilla `attributename` on `'placeofobject#'`. Tehtäviin liittyvä yleinen palaute esitetään riveillä, joilla `attributename` on `'positivefeedback'` tai `'negativefeedback'`.

## QTI:n avulla esitettäviä tehtävätyyppejä

Tässä käytetty QTI:n versio on v.2.0 Public Draft. Lähteenä on käytetty seuraavia dokumentteja: *IMS Question and Test Interoperability: Information Model* ja *IMS Question and Test Interoperability: Item Implementation Guide*.<sup>2</sup>

QTI:ssa voidaan esittää kaikki eAssarissa tällä hetkellä olevat tehtävätyypit eli valinta-, järjestys- ja aukkojentäyttötehtävät melko pitkälle samantyyppisinä. Valintatehtävissä vaihtoehdot esitetään `choiceInteraction`-elementin avulla, järjestystehtävät `orderInteraction`- ja aukkojentäyttötehtävät `textEntryInteraction`-elementtien avulla. QTI-muotoisiin tehtäviin voidaan myös liittää palautetta. Esimerkiksi monivalintatehtävässä (yksi oikea vaihtoehto) voidaan antaa yleistä palautetta siitä, oliko vastaus oikein vai väärin (`modalFeedback`-elementti). Vaihtoehtoihin liittyvää palautetta (`feedbackInline`-elementti) voidaan näyttää, kun tehtävää ratkaistaan uudelleen. Epäselväksi jäi kuitenkin se, voidaanko valintatehtävien eri vaihtoehtoista antaa sekä positiivista että negatiivista palautetta samalla tavoin kuin eAssari-järjestelmässä. Näyttää siltä, että ainakaan järjestystehtävistä ei voisi QTI:n avulla antaa yhtä monipuolista palautetta kuin eAssarissa. Lisäksi tehtävien pisteityksen muuntaminen ei onnistu, koska eAssarin antamia pisteitä ei saada tietokannasta. Pisteitysongelma voidaan kuitenkin ratkaista tässä projektissa siten, että annetaan oikealle vastaukselle yksi piste ja väärälle nolla pistettä.

Tehtävät esitetään QTI:n avulla `assessmentItem`-elementissä, jonka `identifier`-attribuutti kertoo tehtävän tyypin ja `title`-attribuutti tehtävän nimen. Oikeat vastaukset esitetään `responseDeclaration`-elementissä ja näytettävä kysymys tai tehtävän anto esitetään `itemBody`-elementissä. Vastausten käsittelyyn liittyvät tiedot esitetään `responseProcessing`-elementissä. Vastausten käsittelyyn voidaan käyttää valmiita malleja (jolloin annetaan mallien URL) tai käsittely voidaan määrittää itse. eAssarin tehtävien muuntamisessa `responseProcessing`-elementtiä tarvitaan ainakin palautteen näyttämistä varten. Valmiissa malleissa tehtävät yleensä pisteitetään siten, että oikeasta vastauksesta saa yhden pisteen ja väärästä nolla pistettä.

<sup>2</sup>[http://www.imsglobal.org/question/qti\\_item\\_v2p0pd/infomodel.html](http://www.imsglobal.org/question/qti_item_v2p0pd/infomodel.html)  
[http://www.imsglobal.org/question/qti\\_item\\_v2p0pd/implementation.html](http://www.imsglobal.org/question/qti_item_v2p0pd/implementation.html) [26.10.2004].

eAssarin tehtävät on siis mahdollista muuntaa QTI-muotoon melko samanlaisina. Eroja on tehtävien pisteityksessä ja mahdollisesti palautteiden esittämisessä. Sen sijaan QTI-muotoisten tehtävien muuntaminen eAssarin muotoon ei ole kuitenkaan mahdollista tai mielekäästä, koska eAssarin tehtävissä palautteiden anto on monipuolisempaa kuin QTI-tehtävissä.

## QTI-muunnoksen toteuttaminen

eAssarissa olevat tehtävät voidaan muuntaa QTI-muotoon seuraavasti:

Käyttäjät valitsee muunnettavat tehtävät hänelle näytettävästä tehtävälisteristä. Valittujen tehtävien tunnukset `taskid:t` ja tehtävätyypit `tasktypet` haetaan tietokannasta. Sen jälkeen tietokannasta haetaan yksittäisiin tehtäviin liittyvät tiedot `attributevalues`-taulusta ja tietojen avulla kirjoitetaan QTI-muotoiset tehtävät tiedostoon. Kirjoittaminen tehdään lisäämällä tietokannasta haetut tiedot tehtävätyyppien QTI-pohjiin. Lopuksi tiedosto palautetaan käyttäjälle.

Alla olevassa taulukossa on esitetty, miten eAssarin optiontask-tyyppinen tehtävä voidaan muuntaa QTI-muotoon. Ensin eAssarin tietokannasta haetaan muunnettavan tehtävän tunnus eli `taskid` taulusta `task`. Tehtävän määrittelytiedot haetaan `attributevalues`-taulusta etsimällä ne rivit, joissa `objecttype` on 'T' ja `objectid` on sama kuin `taskid`. Taulukossa on vasemmalla `attributevalues`-taulun `attributename`-sarakkeen arvo ja oikealla on se QTI-elementti, jonka avulla kyseisen attribuutin arvo esitetään QTI-muotoisessa tehtävässä. QTI-elementit on esitetty siten, että ensimmäisenä mainittu elementti on puurakenteessa ylimpänä ja viimeksi mainittu alimpana.

attributename	QTI-elementit
task	itemBody: choiceInteraction: prompt
option#	itemBody: choiceInteraction: simpleChoice
isselected#, arvo 'Y'	responseDeclaration: correctResponse: value
positivefeedback	modalFeedback (attr. show)
positivefeedback#	itemBody: choiceInteraction: simpleChoice: feedbackInline (attr. show)
negativefeedback	modalFeedback (attr. hide)
negativefeedback#	itemBody: choiceInteraction: simpleChoice: feedbackInline (attr. hide)

Näiden elementtien lisäksi QTI-muotoisessa tehtävässä pitäisi olla `responseProcessing`-elementti, jotta vaihtoehtoihin liittyviä palautteita voidaan näyttää.

Tehtävistä, joissa on sekä suomen- että englanninkieliset tehtävänannot, tehdään kaksi erillistä tehtävää, jotka erotetaan toisistaan siten, että tehtävän nimen loppuun lisätään kielen tunniste FI tai EN.



## **Liite 2. Testaussuunnitelma**

### **A Johdanto**

AssariXP on Helsingin yliopiston Tietojenkäsittelytieteen laitoksen ohjelmistotuotanto-projekti. Projektin pohjana toimii asiakkaan, Harri Laineen, kehittämä eAssari-opetusohjelmisto [Lai03]. Projektissa toteutetaan tehtävien paketoinnin mahdollistava välineistö ja yksinkertainen palautelomake tehtäville. Projektissa toteutetaan myös IMS-organisaation määrittämän QTI (Question and Test Interoperability) -standardin [IMS] mukainen QTI-muunnostyökalu, jolla järjestelmässä olevia tehtäviä voidaan muuntaa QTI-muotoisiksi.

Tämä dokumentti on projektin testaussuunnitelma, jonka mukaan pyritään toimimaan ohjelmistoa testattaessa. Testauksella on tarkoitus löytää ohjelmistossa olevia virheitä ja varmistua ohjelmiston oikeellisesta toiminnasta yleisissä tilanteissa. Ohjelmiston virheettömyyttä ei pystytä testaamalla osoittamaan.

Luvussa B esitellään testauksessa käytettävät välineet. Testausstrategia esitellään luvussa C. Luvussa D käydään läpi testausraporttien muoto ja luvussa E esitellään suoritettavia testitapauksia.

### **B Testauksessa käytettävät välineet**

Yksikkötestaus voidaan suorittaa käyttämällä JUnit-testauskehystä, jolloin testit on helpompia eritellä toteutuksesta. Tällöin myös testien uusiminen on helpompaa.

Vaihtoehtoisesti testit suorittaa kyseisen luokan toteutuksesta vastaava henkilö omalla laitteistollaan parhaalla näkemällään tavalla. Tällöin luokan toteutuksen tehnyt henkilö on vastuussa siitä, että kaikki luokan metodit testataan huolella ja että kaikki testit säilytetään, jotta testit voidaan myöhemmin tarvittaessa uusida.

### **C Testausstrategia**

Tässä luvussa esitellään testauksessa käytettävä strategia. Ensin kuvataan testausstrategi-aa yleisesti, minkä jälkeen tarkastellaan kaikkia testausvaiheita erikseen.

#### **C.1 Yleistä**

Testien suunnittelu ja toteutus aloitetaan mahdollisimman aikaisessa vaiheessa. Suosituk-sena on, että testit on suunniteltu ennen komponentin ohjelmoimista, mutta tämä ei ole välttämätöntä. Testejä ajetaan ohjelmoinnin ohella. Testit olisi hyvä ajaa toiminnallisen kokonaisuuden ohjelmoinnin (esimerkiksi metodin ohjelmoinnin) jälkeen.

Suurten metodien kohdalla testejä olisi ajettava useammin, aina välittömästi suurten muutosten jälkeen. Yksinkertaisia metodeja, jotka pelkästään palauttavat jonkinlaisen arvon, ei tarvitse erikseen testata ellei ole tarpeen varmistua palautettavan arvon oikeellisuudesta.

Testejä suunniteltaessa on kiinnitettävä huomiota testattavan kohteen reunaehtoihin ja syötteisiin. Jos esimerkiksi reunaehtona on, että hyväksytyt syötteet ovat välillä 0-100, metodi testataan ainakin seuraavilla syötteillä:

- luvulla, joka on  $<-1$  ja luvulla, joka on  $>101$
- luvuilla  $-1$  ja  $101$
- luvuilla  $0$  ja  $100$  eli raja-arvoilla
- luvuilla  $1$  ja  $99$
- luvulla, joka on välillä  $1-99$

Vastaavasti merkkijonoja testataan seuraavasti:

- tyhjällä merkkijonolla
- merkkijonolla, jossa on välilyöntejä
- merkkijonolla, jossa ei ole välilyöntejä
- merkkijonolla, jossa on kirjaimia ja numeroita
- merkkijonolla, jossa on kiellettyjä merkkejä (esim.  $@$ ,  $'$ ,  $''$ ,  $<$ ,  $>$  ja  $;$ )
- merkkijonolla, jossa on muita erikoismerkkejä

Lisäksi tulee kiinnittää huomiota mahdollisiin null-arvoihin ja tyhjiin syötteisiin ja toimintaan niiden kanssa.

SQL-kyselyiden toimivuus tulee varmistaa tietokannan rasitustestauksella. Tarkoituksena on, ettei tietokanta kaadu yllättävästi, kun järjestelmää käyttää useampi käyttäjä. Testaus tehdään tekemällä useita tietokantahakuja samaan aikaan, ja varmistamalla, että SQL-kyselyt ovat riittävän optimaalisia. Riittävä optimaalisuus tarkoittaa sitä, ettei isoja tietokantataulujen ristituloja tehdä missään vaiheessa tietokantahakua.

Luokkien generoimat HTML-tiedostot tulee validoida internetissä olevalla W3C-organisaation validaattorilla<sup>3</sup> (<http://validator.w3.org/>).

Kaikki testit tulee säilyttää, jotta ne voidaan toistaa tarvittaessa osoittamaan, että komponenttien testaus on huolella tehty.

---

<sup>3</sup>W3C Markup Validation Service

## C.2 Yksikkötestaus

Yksikkötestauksessa testataan jokainen luokka. Yksikkötestauksessa testataan yksittäisiä metodeja ja luokan yleistä toteutusta. Yksikkötestauksen suorittaa luokan toteutuksesta vastaava henkilö omalla laitteistollaan. Luokan toteutuksen tehnyt henkilö on siis vastuussa kyseisen luokan testauksesta ja toimivuudesta. Luokalle tehdyt testit tarkastaa ja hyväksyy joku toinen ryhmän jäsen, joka ei ole ollut kyseistä luokkaa toteuttamassa.

Testaus suoritetaan lausekattavasti käymällä kaikki silmukat, if-lauseet ja else-haarat läpi niin, että kaikki luokan lauseet tulevat testatuksi ainakin kerran. Kaikki muuttujat, metodit, paikalliset tietorakenteet ja rajapinnat tulee testata. Virhetilanteet testataan luomalla tarkoituksella virhetilanteita virheellisillä syötteillä ja tarkistamalla, että virheistä selvittää ongelmitta.

## C.3 Integraatiotestaus

Integraatiotestauksessa testataan luokkien yhteistoimintaa kokoamalla tietyistä luokista komponentteja. Oleellisin osa integraatiotestauksessa on rajapintatestaus, jossa etsitään virheitä luokkien välisestä rajapinnasta. Rajapintatestauksessa tarkastellaan, kommunikoivatko komponenttien luokat oikein keskenään, esimerkiksi välittävätkö ne oikeanlaisia parametrejä toisilleen.

## C.4 Järjestelmätestaus

Järjestelmätestauksessa testataan koko järjestelmää. Testauksen avulla varmistetaan toimivuus komponenttien ja laitteiston välillä. Testauksessa tulee kiinnittää huomiota järjestelmän suorituskykyyn, tietoturvaan ja toiminnallisuuteen.

## C.5 Hyväksymistestaus

Hyväksymistestauksessa testataan, miten toteutettu järjestelmä vastaa vaatimusmäärittelydokumentin vaatimuksia. Onnistuneessa hyväksymistestissä kaikki ennalta asetetut vaatimukset täyttyvät.

Hyväksymistestaus suoritetaan Tietojenkäsittelytieteen laitoksen laitteistolla.

## D Testauksen raportointi

Testausraportista tulee käydä ilmi seuraavanlaiset asiat:

Testaaja: Matti Meikäläinen

Päivämäärä: 26.10.2004

Mitä testattu: luoPakkaus()-metodi luokassa Pakkaus.java

Testin kuvaus: Testattiin luoPakkaus()-metodin toimintaa luokassa Pakkaus.java luomalla ”Pakkaus1”-niminen pakkaus.

Odotetut tulokset: Uusi pakkaus nimeltä ”Pakkaus1”

Havaitut virheet: Loi väärän nimisen pakkauksen, ”Pakkaus2”

Testaustulos: Testi epäonnistui

Testausraporteista kootaan testausdokumentti.

## **E Suoritettavat testitapaukset**

Tässä luvussa kuvataan komponenteille suoritettavia testitapauksia. Jotkut testitapaukset ovat käyttöliittymäriippuvaisia. Jos käyttöliittymä toteutetaan eri tavalla kuin on alunperin hahmoteltu, muuttuvat testitapaukset toisenlaisiksi kuin tässä on määritelty.

### **E.1 MainServlet-komponentti**

Oikeelliset tapaukset:

- käyttäjälle näytetään HTML-sivu, jossa on navigointipalkki ja ryhmä-, pakkaus-, palauteyhteenvedo- tai QTI-muunnoskomponentin luoma HTML-osuus

Virheelliset tapaukset:

- käyttäjälle näytetään HTML-sivu, josta puuttuu navigointipalkki tai ryhmä-, pakkaus-, palauteyhteenvedo- tai QTI-muunnoskomponentin luoma HTML-osuus

### **E.2 Navigointikomponentti**

#### **E.2.1 Navigointipalkki**

Oikeelliset tapaukset:

- käyttäjälle näytetään linkki ryhmäkomponenttiin
- käyttäjälle näytetään linkki pakkauskomponenttiin
- käyttäjälle näytetään linkki palauteyhteenvetokomponenttiin
- käyttäjälle näytetään uloskirjautumislinkki
- opettajalle näytetään linkki QTI-muunnoskomponenttiin

Virheelliset tapaukset:

- opiskelijalle näytetään linkki QTI-muunnoskomponenttiin

### **E.3 Käyttäjän syötteen validointikomponentti**

Käyttäjän syötteen validointikomponentin testitapaukset sisältyvät mm. ryhmäkomponentin testitapauksiin, jossa testataan saatujen syötteiden oikeellisuus. Tästä komponentista ei siis ole erikseen omia testitapauksia.

### **E.4 Tietokantakyselyiden suorituskomponentti**

Tietokantakyselyiden suorituskomponentti testataan ryhmä-, pakkaus-, palauteyhteenveto- ja QTI-muunnoskomponenttien testauksen yhteydessä.

### **E.5 Ryhmäkomponentti**

#### **E.5.1 Uuden ryhmän luominen**

Oikeelliset tapaukset:

- uuden ryhmän nimi on yhden merkin pituinen
- uuden ryhmän nimi on ”normaalin” pituinen
- uuden ryhmän nimi on maksimipituinen (20 merkkiä)
- uuden ryhmän nimi sisältää erikoismerkkejä

Virheelliset tapaukset:

- uuden ryhmän nimi on jo olemassa tietokannassa
- uuden ryhmän nimi on nollan merkin pituinen
- uuden ryhmän nimi on 'null'
- uuden ryhmän nimi on yli maksimipituuden (20 merkkiä)
- uuden ryhmän nimi sisältää kiellettyjä merkkejä

#### **E.5.2 Käyttäjän liittäminen ryhmään**

Oikeelliset tapaukset:

- käyttäjä ei kuulu mihinkään ryhmään ja yksi ryhmä on valittuna
- käyttäjä kuuluu jo yhteen ryhmään ja yksi ryhmä on valittuna
- käyttäjä kuuluu useampaan ryhmään ja yksi ryhmä on valittuna

Virheelliset tapaukset:

- yhtään ryhmää ei ole olemassa
- mitään ryhmää ei ole valittuna
- useampi ryhmä on valittuna

### **E.5.3 Käyttäjän poistaminen ryhmästä**

Oikeelliset tapaukset:

- käyttäjä kuuluu yhteen ryhmään ja yksi ryhmä on valittuna
- käyttäjä kuuluu useampaan ryhmään ja yksi ryhmä on valittuna

Virheelliset tapaukset:

- käyttäjä ei ennestään kuulu mihinkään ryhmään
- mitään ryhmää ei ole valittuna
- useampi ryhmä on valittuna

## **E.6 Pakkauskomponentti**

### **E.6.1 Uuden pakkauksen luominen**

Oikeelliset tapaukset:

- uuden pakkauksen nimi on yhden merkin pituinen
- uuden pakkauksen nimi on ”normaalin” pituinen
- uuden pakkauksen nimi on maksimipituinen (20 merkkiä)
- uuden pakkauksen nimi sisältää erikoismerkkejä

Virheelliset tapaukset:

- uuden pakkauksen nimi on olemassa tietokannassa
- uuden pakkauksen nimi on nollan merkin pituinen
- uuden pakkauksen nimi on 'null'
- uuden pakkauksen nimi on yli maksimipituuden (20 merkkiä)
- uuden pakkauksen nimi sisältää kiellettyjä merkkejä

### **E.6.2 Tehtävän lisääminen pakkaukseen**

Oikeelliset tapaukset:

- pakkaus on tyhjä, valittuna on yksi tehtävä, on olemassa ainakin yksi tehtävä
- pakkaus sisältää yhden tehtävän, valittuna on yksi tehtävä, on olemassa ainakin yksi tehtävä
- pakkaus sisältää useampia tehtäviä, valittuna on yksi tehtävä, on olemassa ainakin yksi tehtävä

Virheelliset tapaukset:

- valittuna ei ole yhtään tehtävää
- valittuna on useampia tehtäviä
- olemassa olevia tehtäviä ei ole
- pakkauksessa jo olemassa olevan tehtävän lisääminen samaan pakkaukseen

### **E.6.3 Tehtävän poistaminen pakkauksesta**

Oikeelliset tapaukset:

- valittuna on yksi tehtävä, kun tehtäviä on yksi
- valittuna on yksi tehtävä, kun tehtäviä on useampia

Virheelliset tapaukset:

- pakkaus ei sisällä yhtään tehtävää
- valittuna ei ole yhtään tehtävää
- valittuna on useampia tehtäviä

### **E.6.4 Tehtävien järjestäminen pakkauksessa**

Oikeelliset tapaukset:

- pakkaus sisältää useamman tehtävän ja yksi tehtävä on valittuna

Virheelliset tapaukset:

- pakkaus ei sisällä tehtäviä
- pakkaus sisältää yhden tehtävän
- mitään tehtävää ei ole valittuna
- useampi tehtävä on valittuna

### **E.6.5 Pakkauksen käyttöajan määrittäminen**

Oikeelliset tapaukset:

- alkukäyttöaika on tänään ja loppukäyttöaika on myöhemmin samana päivänä
- alkukäyttöaika on tänään ja loppukäyttöaika on myöhemmin tulevaisuudessa
- alkukäyttöaika on tulevaisuudessa ja loppukäyttöaika on sen jälkeen
- alku- ja loppukäyttöaika rajoittuvat kurssin voimassaoloajan mukaan
- aikasyötteet ovat numeroita
- tunnit ovat välillä 0-23 ja minuutit 0-59
- päivät ovat välillä 1-31, kuukaudet 1-12
- tammi-, maalis-, touko-, heinä-, elo-, loka- ja joulukuun päivät ovat välillä 1-31
- huhti-, kesä-, syys- ja marraskuun päivät ovat välillä 1-30
- helmikuun päivät ovat välillä 1-28 (1-29 karkausvuosina)

Virheelliset tapaukset:

- alkukäyttöaika on jo mennyt nykyhetkestä
- loppukäyttöaika on jo mennyt nykyhetkestä
- loppukäyttöaika on ennen alkukäyttöaikaa
- alkukäyttöaika on sama kuin loppukäyttöaika
- alku- ja loppukäyttöaika rajoittuvat kurssin voimassaoloajan ulkopuolelle
- aikasyötteet ovat muita merkkejä kuin numeroita
- tunnit ovat alle 0 tai yli 23 ja minuutit ovat alle 0 tai yli 59
- päivät ovat alle 1 tai yli 31, kuukaudet alle 1 tai yli 12
- huhti-, kesä-, syys- ja marraskuussa on 31. päivä
- helmikuussa on 30. tai 31. päivä
- helmikuussa on 29. päivä, vaikka ei ole karkausvuosi



### **E.6.6 Pakkauksen näkyvyyden määrittäminen**

Oikeelliset tapaukset:

- yksi ryhmä, jolle pakkaus julkaistaan, on valittuna
- useampi ryhmä, jolle pakkaus julkaistaan, on valittuna

Virheelliset tapaukset:

- yhtään ryhmää, jolle pakkaus julkaistaan, ei ole valittuna
- ei ole olemassa yhtään ryhmää, jolle pakkaus julkistettaisiin

### **E.6.7 Pakkauksen tyypin määrittäminen**

Oikeelliset tapaukset:

- pakkauksen tyyppi on yleinen, käyttäjänä on opettaja
- pakkauksen tyyppi on harjoitustehtävä, käyttäjänä on opettaja
- pakkauksen tyyppi on tentti, käyttäjänä on opettaja
- pakkauksen tyyppi on yleinen, käyttäjänä on opiskelija

Virheelliset tapaukset:

- pakkauksen tyyppi on harjoitustehtävä, käyttäjänä on opiskelija
- pakkauksen tyyppi on tentti, käyttäjänä on opiskelija

### **E.6.8 Palautemoodin määrittäminen**

Oikeelliset tapaukset:

- käyttäjä on opettaja ja palautemoodi on valittuna
- käyttäjä on opettaja ja palautemoodi ei ole valittuna
- käyttäjä on opiskelija ja palautemoodi on valittuna

Virheelliset tapaukset:

- käyttäjä on opiskelija ja palautemoodi ei ole valittuna

## **E.7 Palautelomakekomponentti**

### **E.7.1 Palautteen antaminen**

Oikeelliset tapaukset:

- kaikkiin palautelomakkeen kohtiin vastataan
- vain joihinkin palautelomakkeen kohtiin vastataan
- vain yksi kohta on valintakysymyksessä valittuna
- valintakysymyksessä yhtään kohtaa ei ole valittuna
- tekstikenttä sisältää sallitun määrän tekstiä
- tekstikenttä ei sisällä mitään tekstiä
- lomake tyhjennetään tyhjennä-painikkeella
- lomake lähetetään lähetä-painikkeella
- käyttäjä antaa palautetta tehtävästä, josta hän ei ole vielä antanut palautetta, mutta joku toinen on
- käyttäjä antaa palautetta tehtävästä, josta kukaan ei ole vielä antanut palautetta
- käyttäjä antaa palautetta tehtävästä, josta hän on jo antanut palautetta

Virheelliset tapaukset:

- useampi kuin yksi vaihtoehto on valittuna valintakysymyksessä
- tekstikenttä sisältää merkkejä, jotka eivät ole sallittuja
- tekstikenttä sisältää enemmän tekstiä kuin on vapaan tekstin enimmäisraja

## **E.8 Palauteyhteenvetokomponentti**

Oikeelliset tapaukset:

- näytetään kaikki omistetut pakkaukset
- näytetään valitun pakkauksen palautteen
- näytetään valitun tehtävän tehtäväkohtaisen palautteen

Virheelliset tapaukset:

- valitun pakkauksen yhteenvetoa ei näy tai löydy
- valitun tehtävän palautetta ei näy tai löydy
- näytetään kaikki pakkaukset, ei vain omistetut pakkaukset
- näytetään useamman pakkauksen palautteen, vaikka valittuna on yksi pakkaus
- näytetään useamman tehtävän palautteen, vaikka valittuna on yksi tehtävä

## **E.9 QTI-muunnoskomponentti**

### **E.9.1 Valittujen tehtävien muuntaminen QTI-muotoon**

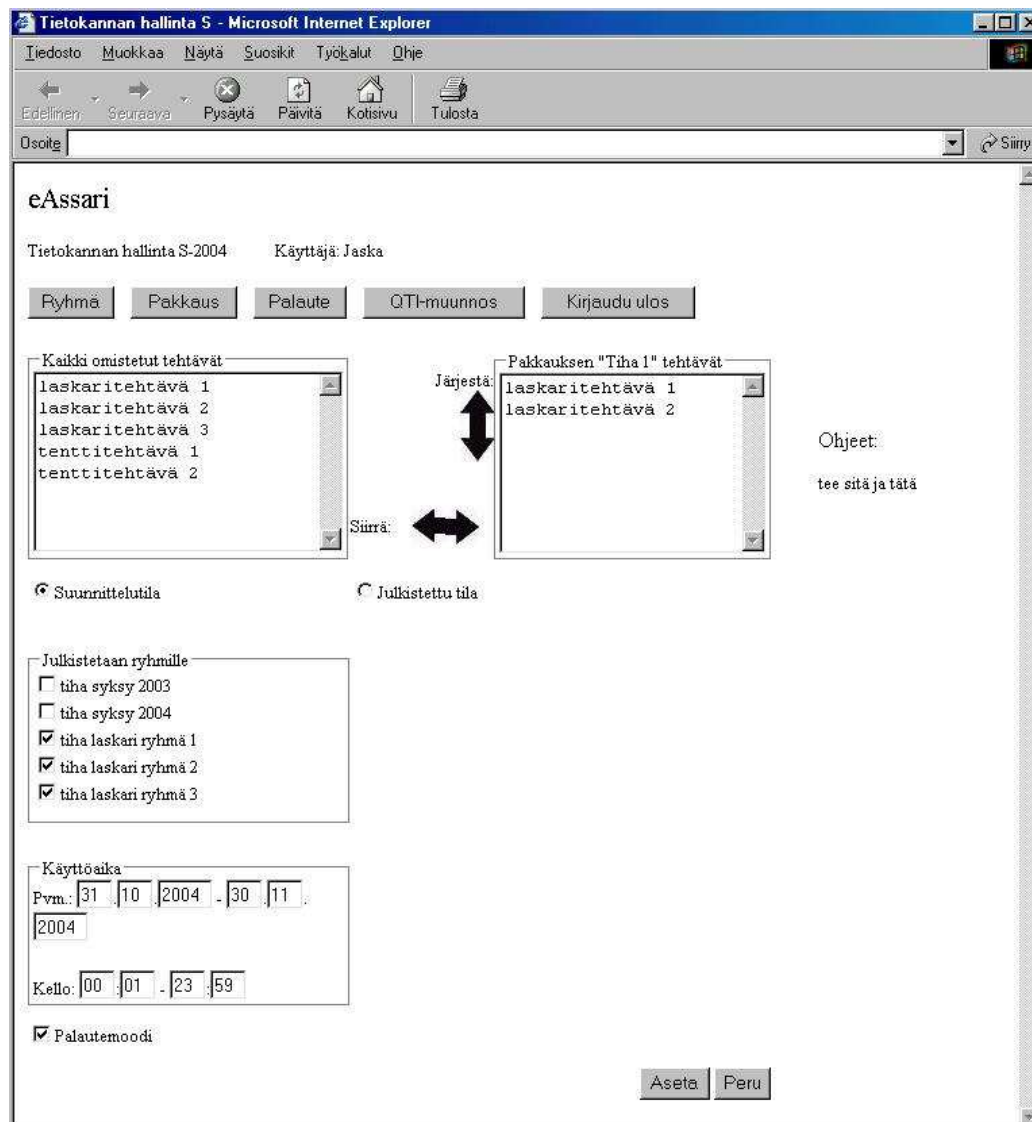
Oikeelliset tapaukset:

- Kaikki valitut tehtävät on muunnettu QTI-muotoon siten, että saman tehtävän erikielisiä versioita on muokattu kaksi erillistä QTI-tehtävää.

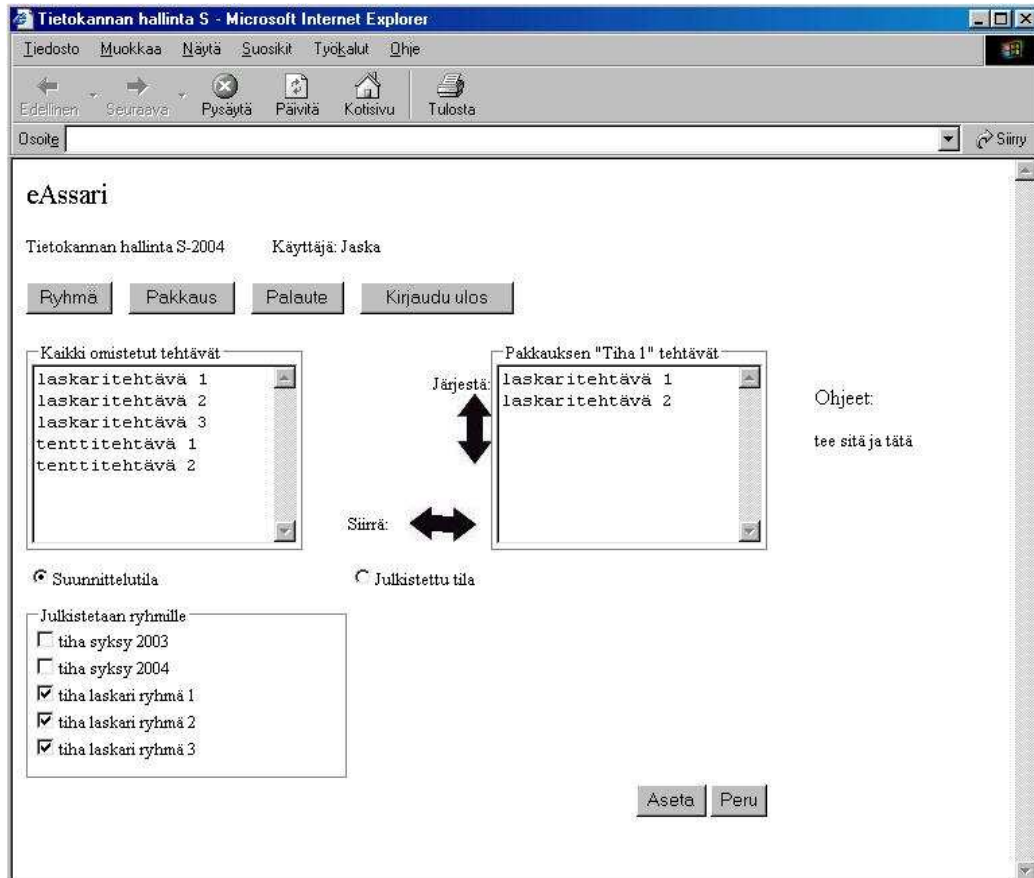
Virheelliset tapaukset:

- Osa tai kaikki valituista tehtävistä on jätetty muuntamatta QTI-muotoon.
- Saman tehtävän erikieliset versiot on muokattu yhdeksi QTI-tehtäväksi.

## Liite 3. Käyttöliittymäkuvia



Kuva 22: Opettajalle näytettävä näkymä



Kuva 23: Opiskelijalle näytettävä näkymä

## Liite 4. Palautelomake

**PALAUTELOMAKE**

Kokonaisuutena tehtävä oli	erittäin huono	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erittäin hyvä
Vaikeus	liian helppo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	liian vaikea
Opettavuus	olematon	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erittäin opettava
Oleellisuus	epäoleellinen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	keskeinen
Edellytetty perehtyneisyys	vähäinen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	syvälinen asiantuntemus
Tehtävän asettele	hyvin vaikeaselkoinen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erittäin selkeä
Saatu palaute	hyödytön	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erittäin hyödyllinen
Työlläisyys	hyvin vähän työtä	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erittäin työläs
Jos kyselyssä annettiin vastausvaihtoehtoja, oliko niiden valinta	huonosti onnistunut	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	erinomainen

Vapaaamuotoinen palaute: Kysytäänkö tehtävässä yksityiskohtia vai periaatteita? Mainitse mielestäsi tärkeimmät asiat, mitä tehtävällä halutaan opettaa. Muuta kommentoitavaa.

Parannusehdotuksia

Lähetä Tyhjennä

Kuva 24: Palautelomake