

Suunnitteludokumentti

Halaan-ryhmä

Helsinki 23.11.2006

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Paula Kemppi
Liisa Oikarinen
Ville Palkosaari
Maria Rinta-Opas
Jaakko Sorri
Marjaana Välisalo

Asiakas

Terttu Nevalainen

Johtoryhmä

Ilari Moilanen
Kimmo Simola

Kotisivu

<http://www.cs.helsinki.fi/group/ohtu/halaan>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
1.0	4.11.2006	Ensimmäinen versio
1.1	10.11.2006	Tarkastuksessa todetut puutteet ja virheet korjattu
1.2	13.11.2006	Luokkakuvaukset lisätty tähän versioon
1.3	20.11.2006	Päivitetty javadoc linkin päässä, liite palautettu

Sisältö

1	Johdanto	1
2	Sanasto	1
3	Järjestelmän yleiskuvaus	2
3.1	Arkkitehtuurin yleiskuvaus	2
3.2	Toteutus- ja toimintaympäristö	3
4	Tietokanta	3
4.1	Tietokannan rakenne	4
4.2	Tietokantataulut	4
4.2.1	Collection	5
4.2.2	Person	6
4.2.3	Letter	7
4.2.4	Text	9
4.2.5	CeecUser	9
4.2.6	CeecGroup	9
4.2.7	CeecQuery	10
5	Osajärjestelmät	10
5.1	Käyttöliittymä	10
5.2	Sovelluslogiikka	10
5.3	Tietokantapalikka	10
5.4	Tiedostopalikka	11
6	Komponentit	11
6.1	Käyttäjien hallinta	11
6.1.1	UserInformation.jsp	13
6.1.2	UpdateUser.jsp	13
6.1.3	AddUser.jsp	13
6.1.4	UpdateUserFunc.jsp	14
6.1.5	AddUserFunc.jsp	14
6.2	Tiedostojen hallinta	14

6.2.1	ExportImage.jsp	14
6.2.2	ExportPersondata.jsp	15
6.2.3	ExportLetterdata.jsp	15
6.2.4	ExportText.jsp	15
6.2.5	ImportData.jsp	16
6.2.6	ImportDataFunc.jsp	16
6.3	Korpustietojen hallinta	16
6.3.1	UpdateCorpusData.jsp	16
6.3.2	CheckNewCorpusData.jsp	18
6.3.3	AddCorpusData.jsp	18
6.3.4	RepairNewCorpusData.jsp	18
6.3.5	UpdateCorpusDataFunc.jsp	19
6.3.6	RepairNewCorpusDataFunc.jsp	19
6.3.7	AddCorpusDataFunc.java	19
6.4	Sisäänkirjautuminen	21
6.4.1	Login.jsp	21
6.4.2	LoginFunc.jsp	21
6.5	Kysely	21
6.5.1	Query.jsp	21
6.5.2	QueryFunc.jsp	23
6.6	Virhesivu	23
6.6.1	Error.jsp	23
6.7	Kaavio	23
6.7.1	Graph.jsp	25
6.7.2	GraphFunc.jsp	25
6.7.3	HrefLogic.jsp	25
6.7.4	ImageFunc.jsp	26
6.8	Kirjenäyttö	26
6.8.1	Letter.jsp	26
6.8.2	LetterFunc.jsp	26
6.9	Kirjelistanäyttö	28
6.9.1	LetterList.jsp	28
6.9.2	LetterListFunc.jsp	28

6.10	Henkilönäyttö	29
6.10.1	Person.jsp	29
6.10.2	PersonFunc.jsp	30
6.11	Henkilölistänäyttö	30
6.11.1	PersonList.jsp	31
6.11.2	PersonListFunc.jsp	31
6.12	Kokoelmanäyttö	31
6.12.1	Collection.jsp	31
6.12.2	CollectionFunc.jsp	33
6.13	Kokoelmalistanäyttö	33
6.13.1	CollectionList.jsp	33
6.13.2	CollectionListFunc.jsp	34
6.14	Uloskirjautuminen	34
6.14.1	LogoutFunc.jsp	34
6.15	Työkalupalkki	34
6.15.1	ToolBar.jsp	34
7	Luokat	35

Liitteet

1 Tietokannan luontitiedosto

1 Johdanto

Halaan-projektin tarkoituksena on kehittää hakukone laajalle annotoidulle tekstiaineistolle VARIENG-tutkimusyksikön käyttöön. Tutkijat ovat koonneet elektronisen tekstiaineiston eli korpuksen vanhoista englanninkielisistä kirjeistä. Kirjeiden lisäksi tutkimusyksiköllä on kerättyä muutakin kirjeisiin liittyvää dataa, kuten tietoja kirjeiden kirjoittajista ja vastaanottajista. Tutkijat tarvitsevat hakukoneen, joka mahdollistaisi sosiolingvistiset haut, eli kirjeiden ja niihin liittyvien tietojen hakemisen tietokannasta esimerkiksi kirjoittajan sukupuolen, asuinpaikan ja kirjoitusajankohdan perusteella.

Projektin tehtävänä on koota metadatan sisältävät tiedostot yhdeksi tietokannaksi ja tuottaa järjestelmä, joka toimii hakukoneena edellä mainitulle aineistolle. Järjestelmä tarjoaa myös muun muassa mahdollisuuden muokata hakutuloksen esitystä ja tallentaa haulla rajattu osa-aineisto tai vain siihen liittyvä metadata. Korpukseen voidaan lisätä uutta aineistoa ja olemassa olevaa voidaan päivittää. Koska järjestelmän käyttö edellyttää kirjautumista, käyttäjätietojen hallinnointi on myös osa järjestelmää.

Suunnitteludokumenttiin kootaan suunnitteluvaiheessa tehdyt ohjelman toteutusta koskevat päätökset. Dokumentti toimii projektiryhmän ohjeena toteutusvaiheen aikana ja tavoitteena on kuvata järjestelmä sellaisella tarkkuudella, että sen toteutus on suoraviivaista. Suunnittelun tuloksena saadaan kuvaukset toteutettavasta ohjelmistosta, sen arkkitehtuurista ja tietokannasta, järjestelmän tarvitsemista ja tuottamista tiedoista, ohjelman eri osien välisistä rajapinnoista sekä järjestelmän komponenteista.

Suunnittelussa otetaan huomioon vaatimusdokumentin luvuissa 4 ja 5 luetelluista vaatimuksista prioriteeteille A ja B sekä tärkeimmät prioriteetille C merkityt toiminnot. Viimeksi mainitun prioriteetin muiden toimintojen toteuttamisesta päätetään aikataulun sallissa toteutusvaiheen aikana.

2 Sanasto

Apache Tomcat Java Servlet-suoritusympäristö

Boolean-tyyppinen arvo Arvo, joka voi olla vain joko *tos*i tai *epätosi*.

Container-olio Olio, johon voidaan tallettaa tietoa.

CSS *Cascading Style Sheets* Tyyllisivut, jotka määrittelevät, miten HTML-elementit esitetään

Data Yhteydestä riippuen viittaa joko tekstidataan (kirjeiden sisältö) tai metadataan (data, johon haut kohdistuvat).

ER-malli *Entity-relationship* Tietokannan suunnittelussa käytettävä yksilö-yhteys -käsitelmä.

Hakulomake Hakusivu, josta käyttäjä voi valita hakuun vaikuttavat parametrit. Haku kohdistuu joko kirjeisiin tai kirjeiden kirjoittajiin tai vastaanottajiin.

Hakutulos Hakuehtoihin täsmävä *lista* joko kirjeistä tai niiden kirjoittajista tai vastaanottajista. Viitattaessa hakutuloksen tallennettavaan versioon pyritään puhumaan *taulukosta* tai CSV-tiedostosta. Hakutuloksesta voidaan tehdä myös *graafinen esitys*.

HTML *HyperText Markup Language* Sivunkuvauskieli

Java Oliopohjainen ohjelmointikieli

Javadoc Dokumentointityökalu

JavaScript Skriptikieli, jolla voidaan lisätä web-sivuille dynaamista toiminnallisuutta.

JDBC *Java Database Connectivity* Luokkakirjasto tietokantayhteyksien muodostamiseen

JSP *JavaServer Pages* Javapohjainen palvelintekniikka dynaamisten web-sivujen tuottamiseen.

JUnit Javapohjainen testikehys, jota voidaan käyttää testien ajamiseen ja joka mahdollistaa testien automatisoinnin.

Korpus Elektroninen tekstiaineisto.

Luokitteleva parametri Graafisen esityksen X-akselille sijoittuva parametri. Tämä parametri on oletuksena *Aika*, mutta se voi olla jokin muukin parametri, kunhan valitun parametrin arvot ovat koodi- tai boolean-tyyppisiä.

Metadata Kirjeteksteihin liittyvä tieto, joka kuvaa itse kirjettä ja sen kirjoittajaa tai vastaanottajaa. Data, jota parametrit kuvaavat ja johon haut kohdistuvat.

Sessio Järjestelmän käyttökerta. Palvelinpäässä ylläpidetään sessio-oliota, johon voidaan liittää erilaisia tietoja.

Web-sovellus Selaimen välityksellä käytettävä palvelinsovellus.

3 Järjestelmän yleiskuvaus

Tässä luvussa kuvataan järjestelmän arkkitehtuuri korkealla abstraktiotasolla sekä järjestelmän toteutus- ja toimintaympäristö.

3.1 Arkkitehtuurin yleiskuvaus

Järjestelmä on tietokantaa käyttävä web-sovellus ja se on periaatteessa mallinnettavissa kolmikerrosarkkitehtuurina (kuva 1). Alimpana on tiedonsaantikerros. Tästä erotetaan edelleen kaksi osajärjestelmää, joista toinen liittyy tietokannan ja toinen tiedoston käsittelyyn. Keskimmäisenä on sovelluslogiikkakerros, joka ohjaa järjestelmän toimintaa

ja tiedonvälitystä kerrokselta toiselle. Ylimmän kerroksen muodostaa JSP-sivuina toteutettava käyttöliittymä, joka ottaa syötettä käyttäjältä ja esittää esimerkiksi hakutulokset asiakkaalle HTML-muodossa. Jaon tarkoituksena on helpottaa ohjelman ylläpitoa ja tukea hyviä ohjelmointitapoja.



Kuva 1: Sovellusarkkitehtuuri

3.2 Toteutus- ja toimintaympäristö

Toteutettava järjestelmä suunnitellaan tietokantaa käyttäväksi web-sovellukseksi. Ohjelmiston pyörittämiseen tarvitaan siis tietokanta- ja www-palvelin ja sen käyttö edellyttää nykyaikaista, JavaScript- ja CSS-kykyistä selainta (Firefox 1.4 tai Internet Explorer 6.0 tai uudemmat). Järjestelmä toteutetaan ja testataan TKTL:n Linux-ympäristössä, jossa sovel-luspalvelimena toimii kone db.cs.helsinki.fi ja ajoympäristönä Apache Tomcat (4.1.18).

Tehokkaiden hakujen mahdollistamiseksi kirjekorpus ja Excel-muodossa oleva metada-ta mallinnetaan relaatiotietokantana. Toteutusvaiheessa järjestelmä käyttää TKTL:n tie-tokantapalvelimella bodbacka.cs.helsinki.fi olevaa Oracle-tietokantaa (versio 10g), mutta lopullinen sijoituspaikka lienee rinkka.helsinki.fi, jossa tarjolla on ainakin Oracle 8i.

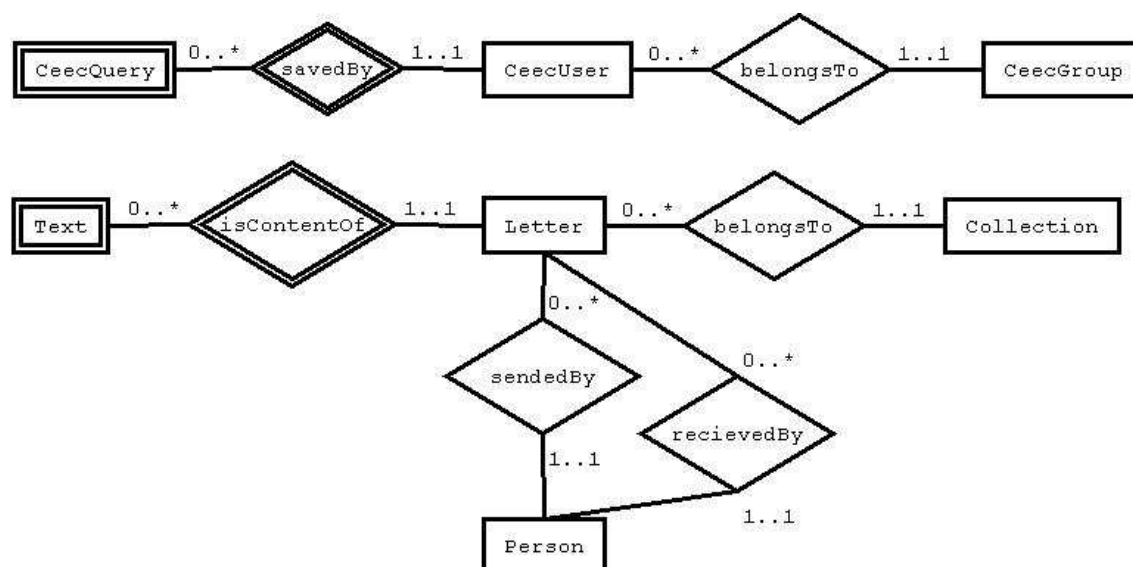
Järjestelmä toteutetaan Java-ohjelmointikielen versiolla 1.5., tietokantarajapintana toimii JDBC ja selainkäyttöliittymä tuotetaan JSP-sivuina. Koodi varustetaan suomenkielisin Javadoc-kommentein ja luokat testataan testaussuunnitelman mukaisesti JUnit:n avulla.

4 Tietokanta

Tässä luvussa esitellään suunnitelma tietokannan rakenteesta ja tietokantatauluista.

4.1 Tietokannan rakenne

Järjestelmän tietokantana käytetään relaatiotietokantaa. Tietokannan taulut ja niiden väliset suhteet esitetään kuvassa 2 yksilö-yhteys -mallin (ER-malli) avulla. Kaaviossa on esitetty vain yksilöjoukot (suorakulmiot) ja niiden väliset yhteydet eli suhteet (salmiakit). Tietokantaan sisältyy oleellisesti kaksi eri osaa, joista toisen lähtökohtana on asiakkaan kirjeaineisto ja siihen liittyvä metadata ja toinen liittyy käyttäjätietojen hallintaan. Ensin mainittu on siis attribuuttien ja niiden arvojoukkojen osalta pitkälti asiakkaan suunnittelema ja se muodostaa datan, johon haut kohdistuvat. Jälkimmäinen vastaa asiakkaan järjestelmään kohdistamiin vaatimuksiin ja mahdollistaa käyttäjien tunnistuksen, kyselyiden tallennuksen ja eri tasoisten oikeuksien määrittämisen käyttäjille käyttäjäryhmien avulla.



Kuva 2: Tietokantakaavio

4.2 Tietokantataulut

Tässä luvussa kuvataan järjestelmän tietokantataulut. Tauluja on yhteensä seitsemän ja jokainen niistä muodostaa oman alilukunsa, jonka otsikkona on taulun nimi. Mahdollista vapaamuotoista kuvausta seuraa taulukko, jossa on lueteltu kaikki taulun attribuutit (sulkeissa vanha nimi, mikäli nimi on oleellisesti muuttunut vaatimusdokumentissa käytettyä), näiden tietotyypit sekä annettu lyhyt kuvaus. Tyhjät arvot ovat sallittuja datatauluissa 4.2.1-4.2.4 lukuun ottamatta kaikkia pää- ja viiteavaimia. Käyttäjähallintaan liittyvissä tauluissa 4.2.5-4.2.7 *not null* -rajoitteet on erikseen mainittu ennen kuvausta. Null-rajoitteista seuraa, että esimerkiksi käyttäjän poistaminen poistaa myös kaikki käyttäjän tallettamat haut ja henkilön poistaminen poistaa kaikki henkilön kirjoittamat tai vastaanottamat kirjeet.

Kunkin rivin yksilöivät attribuutit eli pääavaimet on alleviivattu. Merkintä =taulu:attr tarkoittaa viiteavainta ja käytännössä eheystarkistusta mainitun taulun attribuutin kanssa. Tällainen kenttä voi saada vain arvoja, jotka esiintyvät viitatus taulun mainitussa kentässä. Jos kuvauksen perässä on numero, se tarkoittaa, että kyseisen kentän arvoalue on rajattu. Mahdolliset koodit ja koodiyhdistelmät luetellaan taulun jälkeen tulevassa listassa. Mikäli tyyppinä on *char(1)* eikä erillisiä koodeja ole lueteltu, kenttä viittaa boolean-tyyppiseen muuttujaan ja arvona on tällöin *y* tai *n*. Sisällöllisiä rajoituksia ei tehdä tietokantaan, mutta esimerkiksi koodit tarkistetaan parserissa ennen kuin päivitystiedosto viedään kantaan. Indeksit luodaan ainakin pääavaimille ja keskeisimmille viiteavaimille ja tarpeen mukaan muille kentille.

4.2.1 Collection

Kokoelmaan liittyy varsin vähän omia ominaisuuksia. Kokoelman nimi toimii pääavaimena. Kokoelmaan liittyy kuitenkin useampia arvoltaan johdettavia parametreja, joita asiakas haluaa nähdä ja käyttää.

<u>Name</u>	varchar(20)	nimi
Filename	varchar(8)	tiedostonimi
FromYear	number(4)	vuodesta
ToYear	number(4)	vuoteen

4.2.2 Person

<u>PersonCode</u> (correspondentcode)	varchar(20)	yksilökoodi
Sex	char(1)	sukupuoli (1)
Region (dom reg)	char(1)	asuinalue (2)
County (dom cnt)	varchar(5)	asuinkunta (3)
SocMob	char(1)	sos. liikkuvuus (4)
LastName (lname)	varchar(255)	sukunimi
FirstName (fname)	varchar(255)	etunimi
TitleOccupation (title)	varchar(255)	titteli
YBirth	number(4)	syntymävuosi
YDeath	number(4)	kuolinvuosi
Lifespan (=uusi)	varchar(255)	elinaika epävarmuusilmauksin
Rank (rankst)	varchar(3)	korkein sos. status (5)
FatherRank (frank)	varchar(3)	isän sos. status (5)
Father	varchar(1000)	isän tiedot
PBirth	char(1)	syntymäpaikka (2)
MigCode (mig)	varchar(3)	muuttohistoria (6)
Migration	varchar(1500)	muuttohistoria
EduCode (educ)	varchar(4)	koulutus (7)
Education	varchar(1500)	koulutus
Career	varchar(1500)	ura
Religion (rel)	char(1)	uskonto (8)
DNB	varchar(50)	linkki ODNB Onlineen
Notes	varchar(1500)	lisätietoa
SentLettcont	varchar(10)	kirj. kirjeiden sis.tyypit (9)
RecLettcont	varchar(10)	v-o. kirjeiden sis.tyypit (9)
Complete	char(1)	tiedot valmiit
Updated	date	viimeisin päivitys
NewBoolean1	char(1)	-
NewBoolean2	char(1)	-
NewText1	varchar(50)	-
NewText2	varchar(255)	-
NewNumber	number(5)	-

Käytetyt koodit:

1. Sex: F (female), M (male)
2. Region / PBirth: N (north), F (east anglia), H (home counties), L (london), C (court), O (other), A (abroad)
3. County: avainlistan kolmekirjaiminen lyhenne (lis. myöh) (+A, C, F, H, L, N, O) (*kentän sisältöä ei tarkisteta*)
4. SocMob: U (up), D (down), N (none)

5. Rank / FatherRank: R (royalty), N (nobility), GU (gentry upper), GL (gentry lower), G (gentry), P (professional), CU (clergy upper), CL (clergy lower), M (merchant), O (other), (+?)
6. MigCode: Y (yes), YL (yes: london), YA (yes: abroad), YLA (yes: london & abroad)
7. EduCode: A (apprenticed), E (elementary), H (higher), HC (higher: cambridge), HI (higher: inns of court), HO (higher: oxford), PC (private/self: classical), PN (private/self: non-classical), S (secondary), HF (higher: foreign), (+ C, O, I, F) (+?)
8. Religion: P (protestant), A (anglican), C (catholic), X (unknown)
9. SentLettcont / RecLettcont: M (mixed), B (business), P (private), N (news), O (official), W (other), L (love), D (duty), T (travel), F (family) - mikä tahansa yhdistelmä näistä (*kentän sisältöä ei tarkisteta*)

4.2.3 Letter

Kirjetaulun attribuutit kertovat joko kirjeistä tai niiden kirjoittajista ja henkilöistä sellaista tietoa, joka voi vaihdella kirjeestä toiseen. Jokaisen kirjeen yksilöi erillinen tunniste. Kirje kuuluu aina yhteen kokoelmaan ja siihen liittyy kaksi henkilöä eli kirjoittaja ja vastaanottaja.

<u>LetterID</u>	varchar(15)	tunniste
Collection	=Collection:Name	kokoelma
Sender	=Person:PersonCode	kirjoittaja
SenderRank (rank)	varchar(3)	kirj. sos. status (1)
SenderStatus (status)	varchar(255)	kirj. sos. status
MultiSenders (=uusi)	char(1)	useita kirjoittajia
Recipient	=Person:PersonCode	vastaanottaja
RecRank (rank)	varchar(3)	vo. sos. status (1)
RecStatus (status)	varchar(255)	vo. sos. status
MultiRec (=uusi)	char(1)	useita vastaanottajia
Year	number(4)	kirjoitusvuosi
YearUncertain (=uusi)	char(1)	vuosi epävarma
WordCount	number(5)	sanamäärä
RelCode (relationship)	varchar(2)	kirj-vo -suhde (2)
Relationship (actual relationship)	varchar(255)	kirj-vo -suhde
Place	varchar(255)	kirjoituspaikka
Authenticity	varchar(3)	kirjeen autenttisuus (3)
LetterDate (date)	varchar(255)	aika tarkemmin vapaamuotoisesti
AddressFormula	char(1)	aloituskaava
ClosingFormula	char(1)	lopetuskaava
ContentType	varchar(255)	sisältötyyppi
LetterNotes (notes)	varchar(1000)	muuta kirjeestä
CorrespondentNotes (=uusi)	varchar(1000)	muuta henkilöistä
LetterNumber	varchar(50)	numero editiossa
PageNumber	varchar(50)	sivunumero editiossa
Source	varchar(500)	lähde
Copyright	char(1)	julkaisuoikeus
Complete	char(1)	tiedot valmiit
Updated	date	viimeisin päivitys
NewBoolean1	char(1)	-
NewBoolean2	char(1)	-
NewText1	varchar(50)	-
NewText2	varchar(255)	-
NewNumber	number(5)	-

Käytetyt koodit:

1. SenderRank / RecRank: samat kuin Person-taulun Rank ja FatherRank -kentillä
2. RelCode: FN (family nuclear), FO (family other), FS (family servant), TC (close friend), T (other)
3. Authenticity: A (holograph), B (holograph; writer's social background partly unknown), C (later copy), D (uncertain authenticity; copy & writer's social bg partly unknown), E (modernized), S (scribal/secretarial) tai 2 näistä, esim. AC, CA (+?)

4.2.4 Text

Yhdestä kirjeestä viedään tietokantaan 1-3 versiota.

<u>Letter</u>	Letter:LetterID	kirje
<u>Version</u>	char(1)	korpusversio (1)
Text	clob	tekstisisältö

Käytetyt koodit:

1. Version: P (plain), T (pos-tagged), S (parsed)

4.2.5 CeecUser

Pääkäyttäjän oikeuksilla voi hallinnoida käyttäjiä. Käyttäjä rekisteröidään kantaan yksilöllivällä käyttäjätunnuksella ja lisäksi vaaditaan salasana. Käyttäjä liitetään yhteen käyttäjäryhmään, jonka kautta hänelle määräytyy oikeudet ja rajoitukset hakukoneen käyttöön liittyen.

<u>UserID</u>	varchar(8)	not null	käyttäjätunnus
Password	varchar(10)	not null	salasana
UserGroup	CeecGroup:Name	not null	käyttäjärühmä
FirstName	varchar(50)	not null	etunimi
LastName	varchar(50)	not null	sukunimi
Email	varchar(50)		sähköposti
Phone	varchar(50)		puhelin

4.2.6 CeecGroup

Ryhmä määrää siihen kuuluvien käyttäjien oikeudet ja rajoitukset.

<u>Name</u>	varchar(15)	not null	ryhmän nimi
Description	varchar(50)		lyhyt kuvaus
UserAdmin	char(1)	not null	oikeus: käyttäjätietojen hallinnointi
CorpusAdmin	char(1)	not null	oikeus: tietokannan tietojen päivitys
QuerySave	char(1)	not null	oikeus: hakuparametrin tallennus kantaan
CorpusExport	char(1)	not null	oikeus: osakorpuksen tallennus itselle
MetadataExport	char(1)	not null	oikeus: metadatan ja kuvan tallennus itselle
Copyright	char(1)	not null	aineistorajoite: julkaisu-oikeus

Käyttäjärühmät luodaan kantaan valmiiksi yhteistyössä asiakkaan kanssa. Suunnitelma hierarkkisiksi ryhmiksi:

Admin Kaikki oikeudet, ei aineistorajoitetta.

CEEC Kuten Admin, mutta ei oikeutta käyttäjätietojen hallinnointiin.

Researcher Kuten CEEC, mutta ei oikeutta korpustietokannan päivityksiin.

ResearcherLite Kuten Researcher, mutta aineistorajoite voimassa.

Guest Kuten ResearcherLite, mutta ei oikeutta tallentaa hakuparametreja tietokantaan.

GuestLite Kuten Guest, mutta ei oikeutta osakorpuksen ja metadatan tallentamiseen itselle.

4.2.7 CeecQuery

<u>Time</u>	date	not null	aikaleima
<u>Owner</u>	CeecUser:UserID	not null	tallentaja
Name	varchar(50)		nimi
Query	varchar(255)	not null	kysely

5 Osajärjestelmät

Tässä luvussa kuvataan lyhyesti järjestelmästä erotetut osajärjestelmät ja niiden tehtävät sekä osajärjestelmien välinen yhteistyö. Osajärjestelmien rajapinnat kuvataan vain yleisellä tasolla, formaalimpi määrittely on komponenttien yhteydessä luvussa 6.

5.1 Käyttöliittymä

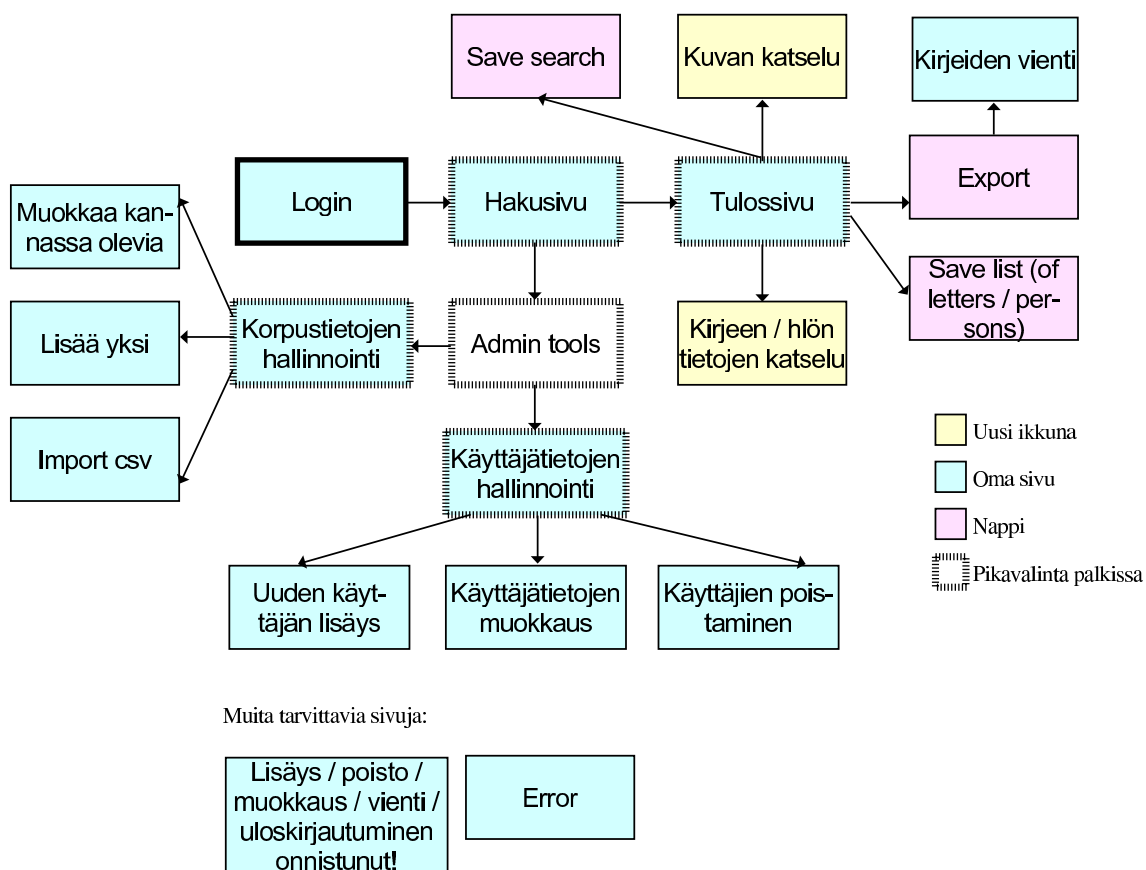
Käyttöliittymä tarjoaa käyttäjälle mahdollisuuden käyttää järjestelmän tarjoamia palveluita. Kuvassa 3 on esitelty käyttöliittymän eri sivut ja niiden väliset suhteet. Lomakkeiden parametreihin päästään käsiksi JSP-sivuilla request-olion avulla. Käyttöliittymään tulee pääsääntöisesti container-olioita sovelluslogiikan suunnasta. Käyttöliittymän osia on kuvattu tarkemmin luvussa 6.

5.2 Sovelluslogiikka

Sovelluslogiikkakerros ohjaa järjestelmän toimintaa ja tiedonvälitystä kerrokselta toiselle. Se muokkaa käyttöliittymäkerrokselta saamansa tiedot tietokanta- ja tiedostopalikan tarvitsemaan muotoon. Samoin se muokkaa esimerkiksi tietokannasta tulevan datan käyttöliittymän tarvitsemaan muotoon. Sovelluslogiikka välittää erilaisia container-olioita käyttöliittymän ja tiedonsaantilogiikan välillä.

5.3 Tietokantapalikka

Tietokantaa käytetään JDBC-rajapinnan kautta. Loogisen tietokantapalikan tehtävänä on esimerkiksi hakujen tapauksessa kyselyiden muodostaminen haluttujen parametrien perusteella ja tuloksen sisällyttäminen container-olioon, joka välitetään edelleen sovellus-



Kuva 3: Käyttöliittymän eri sivut ja niiden väliset suhteet.

logiikkakerrokselle. Tietokantapalikka voi saada myös valmiin container-olion vietäväksi tietokantaan.

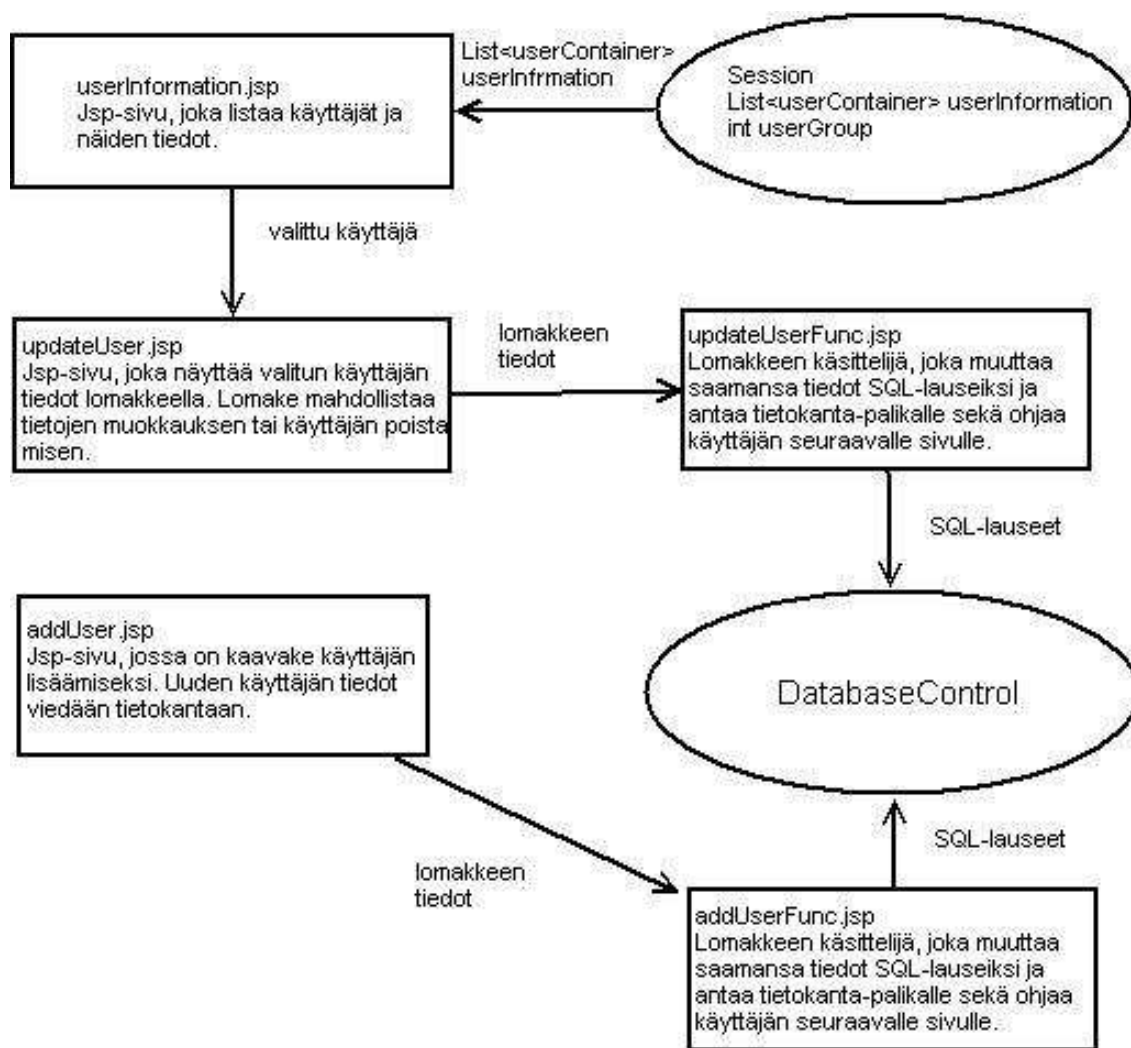
5.4 Tiedostopalikka

Järjestelmässä käyttäjällä on mahdollisuus päivittää ja muokata tietokantaa csv-muotoisen tiedoston avulla ja tallentaa esimerkiksi hakutuloksia ja osakorpuksia itselleen. Tiedostopalikka parseroi tiedoston sisällön ja palauttaa container-olion, jonka sisältö riippuu tiedoston sisällöstä. Jos käyttäjä haluaa tallentaa hakutuloksen itselleen, tulosta vastaavasta container-oliolistasta muodostetaan tiedosto, joka annetaan käyttäjän tallennettavaksi.

6 Komponentit

6.1 Käyttäjien hallinta

Komponentin avulla pidetään yllä käyttäjätietoja (kuva 4). Mahdolliset ylläpitotoimenpiteet ovat uuden käyttäjän lisäys, käyttäjän poistaminen ja käyttäjän tietojen muuttaminen.



Kuva 4: Käyttäjien hallinta

Kaikki komponentin sivut vaativat, että käyttäjällä on UserAdmin -oikeudet.

6.1.1 UserInformation.jsp

Palvelu: Sivulla saa Sessiosta käyttäjätiedot, jotka se tulostaa listaksi. Listasta voi valita yksittäisen käyttäjän muokattavaksi tai poistettavaksi kohdan 6.1.2 lomakkeella.

Parametrit: List<UserContainer>

Tuotokset: Jsp-sivu

Poikkeustilanteet: -

Oikeudet: UserAdmin

6.1.2 UpdateUser.jsp

Palvelu: Sivulla on yksittäisen käyttäjän tiedot lomakkeella, jossa tietojen muokkaus on mahdollista. Lomakkeen arvot tarkistetaan javascriptillä. Myös käyttäjän poistaminen on mahdollista.

Parametrit: UserContainer

Tuotokset: Usp-sivu

Poikkeustilanteet: Javascript ei hyväksy käyttäjän antamia arvoja

Oikeudet: UserAdmin

6.1.3 AddUser.jsp

Palvelu: Sivulla on lomake, jolla voi lisätä käyttäjän. Käyttäjän syöttämät arvot tarkastetaan javascriptillä.

Parametrit: -

Tuotokset: Jsp-sivu

Poikkeustilanteet: Javascript ei hyväksy käyttäjän antamia arvoja

Oikeudet: UserAdmin

6.1.4 UpdateUserFunc.jsp

Palvelu: Lomakkeenkäsittelijä, joka muodostaa saamastaan lomakkeesta UserContainer-olion välitettäväksi eteen päin. Mikäli kyseessä on poisto, poistetaan tietokannasta myös käyttäjän tallentamat kyselyt. Tietokantaoperaatiot suoritetaan Interface-luokan updateUser- ja removeUser-metodeilla.

Parametrit: Lomake (vastaavat tiedot kuin: UserContainer)

Tuotokset: UserContainer

Poikkeustilanteet: Operaatioita ei saada tehtyä tietokantaan

Oikeudet: UserAdmin

6.1.5 AddUserFunc.jsp

Palvelu: Lomakkeenkäsittelijä, joka muodostaa saamastaan lomakkeesta UserContainer-olion välitettäväksi eteenpäin.

Parametrit: lomake (vastaavat tiedot kuin: UserContainer)

Tuotokset: UserContainer

Poikkeustilanteet: Operaatioita ei saada tehtyä tietokantaan

Oikeudet: UserAdmin

6.2 Tiedostojen hallinta

Komponentti käsittelee järjestelmän tarvitsemat tiedostojen siirrot asiakkaan ja palvelimen välillä. Asiakaskoneelle tallennetaan kuvia (kaavioita hakutuloksesta), metadattaa (hakutuloksena saatuja kirje- ja henkilölistoja) ja osakorpuksia eli kirjetekstit sisältäviä tekstitiedostoja. Asiakaskoneelta ladataan järjestelmään korpuksen päivitystiedostoja.

6.2.1 ExportImage.jsp

Palvelu: Sivu saa Sessiosta BufferedImage-olion, josta se muodostaa jpeg-kuvan. Kuva tallennetaan asiakkaan koneelle. Sivulle tullaan vain tiedostoon image.jpg kohdistuvalla pyynnöllä, joka ohjataan tälle sivulle.

Parametrit: BufferedImage

Tuotokset: Jpeg-kuva (asiakkaan tallennettavissa)

Poikkeustilanteet: Tiedoston luonti epäonnistuu

Oikeudet: MetadataExport

6.2.2 ExportPersondata.jsp

Palvelu: Sivu saa Sessiosta List<PersonContainer> -olion, josta se muodostaa asiakas-koneelle tallennettavan tekstitiedoston. Sivulle tullaan vain tiedostoon person.csv kohdistuvalla pyynnöllä, joka ohjataan tälle sivulle.

Parametrit: List<PersonContainer>

Tuotokset: Tekstitiedosto (asiakkaan tallennettavissa)

Poikkeustilanteet: Tiedoston luonti epäonnistuu

Oikeudet: MetadataExport

6.2.3 ExportLetterdata.jsp

Palvelu: Sivu saa Sessiosta List<LetterContainer> -olion, josta se muodostaa asiakas-koneelle tallennettavan tekstitiedoston. Sivulle tullaan vain tiedostoon letter.csv kohdistuvalla pyynnöllä, joka ohjataan tälle sivulle.

Parametrit: List<LetterContainer>

Tuotokset: Tekstitiedosto (asiakkaan tallennettavissa)

Poikkeustilanteet: Tiedoston luonti epäonnistuu

Oikeudet: MetadataExport

6.2.4 ExportText.jsp

Palvelu: Sivu saa Sessiosta List<TextContainer> -olion, josta se muodostaa asiakas-koneelle tallennettavan tekstitiedoston. Sivulle tullaan vain tiedostoon text.txt kohdistuvalla pyynnöllä, joka ohjataan tälle sivulle.

Parametrit: List<TextContainer>

Tuotokset: Tiedosto (asiakkaan tallennettavissa)

Poikkeustilanteet: Tiedoston luonti epäonnistuu

Oikeudet: MetadataExport

6.2.5 ImportData.jsp

Palvelu: Sivun muodostaa lomakkeen, jonka kautta käyttäjän on mahdollista ladata järjestelmään päivitykset sisältävä tiedosto. Asiakkaan kanssa sovitaan tiedostonimet ja -pääteet jotka kertovat tiedoston tyyppin (kokoelma, henkilö, kirje, kirjeteksti).

Parametrit: -

Tuotokset: Tiedosto

Poikkeustilanteet: -

Oikeudet: UserAdmin

6.2.6 ImportDataFunc.jsp

Palvelu: Lomakkeenkäsittelijä, joka ottaa vastaan datan ImportData.jsp-sivun lomakkeelta. Sivun käyttää datan ensin FileStreamAccessorilla, joka poimii siitä tiedostotyyppin ja käyttäjän lataaman tiedoston. FileStreamAccessor palauttaa UpdateCorpusContainer-olion, jonka tämä func-sivu ohjaa edelleen tiedostoparserille eli TextParser-luokalle. Jos parseri hyväksyy datan, ohjataan käyttäjä kohdan 6.3.2 sivulle, jolle välitetään piilokenttänä päivityksen tyyppi ja parserilta saatu lista talletetaan Sessioon List<Container> -oliona nimellä *add*. Jos tiedostossa on jotain ongelmaa ja parserointi ei onnistu, käyttäjälle ilmoitetaan virheestä.

Parametrit: Tiedosto

Tuotokset: List<Container>

Poikkeustilanteet: Parseri hylkää tiedoston

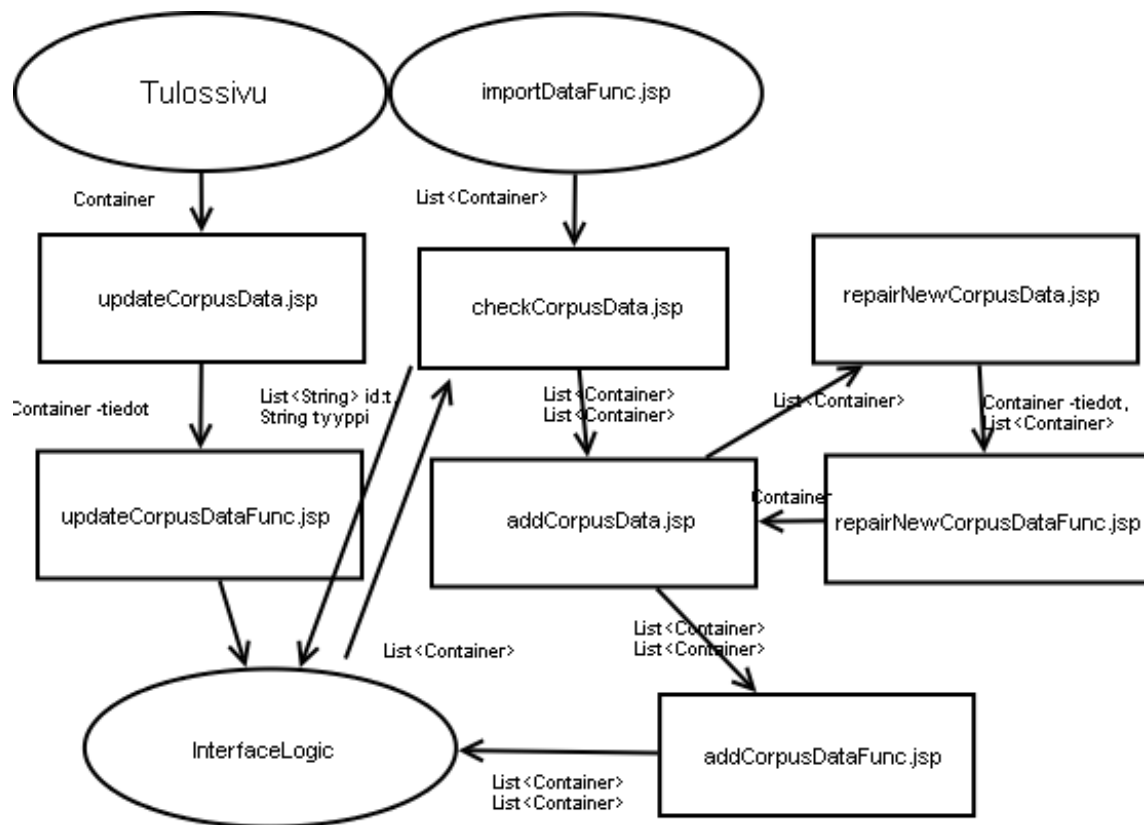
Oikeudet: UserAdmin

6.3 Korpustietojen hallinta

Komponentti tarjoaa mahdollisuuden päivittää korpusta (kuva 5). Tarjotut palvelut ovat tietokannan kirje-, henkilö- ja kokoelmatauluihin kohdistuvat muutos, lisäys ja poisto.

6.3.1 UpdateCorpusData.jsp

Palvelu: Sivun näyttää valitun objektin lomakkeella, jolla tietojen muokkaus ja poistaminen on mahdollista. Myös mahdollisen poiston yhteydessä poistuvat objektit näytetään (katso kohta 6.3.5). Sivulle tullaan tulospäätös sivun kautta, josta välitetään parametrina muokattavan listan tyyppi ja muokattavan rivin indeksi. Lomakkeen kenttien arvot tarkastetaan javascriptillä.



Kuva 5: Korpustietojen hallinta

Parametrit: Container (singleUpdate), Containerin tyyppi (collection, person, letter, text)

Tuotokset: Jsp-sivu, jolla lomake

Poikkeustilanteet: Javascript ei hyväksy syötettyä arvoa

Oikeudet: CorpusAdmin

6.3.2 CheckNewCorpusData.jsp

Palvelu: Sivulle tullaan kohdan 6.2.6 sivun kautta. Sivun ottaa Sessiosta (järjestelmään ladatusta tiedostosta saadun) *add*-nimisen List<Container>-olion ja tarkistaa tietokantahaulla, onko riveistä olemassa aiemmat versiot. Duplikaatit siirretään *add*-listasta *update*-listaan. Käyttäjä ohjataan kohdan 6.3.3 sivulle.

Parametrit: List<Container>, Containerin tyyppi (collection, person, letter, text)

Tuotokset: List<Container> (update)

Poikkeustilanteet: Olemassaolon tarkistus ei onnistu.

Oikeudet: CorpusAdmin

6.3.3 AddCorpusData.jsp

Palvelu: Sivun näyttää tiedostosta saadun datan ennen sen viemistä tietokantaan. Uudet rivit ja päivitettävät rivit näytetään erikseen. Sivulta voi siirtyä kohdan 6.3.4 sivulle, jossa voi korjata yksittäisiä päivitysrivejä. Text-päivityksiä tehtäessä ei kirjeiden sisältöjä näytetä käyttäjälle.

Parametrit: List<Container> (add), List<Container> (update), Containerin tyyppi (collection, person, letter, text)

Tuotokset: Jsp-sivu, jossa näytetään tiedostosta saatu data

Poikkeustilanteet: -

Oikeudet: CorpusAdmin

6.3.4 RepairNewCorpusData.jsp

Palvelu: Tekee lomakkeen, joka mahdollistaa tiedostosta ladatun datan muokkaamisen rivi kerrallaan. Käyttäjän syötteet tarkastetaan javascriptilla.

Parametrit: Add-listassa olevan Container-olion tunnus

Tuotokset: Jsp-sivu, jolla lomake tietokantaan vietävien rivien muokkamiseksi

Poikkeustilanteet: Javascript ei hyväksy lomakkeen kenttää

Oikeudet: CorpusAdmin

6.3.5 UpdateCorpusDataFunc.jsp

Palvelu: Lomakkeenkäsittelijä, joka vie muutokset ja poistot tietokantaan kutsumalla InterfaceLogicin update- ja remove-metodeja. Poistettaessa henkilöä myös henkilön kirjoittamat kirjeet poistetaan. Poistettaessa kokoelmaa myös kokoelman kirjeet poistetaan. Poistettaessa kirjeitä myös kirjeiden tekstit poistetaan.

Parametrit: Lomakkeen tiedot (vastaa Container)

Tuotokset: -

Poikkeustilanteet: Tietokantaoperaatioiden suoritus ei onnistu

Oikeudet: CorpusAdmin

6.3.6 RepairNewCorpusDataFunc.jsp

Palvelu: Lomakkeenkäsittelijä, joka tallentaa kohdan 6.3.4 lomakkeen tiedot Sessiossa olevaan List<Container>:iin ja ohjaa käyttäjän takaisin sivulle 6.3.3.

Parametrit: Lomakkeen tiedot (vastaa Container), Containerin tyyppi (piilokentässä), listan nimi

Tuotokset: Container

Poikkeustilanteet: -

Oikeudet: CorpusAdmin

6.3.7 AddCorpusDataFunc.java

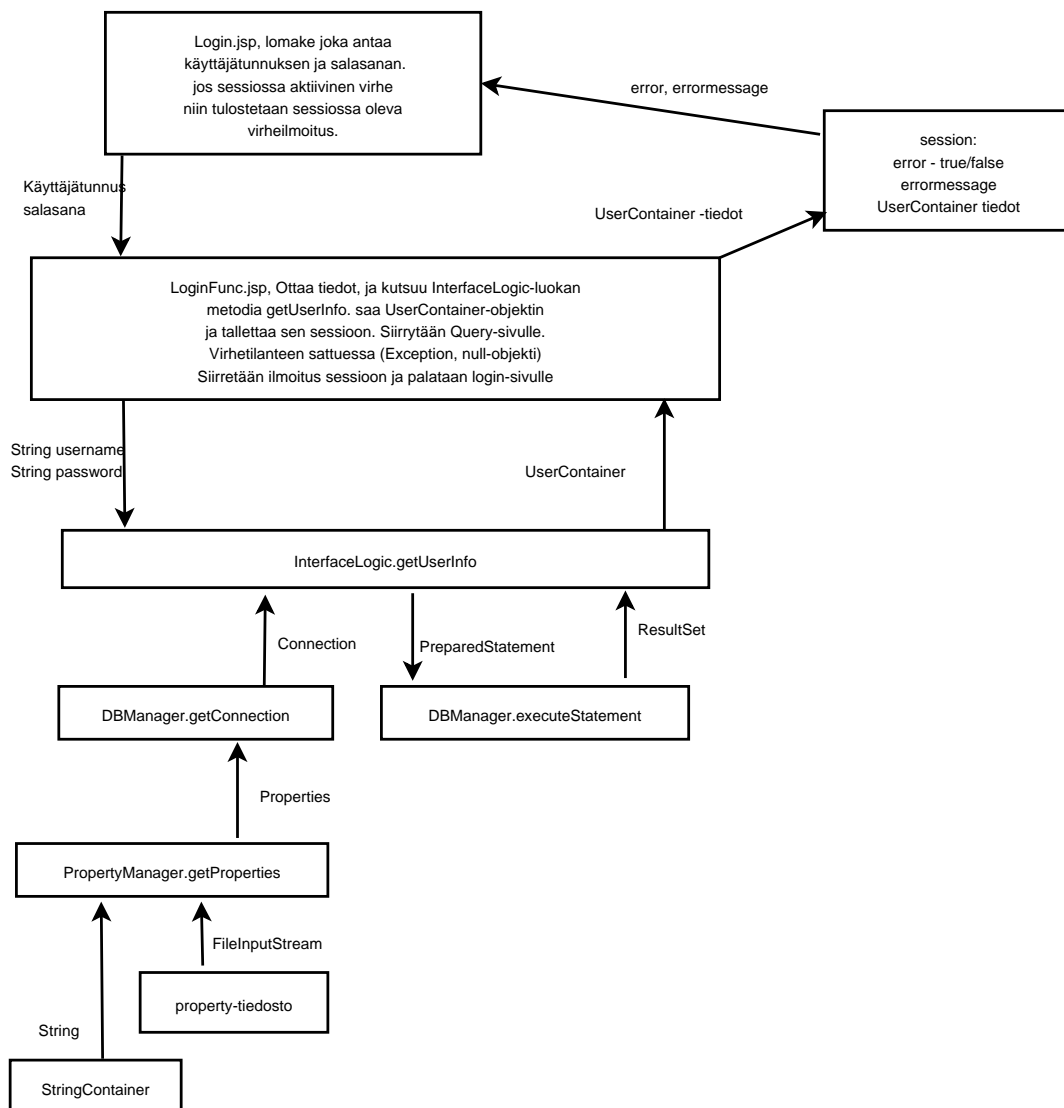
Palvelu: Vie päivitykset tietokantaan välittämällä Sessiosta add-listan InterfaceLogic-luokan add-metodille ja update-listan update-metodille.

Parametrit: List<Container> (add), List<Container> (update)

Tuotokset: -

Poikkeustilanteet:

1. Yritetään lisätä sellaista kirjettä, jonka kirjoittaja, vastaanottaja tai kokoelma ei ole olemassa
2. Yritetään lisätä sellaista tekstiä, jota vastaavaa kirjettä ei ole tietokannassa



Kuva 6: Sisäänkirjautuminen

6.4 Sisäänkirjautuminen

Komponentti hoitaa järjestelmään kirjautumisen ja käyttäjän tietojen tallentamisen Sessioon muiden komponenttien käyttöä varten.

6.4.1 Login.jsp

Palvelu: Sivua vastaanottaa Sessiossa jo olevan mahdollisen virhetilanteen ja tulostaa sen. Sen lisäksi sivulla on lomake, johon syötetään tunnus ja salasana, sekä linkki ylläpidon sähköpostiin.

Parametrit: Sessiosta saatu virhetilanne

Tuotokset: Käyttäjän syöttämät käyttäjätunnus ja salasana

Poikkeustilanteet: Yunnus/salasanaparia ei ole

Oikeudet: -

6.4.2 LoginFunc.jsp

Palvelu: Vastaanottaa login.jsp:n lomakkeen tiedot, ottaa käyttäjätunnuksen ja salasanan. Kutsuu niillä getUserInfo:a. Jos kirjautuminen onnistui, vastaanotetaan UserContainer. Talletetaan sen tiedot sessioon. Ohjataan query.jsp-sivulle, muuten palataan login.jsp-sivulle ja talletetaan virhetilanne sessioon.

Parametrit: Nimi, salasana

Tuotokset: UserContainer

Poikkeustilanteet: Käyttäjätunnus/salasanaparia ei löydy, tietokannan lukeminen ei onnistu

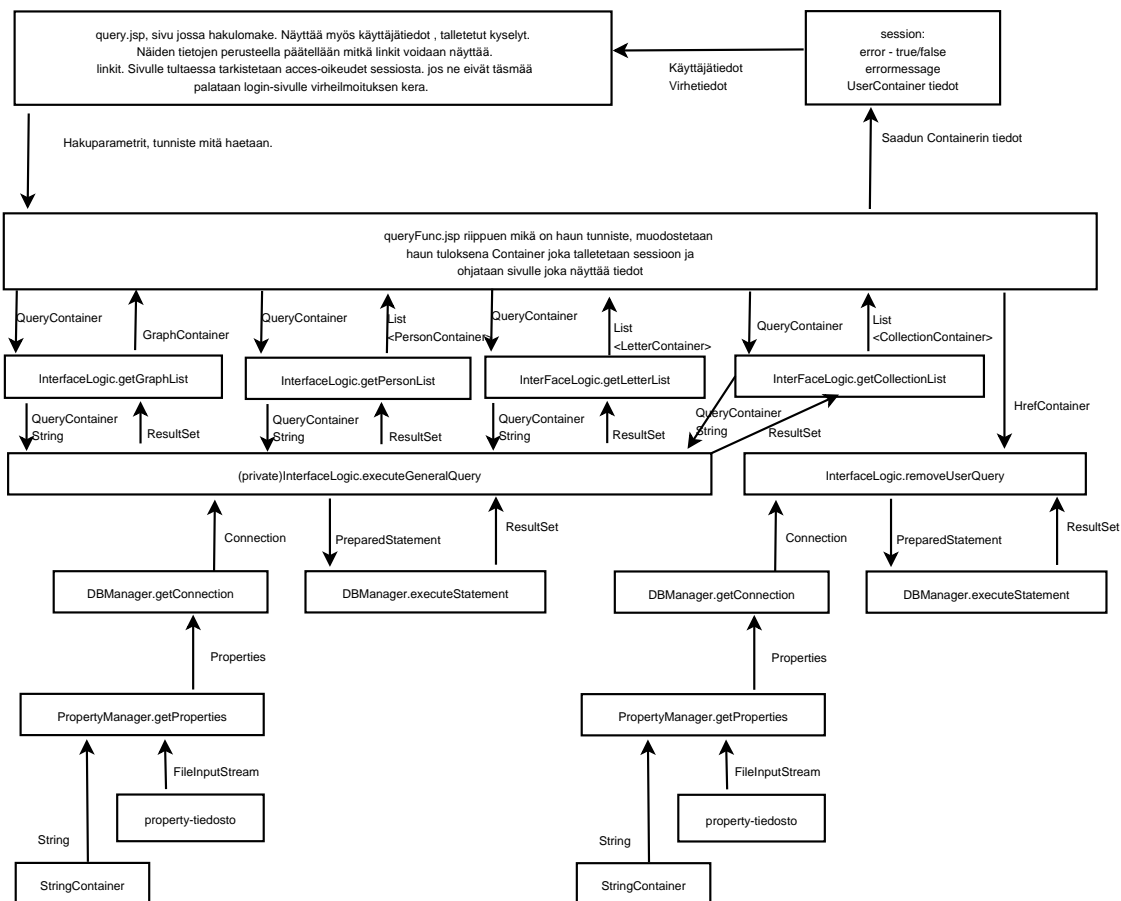
Oikeudet: -

6.5 Kysely

Tämä osa sisältää komponentit jotka hoitavat kyselyn tekemisen, tiedon sisältävien Containerien muodostomisen sekä niiden välittämisen näytösivuille.

6.5.1 Query.jsp

Palvelu: Sivulla on hakulomake, johon syötettyjen tietojen perusteella tehdään haut korpuksesta. Mahdollisuus käyttää käyttäjän talletettuja hakuja. Käyttäjän oikeuksien mukaan linkit muihin järjestelmän osiin.



Kuva 7: Kyselylomake

Parametrit: Sessiosta saatu virhetilanne, käyttäjätiedot

Tuotokset: Hakuparametrit

Poikkeustilanteet: Sivulle tultaessa ei tarvittavia oikeuksia / käyttäjätietoja.

Oikeudet: Käyttäjän pitää olla sessiossa

6.5.2 QueryFunc.jsp

Palvelu: Muodostaa query.jsp:ltä saaduista tiedoista sekä sessiosta saatavilla käyttäjätiedoilla QueryContainerin, jonka perusteella suoritetaan tietokantahaku. Tästä vastauksena palautuu Container tai lista Containereita, jotka välitetään tulossivuille ja näytetään käyttäjälle.

Parametrit: Query.jsp:stä saatu syöte, käyttäjätiedot sessiosta

Tuotokset: Haun tulokset sisältävä Container tai lista Containereita.

Poikkeustilanteet: Tiedoston tai tietokannan lukeminen ei onnistu

Oikeudet: Käyttäjän pitää olla sessiossa

6.6 Virhesivu

Tämä komponentti näyttää ne virheet, joiden tapahtuessa järjestelmä ei voi jatkaa toimintaansa normaalisti.

6.6.1 Error.jsp

Palvelu: Muodostaa virheilmoituksen sessiosta saaduilta virhetiedoilla. Tälle sivulle ohjataan, jos virhe on sellainen, että sitä ei voida esittää muilla sivuilla.

Parametrit: Virhetiedot, käyttäjätiedot.

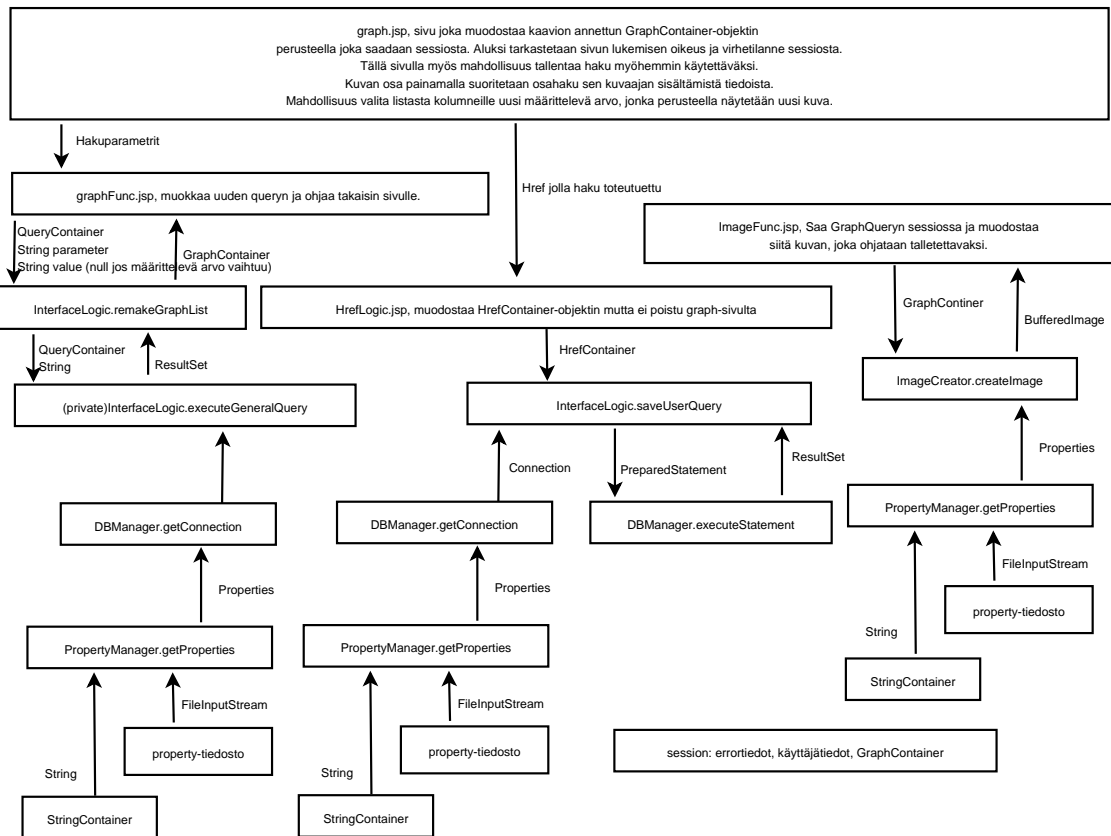
Tuotokset: Virheilmoitus

Poikkeustilanteet: Sessiossa ei virhetietoja

Oikeudet:

6.7 Kaavio

Tämä komponentti näyttää haun perusteella kaavion, joka on mahdollista tallettaa käyttäjälle kuvana. Kaavion muokkaaminen on mahdollista, esimerkiksi luokitteleva parametri on mahdollista vaihtaa.



Kuva 8: Kaavio

6.7.1 Graph.jsp

Palvelu: Sivu muodostaa kaavion annetun GraphContainer-objektin perusteella, joka saadaan sessiosta. Aluksi tarkastetaan sivun lukemisen oikeus ja virhetilanne sessiosta. Tällä sivulla myös mahdollisuus tallentaa haku myöhemmin käytettäväksi. Valitsemalla kuvan osa (yksi tai useampi pylväs) suoritetaan osahaku sen kuvaajan sisältämistä tiedoista. Tarjolla olevasta valikosta voidaan valita uusi luokitteleva parametri joita on mm. Letter(SenderRank, RecRank, RelCode, Copyright, Complete), Person(Sex, Region, County, SocMob, Rank, FatherRank, EduCode, Religion) tai muuttaa yhden pylvään kuvaamaa aikajaksoa, jolloin tieto lähetetään graphFunc.jsp:lle

Parametrit: GraphContainer, käyttäjätiedot

Tuotokset: Kaavio

Poikkeustilanteet: GraphContainerissa ei tarvittavia tietoja

Oikeudet: Käyttäjän pitää olla sessiossa.

6.7.2 GraphFunc.jsp

Palvelu: Kutsuu graph.jsp:stä saatujen tietojen perusteella remakeGraph-metodia, joka palauttaa uuden GraphContainerin, joka laitetaan sessioon. Käyttäjä ohjataan graph.jsp-sivulle.

Parametrit: QueryContainer, String parameter, String value

Tuotokset: GraphContainer

Poikkeustilanteet: Uuden kuvan muokkaaminen ei onnistu, parameter on null

Oikeudet: Käyttäjän pitää olla sessiossa

6.7.3 HrefLogic.jsp

Palvelu: Muodostaa HrefContainer-objektin ja tallettaa sen käyttäjälle, mutta ei poistu graph-sivulta.

Parametrit: Href, jolla kaavio oli muodostettu

Tuotokset: HrefContainer

Poikkeustilanteet: Tiedoston tai tietokannan lukeminen ei onnistu

Oikeudet: QuerySave

6.7.4 ImageFunc.jsp

Palvelu: Saa GraphContainerin sessiosta ja muodostaa siitä kuvan, joka ohjataan tallettavaksi imageExport.jsp-sivulle.

Parametrit: GraphContainer, käyttäjätiedot

Tuotokset: BufferedImage

Poikkeustilanteet: GraphContainerissa ei tarvittavia tietoja. Kuvan muodostus epäonnistuu

Oikeudet: Käyttäjän pitää olla sessiossa

6.8 Kirjenäyttö

Näyttää yhden kirjeen tiedot.

6.8.1 Letter.jsp

Palvelu: Sivun näyttää yhden kirjeen tiedot. Tälle sivulle tullaan ainoastaan sivun letterFunc.jsp:n kautta, joka hakee kirjeen tekstit, jotka tässä näytetään. Listasta voidaan valita lähettäjä, kokoelma tai vastaanottaja. Tällöin henkilön id annetaan letterFunc.jsp-sivulle, joka käsittelee sen.

Parametrit: LetterContainer, ParsedText, AnnotatedText, PlainText

Tuotokset: CollectionID, PersonID.

Poikkeustilanteet: Sessiossa ei tietoja

Oikeudet: Käyttäjän pitää olla sessiossa

6.8.2 LetterFunc.jsp

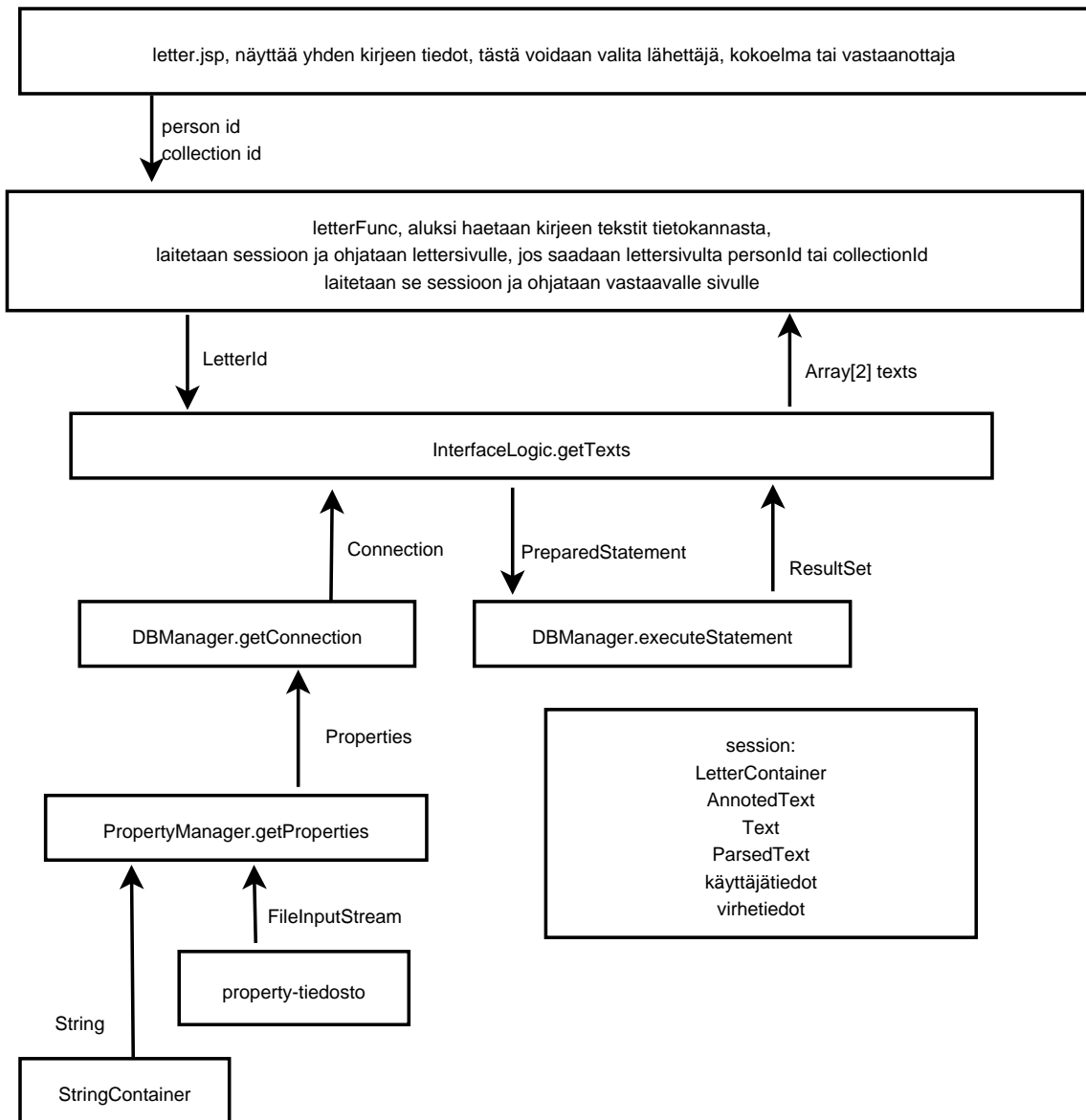
Palvelu: Tämä sivu ottaa vastaan LetterContainerin, hakee sille tekstit ja laittaa sen sessioon. Jos saadaan kokoelman tai henkilön id, talletetaan se sessioon ja ohjataan vastaavalle sivulle.

Parametrit: LetterContainer, Käyttäjätiedot.

Tuotokset: LetterContainer, ParsedText, AnnotatedText, PlainText, PersonID

Poikkeustilanteet: Tiedoston tai tietokannan lukeminen ei onnistu

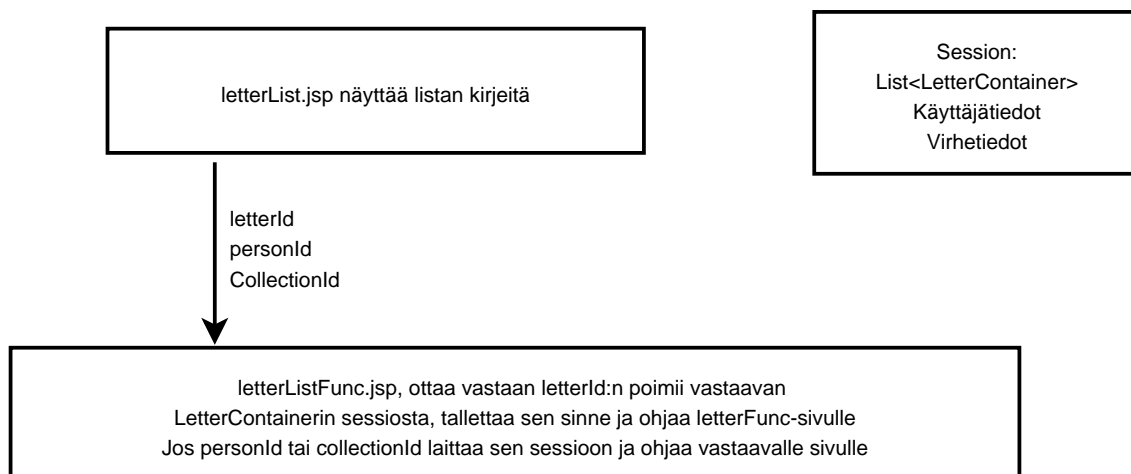
Oikeudet: Käyttäjän pitää olla sessiossa



Kuva 9: Kirjenäyttö

6.9 Kirjelistanäyttö

Näyttää listan kirjeitä.



Kuva 10: Kirjelistanäyttö

6.9.1 LetterList.jsp

Palvelu: Näyttää listan kirjeitä. Tästä näytöstä voidaan valita yksittäinen kirje, kokoelma tai henkilö, jonka tiedot välitetään letterListFunc.jsp-sivulle. Tarjotaan mahdollisuus tallettaa lista tiedostoksi, jolloin ohjataan sivulle exportLetterdata.jsp ja jätetään lista containeriin.

Parametrit: List<LetterContainer>

Tuotokset: CollectionID, PersonID, LetterID

Poikkeustilanteet: Sessiossa ei tietoja

Oikeudet: Käyttäjän pitää olla sessiossa

6.9.2 LetterListFunc.jsp

Palvelu: Ottaa vastaan letterId:n, poimii vastaavan LetterContainerin sessiosta, tallettaa sen sinne ja ohjaa letterFunc-sivulle. Jos CollectionID, PersonID tai LetterID, laittaa sen sessioon ja ohjaa vastaavalle näyttösivulle.

Parametrit: String letterId/personId, Käyttäjätiedot.

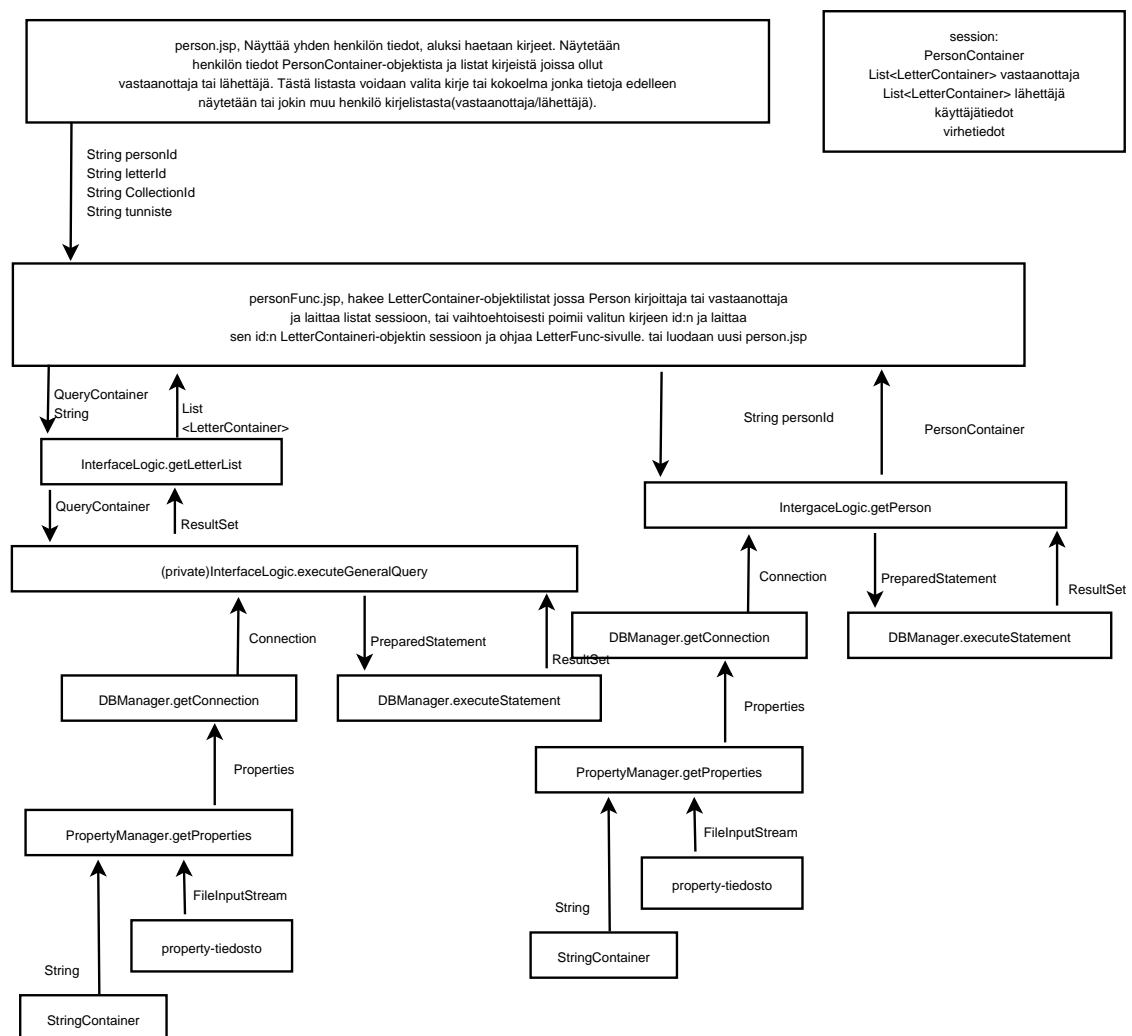
Tuotokset: LetterContainer, CollectionID, PersonID, LetterID

Poikkeustilanteet: Saatu string null

Oikeudet: Käyttäjän pitää olla sessiossa

6.10 Henkilönäyttö

Näyttää yhden henkilön tiedot.



Kuva 11: Henkilönäyttö

6.10.1 Person.jsp

Palvelu: Näyttää yhden henkilön tiedot. Tälle sivulle tullaan ainoastaan sivulta `personFunc.jsp`, jolloin sessiossa on sivun toiminnan kannalta tärkeät tiedot. Näytetään henkilön tiedot `PersonContainer`-objektista ja listat kirjeistä, joissa henkilö on ollut vastaanottaja tai lähettäjä. Tästä listasta voidaan valita kirje tai kokoelma, jonka tietoja edelleen näytetään tai joku muu henkilö kirjelistasta (vastaanottaja / lähettäjä).

Parametrit: `List<LetterContainer>` sender, `List<LetterContainer>` receiver, käyttäjätiedot

Tuotokset: Container ID, Person ID, Letter tai ID Collection ID

Poikkeustilanteet: Sessiossa ei tietoja

Oikeudet: Käyttäjän pitää olla sessiossa

6.10.2 PersonFunc.jsp

Palvelu: Tultaessa sivulle haetaan aluksi kirjeet, jossa Person on kirjoittajana tai vastaanottajana sekä PersonContainer personID:n perusteella, jos sitä ei ole sessiossa (tultaessa henkilölistasivulta sessiossa on valittu PersonContainer, eikä sitä tarvitse hakea). Laitetaan kirjelistat sessioon ja ohjataan käyttäjä person.jsp-sivulle. Jos taas tälle sivulle tullaan person.jsp-sivulta, poimitaan kirjeen, henkilön tai kokoelman Id ja laittetaan se sessioon ja siirrytään vastaavalle näyttösivulle.

Parametrit: String letterId/personId/collectionID, tunniste, jonka perusteella päätellään mikä id saatiin, käyttäjätiedot

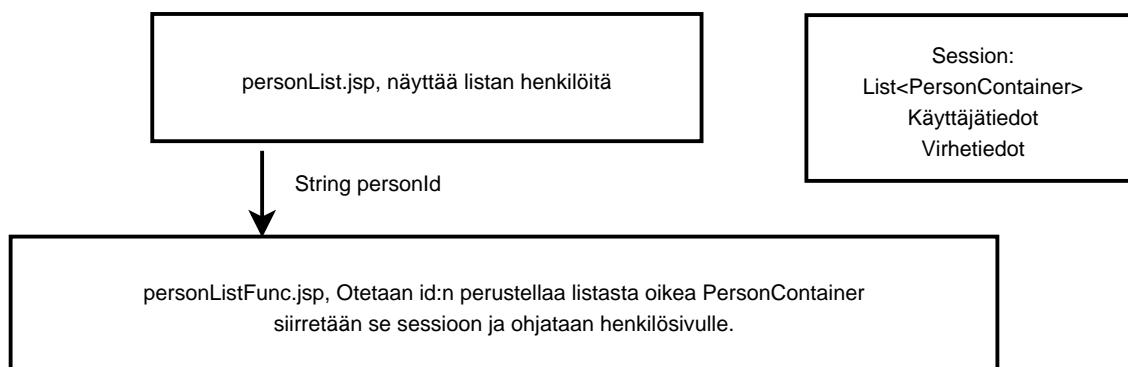
Tuotokset: PersonContainer, Lista LetterContainereita, joissa henkilö on kirjoittaja tai vastaanottaja, letterId/personId/collectionID

Poikkeustilanteet: Saatu string null, tietokannan luku ei onnistu

Oikeudet: Käyttäjän pitää olla sessiossa

6.11 Henkilölistanäyttö

Näyttää listan henkilöitä



Kuva 12: Henkilölistanäyttö

6.11.1 PersonList.jsp

Palvelu: Näyttää listan henkilöitä. Listasta voidaan valita henkilö, jonka tiedot näytetään omassa ikkunassa. Lista voidaan tallettaa käyttäjälle, jolloin lista jätetään sessioon ja käyttäjä ohjataan sivulle exportPersondata.jsp.

Parametrit: List<PersonContainer>, käyttäjätiedot

Tuotokset: -

Poikkeustilanteet: Sessiossa ei tietoja

Oikeudet: Käyttäjän pitää olla sessiossa

6.11.2 PersonListFunc.jsp

Palvelu: Ottaa id:n perusteella listasta oikean PersonContainerin. Se siirretään sessioon ja ohjataan personFunc.jsp-sivulle, joka näyttää sen tiedot.

Parametrit: String personId, käyttäjätiedot

Tuotokset: PersonContainer

Poikkeustilanteet: Saatu string null tai vastaavaa Containeria ei löydy sessiosta

Oikeudet: Käyttäjän pitää olla sessiossa

6.12 Kokoelmanäyttö

Näyttää yhden kokoelman tiedot ja sen sisältämät kirjeet.

6.12.1 Collection.jsp

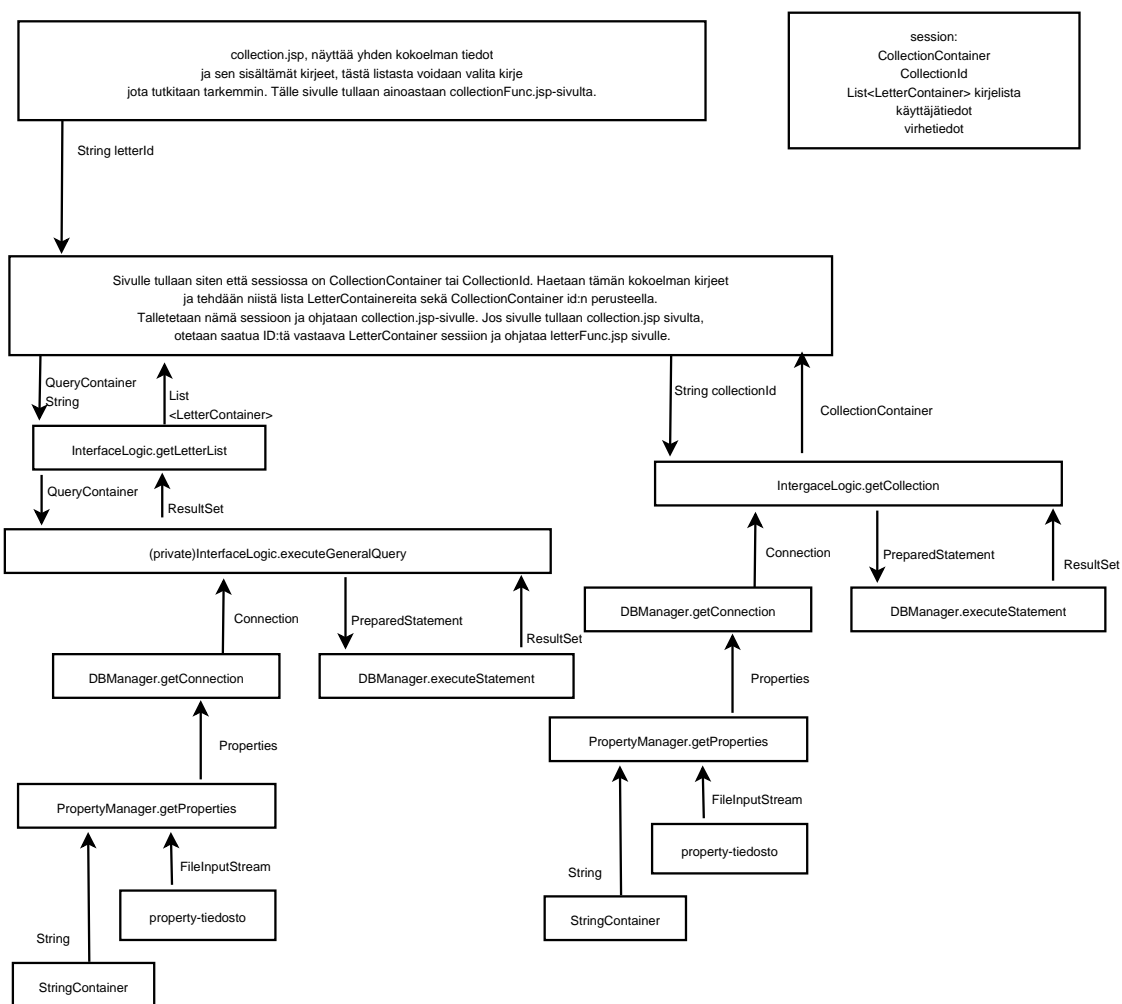
Palvelu: Näyttää yhden kokoelman tiedot ja sen sisältämät kirjeet. Listasta voidaan valita kirje, jota tutkitaan tarkemmin. Tälle sivulle tullaan ainoastaan collectionFunc.jsp-sivulta.

Parametrit: CollectionContainer, List<LetterContainer>

Tuotokset: Letter ID.

Poikkeustilanteet: Sessiossa ei tietoja

Oikeudet: Käyttäjän pitää olla sessiossa



Kuva 13: Kokoelmanäyttö

6.12.2 CollectionFunc.jsp

Palvelu: Sivulle tullaan siten, että sessiossa on CollectionContainer tai CollectionId. Haetaan tämän kokoelman kirjeet ja tehdään niistä lista LetterContainereita sekä CollectionContainer id:n perusteella. Talletetaan nämä sessioon ja ohjataan collection.jsp-sivulle. Jos sivulle tullaan collection.jsp-sivulta, otetaan saatua ID:tä vastaava LetterContainer sessioon ja ohjataan letterFunc.jsp sivulle.

Parametrit: String letterID, CollectionID, CollectionContainer

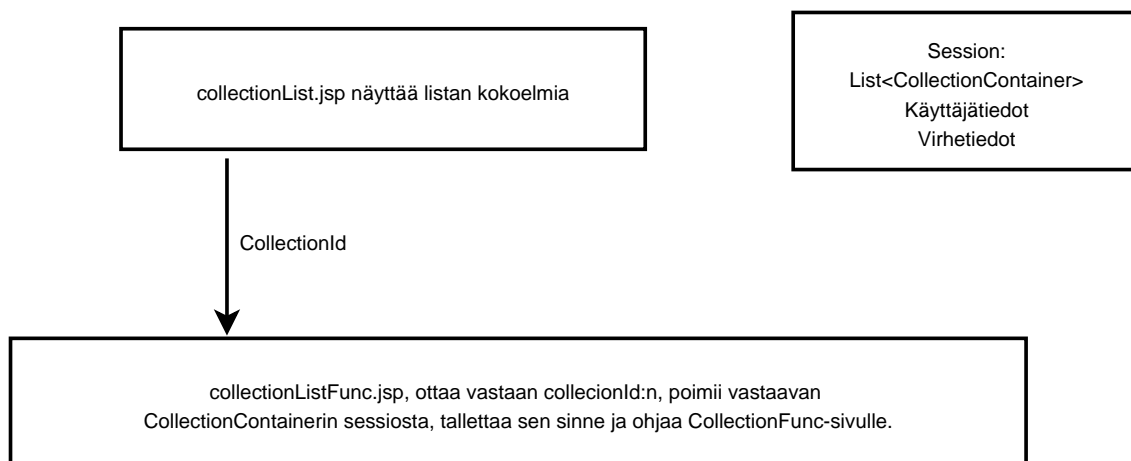
Tuotokset: LetterContainer, List<LetterContainer>, CollectionContainer

Poikkeustilanteet: ID:lle ei vastaavaa Containeria

Oikeudet: Käyttäjän pitää olla sessiossa

6.13 Kokoelmalistanäyttö

Näyttää listan kokoelmia.



Kuva 14: Kokoelmalistanäyttö

6.13.1 CollectionList.jsp

Palvelu: Näyttää listan kokoelmia, voidaan valita container, jonka tarkat tiedot näytetään.

Parametrit: List<CollectionContainer>

Tuotokset: Container ID

Poikkeustilanteet: Sessiossa ei tietoja

Oikeudet: Käyttäjän pitää olla sessiossa

6.13.2 CollectionListFunc.jsp

Palvelu: Otetaan saatu kokoelman ID:n perusteella sessiosta CollectionContainer, laiteetaan se sessioon ja ohjataan collectionFunc.jsp-sivulle.

Parametrit: String container ID

Tuotokset: CollectionContainer

Poikkeustilanteet: Sessiossa ei tietoja

Oikeudet: Käyttäjän pitää olla sessiossa

6.14 Uloskirjautuminen

Komponentti uloskirjautumista varten

6.14.1 LogoutFunc.jsp

Palvelu: Tämä sivu liitetään muihin käyttäjälle näytettäviin sivuihin. Se tarjoaa ainoastaan mahdollisuuden kirjautua ulos. Tällöin sessio poistetaan ja käyttäjä ohjataan login.jsp-sivulle.

Parametrit: -

Tuotokset: Logout-viesti

Poikkeustilanteet: -

Oikeudet: Käyttäjän pitää olla sessiossa

6.15 Työkalupalkki

Komponentti sivun yläosassa näytettäviä linkkejä ja käyttäjätietoja varten

6.15.1 ToolBar.jsp

Palvelu: Tämä sivu liitetään muihin käyttäjälle näytettäviin sivuihin. Se tarjoaa käyttäjätietojen perusteella linkkejä eri toimintoihin, sekä tietoja käyttäjästä.

Parametrit: -

Tuotokset: Työkalupalkki

Poikkeustilanteet: Sessiossa ei käyttäjätietoja

Oikeudet: Käyttäjän pitää olla sessiossa

7 Luokat

Javadoc löytyy osoitteesta

<http://www.cs.helsinki.fi/group/halaan/doc/>

Liite 1. Tietokannan luontitiedosto

```
CREATE TABLE Collection (  
Name varchar(20) NOT NULL,  
Filename varchar(8),  
FromYear number(4),  
ToYear number(4),  
CONSTRAINT pk_col PRIMARY KEY (Name)  
);  
  
CREATE TABLE Person (  
PersonCode varchar(20) NOT NULL,  
Sex char(1),  
Region char(1),  
County varchar(5),  
SocMob char(1),  
LastName varchar(255),  
FirstName varchar(255),  
TitleOccupation varchar(255),  
YBirth number(4),  
YDeath number(4),  
Lifespan varchar(255),  
Rank varchar(3),  
FatherRank varchar(3),  
Father varchar(1000),  
PBirth char(1),  
MigCode varchar(3),  
Migration varchar(1500),  
EduCode varchar(4),  
Education varchar(1500),  
Career varchar(1500),  
Religion char(1),  
DNB varchar(50),  
Notes varchar(1500),  
SentLettcont varchar(10),  
RecLettcont varchar(10),  
Complete char(1),  
Updated date,  
NewBoolean1 char(1),  
NewBoolean2 char(1),  
NewText1 varchar(50),  
NewText2 varchar(255),  
NewNumber number(5),  
CONSTRAINT pk_per PRIMARY KEY (PersonCode)  
);
```

```
CREATE TABLE Letter (  
LetterID varchar(15) NOT NULL,  
Collection varchar(20) NOT NULL,  
Sender varchar(20) NOT NULL,  
SenderRank varchar(3),  
SenderStatus varchar(255),  
MultiSenders char(1),  
Recipient varchar(20) NOT NULL,  
RecRank varchar(3),  
RecStatus varchar(255),  
MultiRec char(1),  
Year number(4),  
YearUncertain char(1),  
WordCount number(5),  
RelCode varchar(2),  
Relationship varchar(255),  
Place varchar(255),  
Authenticity varchar(3),  
LetterDate varchar(255),  
AddressFormula char(1),  
ClosingFormula char(1),  
ContentType varchar(255),  
LetterNotes varchar(1000),  
CorrespondentNotes varchar(1000),  
LetterNumber varchar(50),  
PageNumber varchar(50),  
Source varchar(500),  
Copyright char(1),  
Complete char(1),  
Updated date,  
NewBoolean1 char(1),  
NewBoolean2 char(1),  
NewText1 varchar(50),  
NewText2 varchar(255),  
NewNumber number(5),  
CONSTRAINT pk_let PRIMARY KEY (LetterID),  
CONSTRAINT fk_letcol FOREIGN KEY (Collection)  
REFERENCES Collection (Name) ON DELETE CASCADE,  
CONSTRAINT fk_letsen FOREIGN KEY (Sender)  
REFERENCES Person (PersonCode) ON DELETE CASCADE,  
CONSTRAINT fk_letrec FOREIGN KEY (Recipient)  
REFERENCES Person (PersonCode) ON DELETE CASCADE  
);
```

```
CREATE TABLE Text (  
Letter varchar(15) NOT NULL,  
Version char(1) NOT NULL,  
Text clob,  
CONSTRAINT pk_txt PRIMARY KEY (Letter, Version),  
CONSTRAINT fk_txtlet FOREIGN KEY (Letter)  
    REFERENCES Letter (LetterID) ON DELETE CASCADE  
);
```

```
CREATE TABLE CeecGroup (  
Name varchar(15) NOT NULL,  
Description varchar(50),  
UserAdmin char(1) NOT NULL,  
CorpusAdmin char(1) NOT NULL,  
QuerySave char(1) NOT NULL,  
CorpusExport char(1) NOT NULL,  
MetadataExport char(1) NOT NULL,  
Copyright char(1) NOT NULL,  
CONSTRAINT pk_grp PRIMARY KEY (Name)  
);
```

```
CREATE TABLE CeecUser (  
UserID varchar(8) NOT NULL,  
Password varchar(10) NOT NULL,  
UserGroup varchar(15) NOT NULL,  
FirstName varchar(50) NOT NULL,  
LastName varchar(50) NOT NULL,  
Email varchar(50),  
Phone varchar(50),  
CONSTRAINT pk_use PRIMARY KEY (UserID),  
CONSTRAINT fk_usegrp FOREIGN KEY (UserGroup)  
    REFERENCES CeecGroup (Name) ON DELETE CASCADE  
);
```

```
CREATE TABLE CeecQuery (  
Time date NOT NULL,  
Owner varchar(8) NOT NULL,  
Name varchar(50),  
Query varchar(500) NOT NULL,  
CONSTRAINT pk_qry PRIMARY KEY (Time, Owner),  
CONSTRAINT fk_qryuse FOREIGN KEY (Owner)  
    REFERENCES CeecUser (UserID) ON DELETE CASCADE  
);
```

```
CREATE INDEX text_letter on Text(Letter);
```

```
CREATE INDEX letter_sender on Letter(Sender);  
CREATE INDEX letter_recipient on Letter(Recipient);  
CREATE INDEX letter_collection on Letter(Collection);  
CREATE INDEX ceecquery_owner on CeecQuery(Owner);
```