KOPPELO

Tietotekniikan cum laude -työprojekti

Minna Hillebrand
Markus Silván
Antti Vanhanen
Marko Ylitalo

Sovellusraportti
18.6.2002

Jyväskylän yliopisto
Tietotekniikan laitos

Tiivistelmä

KOPPELO-projekti kehitti Jyväskylän yliopiston tietotekniikan laitokselle WWW-pohjaisen opinnäytteiden hallintasovelluksen. Sovellusraportissa käsitellään toteutetun sovelluksen kokonaisrakennetta sekä koodi- että käyttöliittymätasolla. Kaikkia sovellukseen suunniteltyja ominaisuuksia ei projektin puitteissa ehditty toteuttaa, joten tärkeimmät toteuttamatta jääneet osat on kuvattu tässä dokumentissa jatkokehitysideoina.

Tietoa projektista

Tekijät:

Minna Hillebrand (`mmhilleb@st.jyu.fi`)
Markus Silván (`mape@st.jyu.fi`)
Antti Vanhanen (`vanhanan@st.jyu.fi`)
Marko Ylitalo (`ylimark@st.jyu.fi`)

Työ: Sovellussuunnitelma tietotekniikan cum laude -työprojektiin.

Tilaaja ja teettäjä: Jyväskylän yliopiston tietotekniikan laitos.
Jukka-Pekka Santanen toimi sekä tilaajan edustajana että laitoksen vastaavana ohjaajana. Teknisinä ohjaajina toimivat Heikki Uuksulainen ja Pauli Kujala.

Työtila: Agora, huone AgC222.2

Avainsanat: tietotekniikan cum laude -työprojekti, opinnäytteet, JSP, KOTKA-järjestelmä, KORPPI-sovellus, KOLIBRI-sovellus.

Sisältö

1 Johdanto

KOPPELO-projekti toteutti Jyväskylän yliopiston tietotekniikan laitokselle opinnäytteiden hallintasovelluksen. KOPPELO-projekti oli tietotekniikan laitoksen kevään 2002 cum laude -työprojekti.

Kevään 2002 KOPPELO-projektiryhmään kuuluivat Minna Hillebrand, Markus Silván, Antti Vanhanen sekä Marko Ylitalo. Tilaajana toimi Jyväskylän yliopiston tietotekniikan laitos, jota edusti Jukka-Pekka Santanen. Teknisinä ohjaajina toimivat Pauli Kujala sekä Heikki Uuksulainen.

KOPPELO-projekti oli jatkoa KURKI-sovelluksen jatkokehittämisestä liikkeelle lähteneeseen opetus- ja tutkimushallintajärjestelmän kehittämiseen. Vuonna 1998 toteutettua KURKI-projektia seurasivat syksyn 2000 KOTKA-projekti, kevään 2001 KORPPI- ja HALKO-projektit sekä syksyn 2001 KOLIBRI-projekti. Projekteissa tuotettu opetus- ja tutkimushallintajärjestelmä on nimeltään KOTKA.

KOPPELO-projekti toteutti KOTKAan opinnäytteiden hallintasovelluksen, jolla voidaan keskitetysti hallita laitoksilla tehtäviä ja valmistuvia opinnäytteitä. Lisäksi sovellus toimii rekisterinä toteutetuille opinnäytteille, jolloin mm. laitoksen henkilökunta ja opiskelijat voivat hakea tietoa valmistuneista opinnäytteistä.

Raportissa kuvataan sovelluksen arkkitehtuuria ja toteutettuja ominaisuuksia sekä esitetään sovelluksen toimintaa myös yksittäisten JSP-sivujen tasolla. Projektin etenemistä ja tavoitteiden toteutumista käsitellään projektiraportissa. Sovelluksen testausmenetelmistä ja -tuloksista on kirjoitettu erillinen testausraportti. Sovelluksen tarvitsemista tietokantamuutoksista on myös kirjoitettu oma raportti.

Lukuun 2 on kerätty projektissa ja sen dokumenteissa käytettyjä termejä. Luvussa 3 on lyhyesti kuvattu sovellukselle asetettujen tavoitteiden ja ominaisuuksien toteutumista. Lukuihin 4 ja 5 on koottu sovelluksen taustalla oleva palvelinratkaisu, toteuttamistekniikoita ja tietokantaratkaisut.

Sovelluksen toteuttaminen JSP-sivuiksi ja sovellusta hahmottavat sivustokartat on esitelty luvuissa 6 ja 7. Käyttöliittymä ja tekniset kuvaukset yksittäisistä JSP-sivuista on esitetty luvuissa 8–14. Sovelluksen testausta on lyhyesti kuvattu luvussa 15. Sovelluksesta havaitut virheet ja jatkokehitysideat on koottu lukuihin 16 ja 17. Sovelluksen asennusohjeet ja ylläpitäjän huomioita on kirjattu lukuihin 18 ja 19.

2 Termistöä

Taulukoissa 1 ja 2 on esitelty toteutusympäristöön ja projektiin liittyvää termistöä.

TERMI	**SELITYS**
Apache	on ilmainen HTTP-palvelinohjelmisto.
CSS	eli Cascading Style Sheets on WWW-sivujen ulkoasua kuvaava kieli.
CVS	eli Concurrent Versions System on versionhallintaan tarkoitettu ohjelmisto.
Delphi	on Inprisen ohjelmistonkehitysympäristö, jossa ohjelmointikielenä käytetään Object Pascal -kieltä.
GNUJSP	on ilmainen JSP-moottori.
HTML	on WWW-sivujen rakennetta kuvaava kieli.
HTTP	on WWW-arkkitehtuurin käyttämä tiedonsiirtoprotokolla.
Java	on Sunin kehittämä laitteistoriippumaton olio-ohjelmointikieli.
Java-pavut	(engl. *Java Beans*) ovat Java-ohjelmointikielellä luotuja komponentteja, joita voidaan kutsua JSP-sivuilla.
JDBC	eli Java Database Connectivity on Java-teknologian käyttämä tietosilta erilaisiin tietokantoihin.
Jserv	on ilmainen Java-servlettimoottori.
JSP	eli Java Server Pages on skriptaustyylinen ohjelmointikieli.
Keksi	(tai eväste, engl. *cookie*) on Netscapen luoma, mutta nykyään jo standardoitu menetelmä saada WWW-sivuista vuorovaikutteisempia. Palvelin lähettää pieniä tietopaketteja selaimelle, jonka perusteella käyttäjä asetuksineen tunnistetaan jatkossa.
Poolman	on ohjelma, joka kontrolloi tietokantayhteyksiä.
PostgreSQL	on ilmainen tietokannanhallintajärjestelmä.
Skripti	on ohjelmointikieltä muistuttava ja usein hieman yksinkertaistettu tulkattava kieli.
Skriptaus	tarkoittaa WWW-sivujen tapauksessa ohjelmakoodin kirjoittamista HTML-dokumenttien sisään. Palvelinpuolen skriptauksessa koodi ajetaan palvelinkoneessa ja asiakaspään skriptauksessa selaimessa.
Selain	on ohjelma, joka käyttäjän koneella tulkkaa HTML-kieliset sivut esitettävään muotoon.
Palvelin	on ohjelmisto, joka WWW-sovellusten tapauksessa palvelee asiakkaana toimivien selainten hakupyyntöjä.
Servletti	on palvelimella sijaitseva sovelma (engl. *applet*), joka toteuttaa HTTP-palvelimen pyynnöstä tietyn toiminnon.
SQL	eli Structured Query Language on tietokantojen hallintaan kehitetty standardi kieli.
Tomcat	on ilmainen servletti- ja JSP-moottori.

Taulukko 1: Toteutusympäristöön liittyvää termistöä.

TERMI	SELITYS
HALKO	on Heinolan kansalaisopistolle toteutettu opetuksenhallintajärjestelmä.
KOTKA	on Jyväskylän yliopiston tietotekniikan laitoksella kehitetty opetus- ja tutkimushallintajärjestelmä.
KORPPI	on KOTKA-järjestelmän kurssikirjanpitomoduuli.
KOLIBRI	on KOTKA-järjestelmän kalenteriosio.
KURKI	on Jyväskylän yliopistossa käytössä oleva Intrabuilderilla toteutettu WWW-pohjainen kurssikirjanpitojärjestelmä.
KUUKKELI	on Antti Viinikan Jyväskylän yliopistossa kehittämä järjestelmä ohjausaikojen ajanvaraukseen ja hallintaan.
Käyttötapaus	(engl. *use case*) on käyttäjän tai sovelluksen toimintoa tietyn tehtävän suorittamiseksi kuvaava dokumentti.
Moduuli	on Kotka-järjestelmässä tietyn tehtäväjoukon suorittamiseen kehitetty sovellus ja käyttöliittymä.

Taulukko 2: KOTKA-järjestelmään liittyvää termistöä.

Taulukossa 3 esitetään projektin eri dokumenteissa käytettävää opinnäytteisiin liittyvää termistöä.

TERMI	SELITYS
Aihe	on aihe-ehdotuksesta tietylle linjalle hyväksytty opinnäytteen aihe.
Aihe-ehdotus	on opinnäytteen aiheeksi esitetty idea.
Opinnäyte	on opintojakso, jossa korostuu opiskelijan oma työ. Sen tuloksena syntyy tutkielma, dokumentti, sovellus tai jokin muu teos. Opinnäytteellä tarkoitetaan esimerkiksi graduja, seminaareja ja työharjoitteluja.
Vastaava ohjaaja	on opinnäytteen pääasiallinen ohjaaja. Vastaavalta ohjaajalta voi käydä varaamassa aiheen itselleen, ja yleensä vastaava ohjaaja myös ohjaa opinnäytettä sen edetessä.
Ohjaaja	on opinnäytteen ohjaaja.
Tarkastaja	tarkastaa ja arvostelee opinnäytteen.
Kurssi	on KOTKA-tietokannan käsite yleiselle kurssinimikkeelle. Jokainen opinnäyte liittyy johonkin kurssiin, jotta opinnäytteeseen liittyisi yksikäsitteinen kurssikoodi.
Kurssi-instanssi	on KOTKA-tietokannan käsite tiettynä lukukautena tai muuna ajankohtana järjestettävälle kurssille.
Opinnäytteen vaihe	jakaa opinnäytteen elinkaaren ja ohjausprosessin osiin. Niitä ovat mm. ehdotettu aiheeksi, hyväksytty aiheeksi, tekeillä, hyväksytty ja keskeytetty.
Tapahtuma	on opinnäytteeseen liittyvä raportointi- tai ohjaustapahtuma. Jokaiseen tapahtumaan voi edelleen liittyä oma työvaihe.
Työtehtävä	Esimerkiksi työprojektin tapauksessa opinnäytteen suorituksesta voidaan erottaa projektisuunnitelman, sovellussuunnitelman ja varsinaisen ohjelmoinnin suorittaminen. Nämä työtehtävät voidaan edelleen luokitella esimerkiksi suunnitteluun, ohjelmointiin ja dokumentointiin.

Taulukko 3: KOPPELO-projektiin liittyvää termistöä.

3 Asetettujen tavoitteiden toteutuminen

KOPPELO-sovelluksen yleisenä vaatimuksena oli olla helppokäyttöinen opinnäytetöiden hallintasovellus Jyväskylän yliopiston opiskelijoiden ja henkilökunnan käyttöön. Projektin alussa ryhmälle annettiin lista ohjelmalta toivottavista ominaisuuksista ja listaa laajennettiin sovelluksen potentiaalisten käyttäjien haastattelujen pohjalta. Luvussa on esitetty sovellukselle määriteltyjen vaatimusten toteutuminen.

3.1 Opiskelijan käyttöliittymä

Opiskelijoiden osaan KOPPELO-sovellusta kehitettiin ilmoitustaulun tyyppinen pääsivu opiskelijan omien opinnäytteiden ja näiden etenemisen seurantaan. Lisäksi opiskelija pääsee tarkastelemaan yksityiskohtaisemmin tekemiään opinnäytteitä. Opiskelijalla on oikeus lisätä tapahtumia niihin opinnäytteisiin, joissa hän on tekijänä. Tapahtumat näkyvät kaikille opinnäytteeseen liitetyille henkilöille.

Opiskelija voi sovelluksella esittää laitokselleen opinnäytteiden aiheita KOTKA-järjestelmään. Hän voi muuttaa aihe-ehdotuksensa tietoja ja seurata aihe-ehdotuksen tilaa.

Opiskelija voi hakea vapaalla sanahaulla tai asettamillaan hakuehdoilla tietoa opinnäytteiden aiheista tai valmistuneista opinnäytteistä. Alkuperäisten vaatimusten mukaan opiskelijan ei pitänyt pystyä hakemaan sovelluksella keskeytettyjä tai poistettuja opinnäytteitä. Sovelluksen kehittämisen loppuvaiheilla tätä ominaisuutta ei kuitenkaan poistettu, sillä se oli jo toteutettu henkilökunnan käyttöliittymän yhteydessä.

Opiskelijan käyttöliittymältä vaaditut ominaisuudet toteutettiin kokonaisuudessaan KOPPELO-sovelluksessa. Toteutettua opiskelijan sivustoa on kuvattu luvussa 7.3.

3.2 Henkilökunnan käyttöliittymä

Laitoksen henkilökunnalle toteutettiin ilmoitustaulun tyyppinen pääsivu, jolla henkilökuntaan kuuluvat voivat seurata omiin ohjaamiinsa opinnäytteisiin liittyvien opinnäytteiden tilaa ja tapahtumia. Lisäksi opinnäytteisiin liitetyillä henkilökunnan jäsenillä on oikeus muokata opinnäytteen tietoja, liittää opinnäytteeseen henkilöitä ja työtehtäviä sekä arvioida opiskelijoita ja hyväksyä opinnäyte.

Opinnäytteen tietojen hallintaoikeuksia voidaan rajata edelleen rooli- ja käyttäjäkohtaisesti esimerkiksi siten, että opinnäytteen ohjaajalla on ainoastaan oikeus muuttaa opinnäytteen yleisiä tietoja. Arvostelijalle voidaan sen sijaan antaa vaikkapa tietojen ylläpitämisen ja opinnäytteen hyväksymisen oikeudet.

Henkilökuntaan kuuluvat voivat ehdottaa sovelluksella opinnäytteiden aiheita. Rajatulle käyttäjäryhmälle voidaan sovelluksessa antaa oikeus hyväksyä aihe-ehdotuksia opinnäytteiden aiheiksi. Aihe-ehdotuksen antaja voi seurata ehdotuksensa tilaa erillisellä sovelluksen tilalla.

Henkilökuntaan kuuluvilla on oikeus lukea kaikkia järjestelmään syötettyjä aihe-ehdotuksia. Aihe-ehdotusten hyväksyjä voi lisäksi hyväksyä tai hylätä valitsemiaan aihe-ehdotuksia. Määrätyllä käyttäjäjoukolla on oikeus aloittaa opinnäytteen suoritus sellaisista opinnäytteen aiheista, jotka on merkitty sovellukseen varattaviksi.

Hakutoiminnot ovat henkilökunnalle samanlaiset kuin opiskelijoiden puolella. Henkilökunta on oikeutettu hakemaan opinnäytteitä myös keskeytetyistä tai järjestelmästä poistetuksi merkityistä aiheista. Raportointi toteutettiin sovelluksessa siten, että valitsemalla selaimen tulostustoiminnon käyttäjä saa sivusta raporttimaisen version ilman sovelluksen käyttöliittymään liittyviä navigointipalkkeja tai tekstin muotoiluja.

Kaikki henkilökunnan käyttöliittymään määritellyt ominaisuudet toteutettiin KOPPELO-sovellukseen. Joiltain osin sivuston viimeistely jäi kuitenkin kesken. Esimerkiksi yhteyttä JORE-opintorekisterijärjestelmään ei toteutettu. Muut puutteet ovat lähinnä sivujen ulkoasuun liittyviä huomioita.

Toteutettua henkilökunnan sivustoa on kuvattu luvussa 7.4.

3.3 Ylläpidon käyttöliittymä

Sovellukseen liittyvien oikeuksien tarkastelemiseen ja hallintaan luotiin erillinen käyttöliittymä järjestelmän ylläpitäjälle. Ylläpidon käyttöliittymä oli projektin alusta saakka ajateltu vain aikataulun salliessa toteutettavaksi, ja tämän vuoksi se toteutettiin prototyyppitasoisesti.

Oikeuksien hallintaan kuitenkin tarvitaan välttämättä erillinen käyttöliittymä, sillä sovellukseen liittyvien oikeuksien antaminen jo muutamalle käyttäjälle tai käyttäjäryhmälle tuottaa suuren määrän SQL-lauseita. Toteutetulla käyttöliittymällä oikeuksien määrääminen sujuu huomattavasti helpommin kuin tietokantaa suoraan päivittämällä.

Ylläpidolle kaavaillusta käyttöliittymästä jätettiin toteuttamatta valikkomuotoisen tiedon hallintamahdollisuus. Näihin tietoihin liittyvät muutokset ovat lähinnä uusien tietojen lisäystä, joka saadaan nopeammin tehtyä suoraan tietokantaan. Myös tietojen hallintaan liittyvä monikielistäminen jätettiin toteuttamatta, sillä monikielistämisen periaatteita täytyy tarkastella laajemmin koko KOTKA-järjestelmässä.

Toteutettua ylläpitäjän sivustoa on kuvattu luvussa 13.

3.4 KOTKA-järjestelmän ulkopuoliset käyttäjät

Tavoitteena oli toteuttaa KOTKA-järjestelmän ulkopuolisille käyttäjille mahdollisuus hakea laitoksilla valmistuneita opinnäytteitä erilaisin hakuehdoin. Laitokseen sisäiseen hakuun käytetyn viimeistelyn vuoksi tämä ominaisuus jäi kuitenkin KOPPELO-projektin puitteissa toteuttamatta.

3.5 Automaattiset sähköpostiviestit

KOPPELO-sovellus tukee käyttäjien välistä vuorovaikutusta lähettämällä automaattisia sähköposteja määrätyissä tilanteissa. Automaattisia viestejä lähetetään sekä käyttäjien aktiivisen että passiivisen toiminnan perusteella.

Käyttäjän aktiivisen toiminnan tuloksena sähköpostiviestejä lähetetään aiheen käsittelemisen yhteydessä. Aihe-ehdotuksen jättämisestä lähetetään viesti kaikille henkilöille, joilla on oikeus hyväksyä opinnäyte jollekin ehdotetulle linjalle ja opinnäytetyypille.

Aihe-ehdotuksen käsittelemisen yhteydessä aiheen hyväksyjä voi sovelluksella lähettää viestin aiheen ehdottajalle esimerkiksi pyytääkseen aiheesta lisätietoja. Kun aiheen käsittelijä hyväksyy tai hylkää aihe-ehdotuksen, lähetetään tapahtumasta viesti aiheen ehdottajalle.

Aktiivisen toiminnan lisäksi sovellus lähettää automaattisia sähköpostiviestejä sovelluksen käyttäjille. Automaattisten viestien lähettämisen periaatteita on käsitelty tarkemmin luvussa 14.

Sovellus lähettää automaattisen muistutusviestin opinnäytteen aiheen ehdottajalle, jos aiheen tai aihe-ehdotuksen voimassaoloaikaa on jäljellä enää viikko. Varoitusviesti lähetetään päivittäin aiheen poistamiseen saakka, ellei aiheelle anneta lisäaikaa tai aihetta hyväksytä viikon kuluessa. Toteuttamatta jäi automaattisen sähköpostin lähetys silloin, kun aihe merkitään poistetuksi järjestelmästä.

Opinnäytteen suorittajille lähetetään muistutusviesti silloin, jos opinnäytteen ohjaaja on määrittänyt opinnäytteelle muistutusvälin ja opinnäytteeseen viimeksi kirjatusta tapahtumasta on kulunut muistutusvälin pituinen aika.

Automaattisten sähköpostiviestien lähettämisen onnistuminen edellyttää, että käyttäjät ovat syöttäneet KOTKA-järjestelmään sähköpostiosoitteensa.

3.6 Sovelluksen monikielisyys

Sovellukseen liittyvien WWW-sivujen vakiotekstejä ei ole tallennettu tai monikielistetty KOPPELO-projektin puitteissa. Vakiotekstit on kuitenkin asetettu käännöstagin (katso luku 6.7) sisään, jolloin järjestelmään myöhemmin toteutettava monikielisyys heijastuu myös KOPPELO-sovelluksen sivuihin.

Sovelluksen taustalle luodut tietokannan taulut on toteutettu siten, että niihin voidaan tallentaa myös käyttäjän syöttämä tieto monikielisesti. Ajanpuutteen vuoksi sovellukseen ei toteutettu käyttöliittymää arvojen tallentamiseksi eri kielillä. Sovelluksen esittämä tieto kuitenkin esitetään käyttäjälle hänen määräämällä kielellä.

3.7 Yhteenveto sovelluksen ominaisuuksista

Sovelluksesta toteutettiin **opiskelijan käyttöliittymä kokonaisuudessaan**. Oppilas voi ehdottaa opinnäytteen aiheita järjestelmään, seurata opinnäytteensä etenemistä ja liittää siihen tapahtumia. Lisäksi hän voi hakea määräämillään hakuehdoilla tietoa järjestelmässä olevista varattavista ja valmistuneista opinnäytteistä ja niiden aiheista.

Henkilökunnan käyttöliittymä toteutettiin myös kokonaisuudessaan. Henkilökunta voi ehdottaa aiheita järjestelmään ja rajattu käyttäjäjoukko voi hyväksyä aihe-ehdotuksia aiheiksi. Rajatulla joukolla henkilökuntaa on oikeus kiinnittää aiheita opiskelijoille eli aloittaa opinnäytteen suorittaminen. Opinnäytteen ohjaajat voivat ylläpitää opinnäytteen tietoja ja suorittaa opiskelijoiden arvioinnin. Henkilökunta voi hakea määräämillään hakuehdoilla tietoa järjestelmässä olevista vapaista, valmistuneista ja keskeneräisistä opinnäytteistä ja aiheista. Henkilökunta saa raportin hakuehdon täyttävistä tuloksista valitsemalla selaimesta tulostustoiminnon.

Ylläpidon käyttöliittymään toteutettiin oikeuksien hallinnointi KOPPELO-sovelluksessa. **Ylläpidolle ei toteutettu** käyttöliittymää sovellukseen liittyvän **vakiomuotoisen tiedon hallintaan tai monikielistämiseen**.

KOTKA-järjestelmän **ulkopuolisille käyttäjille ei toteutettu** lainkaan mahdollisuutta nähdä sovellukseen syötettyjä hyväksyttyjä opinnäytteitä.

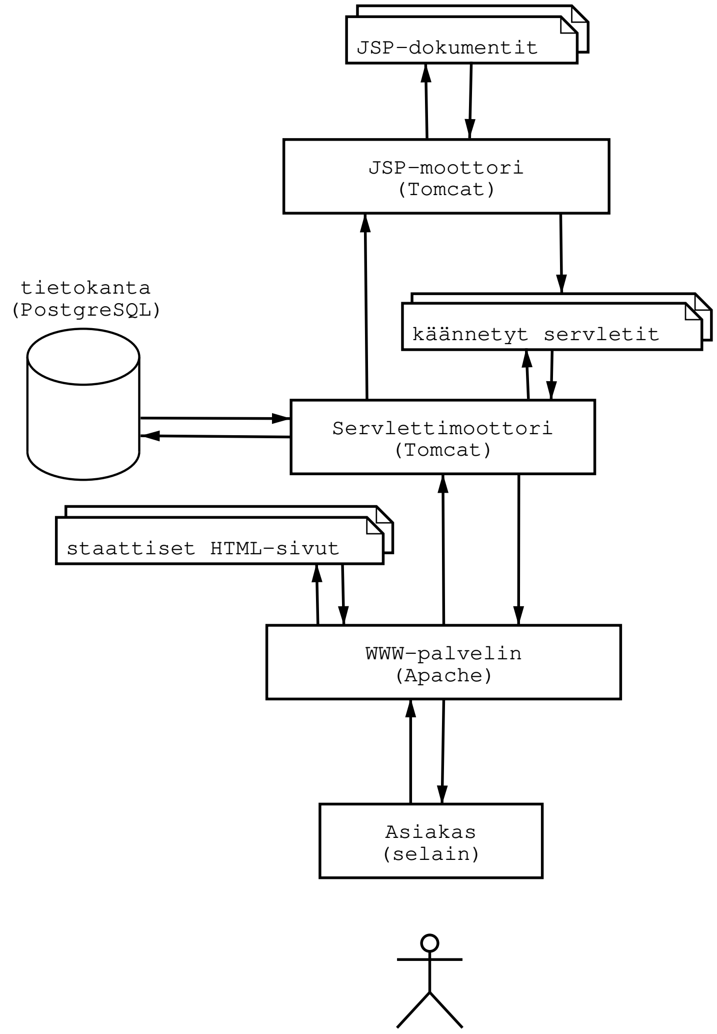
Sovellus lähettää automaattisia sähköpostiviestejä aihe-ehdotuksen antamisesta, aihe-ehdotuksen käsittelystä (hyväksyminen tai hylkääminen) ja opinnäytteen tapahtumien liian pitkästä aikavälistä. Lisäksi automaattinen sähköpostiviesti lähetetään, kun aiheen tai aihe-ehdotuksen voimassaoloaika on päättymässä ja siten se aiotaan automaattisesti merkitä poistetuksi järjestelmästä. **Toteuttamatta jätettiin automaattisen sähköpostiviestin lähettäminen, kun erääntynyt aihe tai aihe-ehdotus poistetaan järjestelmästä**.

Sovelluksessa havaitut virheet on koottu lukuun 16 ja jatkokehitysideat on koottu lukuun 17.

4 Palvelinohjelmistot, toteutustekniikat ja työkalut

Luvussa esitellään projektissa käytetyt työkalut ja ohjelmointitekniikat. Niihin päädyttiin KOPPELO-projektissa, koska KOTKA-järjestelmän muut osiot on toteutettu niillä. Lisäksi KOTKA-projektin testitulosten perusteella ne soveltuivat ko. järjestelmän toteuttamiseen paremmin kuin muut kilpailevat ohjelmistot.

Kuvassa 1 on esitetty KOTKA-järjestelmän palvelinrakennetta.



Kuva 1: KOTKA-järjestelmän rakenne.

4.1 Palvelinohjelmistot

KOTKA-projektin valitsemaa palvelinrakennetta käytettiin myös KOPPELO-projektissa. Palvelinrakenteen pääosat ovat Apache-palvelin, TomCat-niminen servlettimoottori JSP-sivujen kääntämiseen, PostgreSQL tietokantana sekä Poolman-ohjelma servlettimoottorin ja tietokannan välissä hoitamassa tietokantayhteyksiä. Käytetty palvelinrakenne on kuvattu kuvassa 1.

Apache on erittäin suosittu Linux-käyttöjärjestelmän HTTP-palvelin. KOTKA-järjestelmässä HTTP-palvelimena on Apache,joten myös KOPPELO-sovellus käyttää Apache-palvelinohjelmistoa.

TomCat on Apache-projektin haarauma, joka on pääasiassa yhdistetty JSP- ja servlettimoottori, mutta toimii myös itsenäisenä HTTP-palvelimena. KOPPELO-projektin puitteissa TomCat-palvelinta käytetään ainoastaan JSP- ja servlettimoottorina.

Poolman on ilmainen apuohjelma, joka hallinnoi tietokantayhteyksiä. Se toimii myös välimuistina TomCat-palvelimen ja **PostgreSQL**-tietokannan välillä. Tietokannanhallintajärjestelmä PostgreSQL valittiin KOTKA-projektin testeissä syksyllä 2000. Se on Ingres-tietokannanhallintajärjestelmän pohjalta kehitetty vapaaseen lähdekoodiin (engl. *Open Source*) perustuva ohjelmisto.

4.2 JSP-tekniikka

Kuvassa 1 kuvataan järjestelmän rakennetta ja toimintaa. Kun WWW-palvelin vastaanottaa pyynnön selaimelta, se päättelee, onko kyseessä staattinen HTML-sivu vai JSP-sivu. Staattiset sivut Apache palauttaa heti takaisin käyttäjälle.

JSP-sivuja pyydettäessä pyyntö välitetään edelleen servlettimoottorille. Se tutkii, onko kyseistä sivua vastaava servletti jo olemassa. Jos servlettiä ei ole, pyyntö ohjataan edelleen JSP-moottorille, joka generoi sivua vastaavan servletin.

Tämän jälkeen jo valmiina palvelimella ollut tai juuri luotu servletti suoritetaan, jolloin muodostuu selaimen pyytämä HTML-sivu ja se lähetetään takaisin selaimelle. Näin asiakassovelluksena toimiva WWW-selain saa pelkkää HTML-koodia sisältävän sivun esitettäväksi.

4.3 Java-pavut

Java-pavut ovat Java-ohjelmointikielellä toteutettuja luokkia. Papuihin pyritään sijoittamaan mahdollisimman paljon JSP-sivujen Java-koodia, jolloin itse JSP-sivujen luettavuus paranee. Kaikki yleiskäyttöiset metodit sijoitetaan Java-papuihin. Tällöin niitä voidaan kutsua JSP-koodista, eikä samaa Java-koodia tarvitse kirjoittaa uudelleen

JSP-koodin sekaan.

4.4 JSP:n Tag-kirjastot

Tagit ovat HTML-koodin sekaan upotettavia komentoja. Java-papujen tapaan tagit parantavat koodin joustavuutta ja uudelleenkäytettävyyttä. Tageihin kapseloidaan haluttu ohjelmakoodin toiminto.

JSP:n tageista muodostuu kirjastoja, joita voidaan ottaa käyttöön yksittäisissä JSP-sivuissa. Valmiita julkisia JSP Tag -kirjastoja on muun muassa Jakarta-projektilla.

4.5 Tietokanta

KOPPELO-sovellusta suunniteltaessa ei käytetty mitään PostgreSQL-tietokannanhallintajärjestelmän erikoispiirteitä. Toteutetuissa SQL-lauseissa on noudatettu SQL99-standardia. Tuloksena on saatu helposti ympäristöstä toiseen siirrettävä ratkaisu.

KOPPELO-sovelluksen tarvitsemia täydennyksiä KOTKA-järjestelmän tietokantaan suunniteltiin ER-kaavioiden avulla. Lopullista KOPPELO-sovelluksen taulurakennetta käsitellään luvussa 5. Lisäksi tietokannan muutoksista on kirjoitettu erillinen tietokantaraportti ja muutokset on lisäksi kirjattu KOTKA-järjestelmän tietokantaraporttiin.

5 KOPPELO-sovelluksen taulut KOTKA-tietokannassa

KOPPELO-sovelluksen käyttämiä tauluja hahmoteltiin ER-kaavion avulla, jota asteittain tarkennettiin. Kuvassa 2 on lopullinen kuva KOPPELO-sovelluksen tauluista KOTKA-tietokannassa, taulujen kentistä ja yhteyksistä muuhun KOTKA-kantaan.

Tämä luku ei ole täydellinen kuvaus KOPPELO-sovelluksen käyttämistä tauluista ja niiden mahdollistamasta toiminnallisuudesta. Tämän luvun tarkoitus on antaa yleiskuva tauluista ja joistain niiden välisistä suhteita. Tietokantaan liittyvät tiedot on kirjattu tarkemmin sekä projektin tuottamaan tietokantaraporttiin että KOTKA-järjestelmän tietokantaraporttiin.

5.1 Opinnäytteen aihe

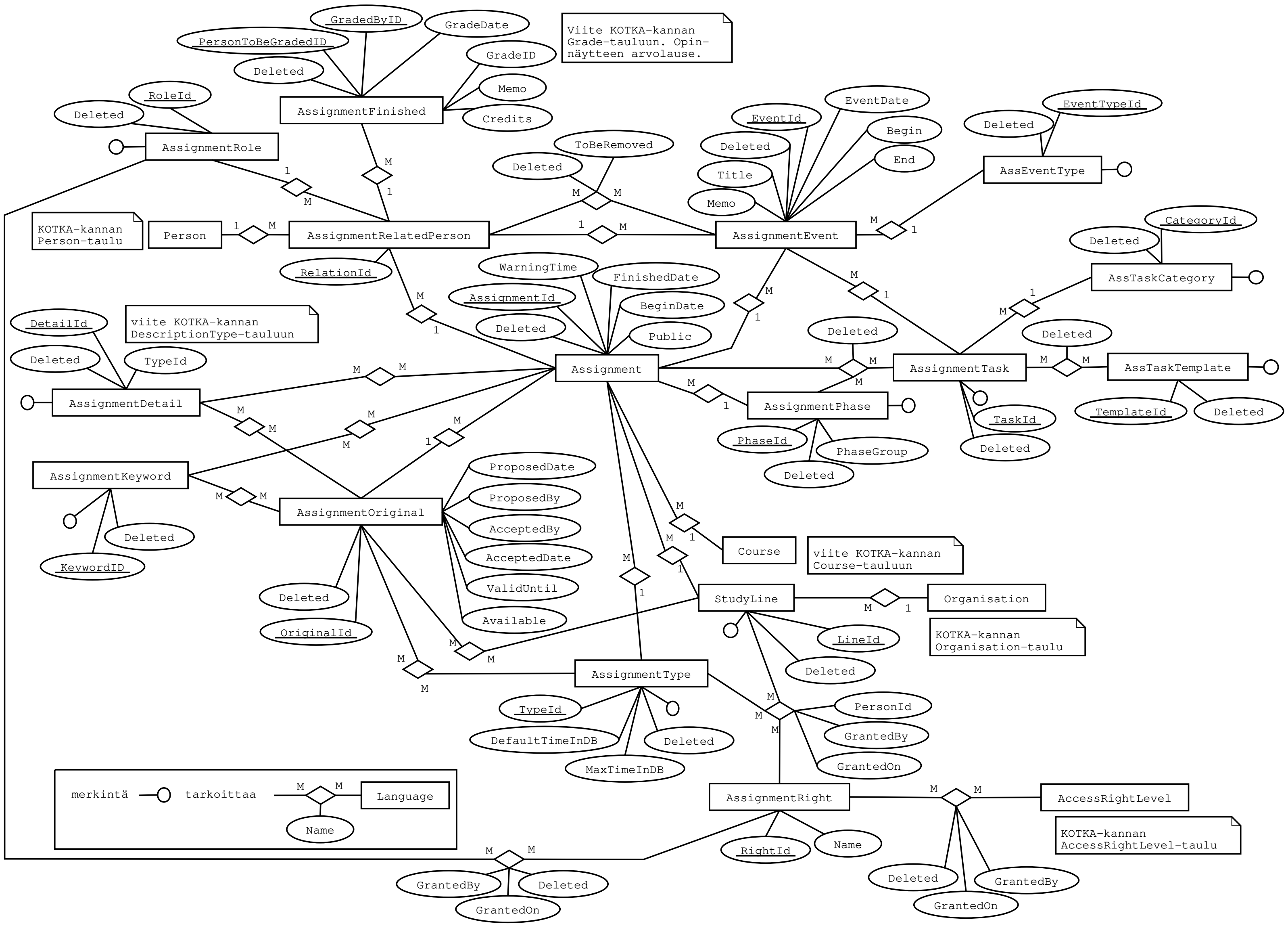
Opinnäytteiden aihe-ehdotukset ja aiheet tallennetaan kuvan 2 ER-kaaviossa esitettyyn tauluun `AssignmentOriginal`. Aiheiden otsikkotiedot, kuvaukset ja muut yleiset tiedot tallennetaan tauluun `AssignmentDetail`. Kyseisen taulun yhteys tietyn opinnäytteen aiheeseen määrätään taulussa `AssOriginalRelDetail`, joka näkyy ER-kaaviossa MtoM-suhteena taulujen `AssignmentOriginal` ja `AssignmentDetail`. Kukin tieto voidaan tallentaa kantaan monikielisesti.

Aiheeseen liittyvät tiedot on tyypitetty KOTKA-järjestelmän taulun `DescriptionType` mukaisesti. Tällöin uusien tietotyyppien lisääminen sovellukseen ei vaikuta kannan taulujen rakenteisiin millään tavalla.

Aiheeseen voidaan liittää taulussa `AssignmentKeyword` esitettyjä avainsanoja. Lisäksi aiheeseen voidaan liittää useita opinnäytetyyppejä. Opinnäytetyyppiehdotukset tallennetaan tauluun `StudyModulesPosType`, joka näkyy kaaviossa MtoM-suhteena taulujen `AssignmentOriginal` ja `AssignmentType` välillä. Työn alle otetun opinnäytteen tyypiksi valitaan jokin näistä ehdotetuista tyypeistä. Edelleen aihetta voidaan ehdottaa joillekin linjoille, jolloin linjaehdotukset tallennetaan tauluun `StudyModulesPosLine` (ER-kaavio MtoM-suhde taulujen `AssignmentOriginal` ja `StudyLine` välillä).

5.2 Opinnäyte

Työn alle otettu opinnäyte tallennetaan tauluun `Assignment` (katso kuva 2). Opinnäytteeseen liittyvät kuvaukset tallennetaan tauluun `AssignmentDetail`, ja kuvaukset liitetään opinnäytteeseen MtoM-suhteella vastaavasti kuin opinnäytteen aiheen tapauksessa. Aihe-ehdotukseen liitettyä kuvausta ei voida käyttää suoraan, sillä jos ohjaajat haluavat muuttaa yksittäisen opinnäytteen kuvaus- tai otsikkotietoja, muutos ei saa heijastua toisiin alkuperäisestä aiheesta tehtäviin opinnäytteisiin.



Kuva 2: KOPPELO-sovelluksen ER-kaavio.

Opinnäyte voidaan liittää johonkin KOTKA-kannassa olevaan kurssiin. Edelleen opinnäytteelle voidaan valita linja, jolle opinnäyte toteutetaan. Linjat on tallennettu tauluun `StudyLine`. Opinnäytteen suoritukseen liittyy vaihe, esimerkiksi `tekeillä`, `keskeytetty` ja `hyväksytty`. Vaiheet on lueteltu taulussa `AssignmentPhase`.

Opinnäytteeseen liittyvät henkilöt tallennetaan tauluun `AssignmentRelatedPerson`. Jokaiseen henkilöön liitetään rooli opinnäytteen suorituksessa (mm. opiskelija, ohjaaja tai tarkastaja). Opinnäytteeseen liittyvät tapahtumat tallennetaan kannan tauluun `AssignmentEvent`. Tapahtumaan liittyy tyyppi ja mahdollisesti työtehtävä. Tapahtuman tyypit tallennetaan tauluun `AssignmentEventType`

5.3 Työtehtävät

Jokaiseen opinnäytteeseen voidaan liittää taulun `AssignmentTask` (katso kuva 2) avulla työtehtäviä, jotka voidaan edelleen luokitella esimerkiksi suunnitteluun ja toteutukseen liittyviksi. Luokkakäsitteet on määrätty taulussa `AssTaskCategory`.

Opinnäytteen työtehtäviin voidaan liittää vaiheita, jotka on esitetty taulussa `AssignmentPhase`. Työtehtävien liittäminen opinnäytteeseen tehdään taulussa `AssTaskRelation`, joka näkyy ER-kaaviossa MtoM-suhteena taulujen `Assignment`, `AssignmentPhase` ja `AssignmentTask` välillä.

Työtehtävien liittämistä opinnäytteeseen helpottaa työmallin käsite. Työmalleihin on koottuna erilaisia työtehtäviä. Työmallit tallennetaan taulun `AssignmentTemplate`, ja malleihin liitetyt työvaiheet ovat taulussa `AssTaskTemplateRelation`, joka näkyy ER-kaaviossa MtoM-suhteena.

5.4 Arvostelu

Opinnäytteen yleinen tai koko ryhmän yhteinen arvostelu voidaan tallentaa tapahtumaksi `AssignmentEvent`-tauluun, jolloin tapahtuman tyypiksi valitaan arvostelu.

Jos jokainen opinnäytteen tekijä halutaan arvioida erikseen, tiedot näistä arvosteluista ja arvostelijasta tallennetaan tauluun `AssignmentFinished`. Tämän taulun avulla voidaan jokaiselle opinnäytteen tekijälle antaa henkilökohtainen JORE-opintosuoritusrekisteriin ilmoitettava arvosana ja opintoviikkomäärä.

5.5 Oikeudet

KOPPELO-sovellukseen liittyy käyttäjä- ja rooli- ja käyttäjäryhmäkohtaisia oikeuksia jotka tallennetaan omiin tauluihinsa. KOPPELO-sovelluksessa käytettävät oikeudet tallennetaan `AssignmentRight`-tauluun.

6 Sovelluksen toteuttaminen

Luvussa kuvataan sovelluksen JSP-sivujen totetusperiaatteita ja sovelluksen toteuttamisen kehtysprosessia. Lisäksi kuvataan sovelluksessa käytetyt Java-pavut ja JSP-Tagit.

Sovelluksen toteuttaminen aloitettiin toteuttamalla kaksi demoversiota varsinaisesta ohjelmasta. Delphi-ohjelmankehitysympäristössä toteutettiin sivuilla liikkumista mallintava sovellus ja sivujen ulkoasu mallinnettiin WWW-demossa HTML-kielellä staattisina WWW-sivuina.

6.1 JSP-sivujen koodausperiaatteista

Lopullisen sovelluksen toteuttaminen alkoi JSP-kieleen ja KOTKA-järjestelmän lähdekoodeihin tutustumisella. KOTKA-järjestelmän sivujen koodauksessa on noudatettu sitä periaatetta, että sovelluksen sivun tarvitsemat pavut otetaan mukaan jokaisella sivulla `file.jsp` erikseen ja samalla tehdään käyttäjän oikeellisuustarkastukset. Lisäksi jokaisella JSP-sivulla lisätään WWW-sivun navigointi- ja yläpalkit. Varsinainen sivun sisältö toteutetaan tiedostossa `pageFile.code`. Tämä `code`-tiedosto otetaan vastaavaan JSP-sivuun mukaaan `include`-lauseella.

KOPPELO-projekti päätti poiketa tästä yleisestä tavasta, sillä sivujen kahtiajako KOTKA-järjestelmän mallin mukaisesti vaikutti hankalalta. KOPPELO-sovelluksessa kunkin sivun sisältö oikeellisuustarkastuksineen toteutetaan yhdessä JSP-tiedostossa. Tämän koettiin olevan erillisiä tiedostoja loogisempi ja ylläpidettävämpi ratkaisu.

Java-papujen ja tagien mukaan ottamiseen tehtiin erillinen include-tiedosto `beans.inc`. KOPPELO-sovelluksen tagien tarvitsema tietokantayhteys avataan tiedostossa `dbconnect.inc`. Sovelluksen tarvitsemat kovakoodatut vakiot koottiin tiedostoon `constants.inc`. HTML-sivun aloitustagit tulostetaan tiedostossa `header.inc` ja navigointipalkkien tulostus tehdään tiedostossa `menus.inc`. Sivun päättävät HTML-tagit koottiin tiedostoon `pageEnd.inc`. Näistä kuudesta include-tiedostosta voidaan kullekin JSP-sivulle valita vain kyseisen sivun tarvitsemat elementit.

KOPPELO-sovelluksen käyttämän sivujaon etuna KOTKA-järjestelmän sivujakoon verrattuna on sivujen muokattavuus. Jos sivuille esimerkiksi tarvittaisiin kolmas navigointipalkki, riittäisi lisäys ainoastaan tiedostoon `menus.inc`. Muutos näkyisi automaattisesti kaikissa KOPPELO-sovelluksen sivuissa. Sen sijaan KOTKA-järjestelmän sivuissa jouduttaisiin muutos tekemään jokaiseen JSP-sivuun erikseen.

6.2 Käyttöliittymien erotteleminen

Oppilaan ja ohjaajan käyttöliittymiä ei ole jaettu erillisiksi JSP-sivuiksi, sillä merkittävä osa sivujen sisällöstä on sama sekä oppilaalle että ohjaajille. Tämä lisää sivujen ylläpidettävyyttä, sillä nyt sivuihin tulevat esitystapaan liittyvät muutokset (kuten sarakkeiden lisääminen taulukoihin) tarvitsee toteuttaa vain yhteen tiedostoon. Tiedon rajaaminen eri käyttäjäryhmille hoidetaan oikeuspavun avulla.

6.3 Virhetilanteiden huomioiminen

Sovelluksessa voi tapahtua useita erilaisia virhetilanteita. Tavallisin virhetilanne lienee se, että käyttäjä antaa väärässä formaatissa olevia syötteitä esimerkiksi päivämäärä-, kellonaika- tai muihin numeerisen tiedon kenttiin. Nämä virhetilanteet on etukäteen huomioitu ympäröimällä kriittiset tietokannan päivitysosat `try-catch` -lohkoilla, jolloin virhetilanteesta saadaan annettua käyttäjälle asianmukainen palaute.

Toisaalta käyttäjä saattaa yrittää päästä sellaiselle sivulle, johon hänellä ei ole käyttöoikeutta. Tällöin hänet ohjataan tilanteesta ilmoittavalle virhesivulle.

WWW-sovellusten ongelmana on käyttäjän mahdollisuus liikkua selaimen välimuistissa sovelluksen edellisille sivuille ilman, että palvelin saa tietää asiasta. Tällöin käyttäjä saattaa päästä esimerkiksi lisäämään saman tapahtuman tai opinnäytteen moneen kertaan tietokantaan. Kaikkia tällaisia tilanteita ei sovelluksen puolesta voi poissulkea. Jos tiedon kahteen kertaan lisääminen aiheuttaisi virheitä avainkenttien määrityksessä, on ajettava SQL-lause ympäröitynä `try-catch` -lohkolla. Siten käyttäjän toiminnosta aiheutuva virhe näkyy käyttäjälle selkokielisenä virheilmoitustekstinä.

6.4 Sovelluksessa käytetyt Java-pavut

Koska sovellus käyttää KORPPI-projektin luomia osia esimerkiksi navigointipalkkien tulostamiseen, käyttää sovellus välillisesti myös monia KOTKA-järjestelmään aiemmin luotuja papuja. JSP-sivuilta ei suoraan viitata kuin kolmeen Java-papuun: `DB`, `KoppeloRights` ja `time`. Näistä kaksi ensimmäistä viittausta tarvitaan käyttäjän sivunkäyttöoikeuksien määrittämiseen. `DB` on KORPPI-projektin luoma papu tietokantayhteyden hoitamiseen ja `KoppeloRights` on KOPPELO-sovelluksen luoma uusi papu oikeuksien kyselemiseen.

KORPPI-projektin tuottamaa papua `time` käytetään päivämäärämuunnosten tekemiseen. Muunnos tarvitaan päivämäärän asettamiseksi tietokannan formaatista eurooppalaiseen formaattiin, ja vastaavasti käyttäjän syöttämästä eurooppalaisesta formaatista tietokannan ymmärtämään päivämääräformaattiin. Vastaava muunnos tehdään muutamassa yhteydessä kellonajoille.

6.5 Muokatut Java-pavut

KOPPELO-sovellusta toteutettaessa jouduttiin muokkaamaan KORPPI-projektin luomaa `HTMLMenu`-papua, jotta sovelluksen vaatima navigointiosuus saatiin näkyviin olemassaolevaan navigointipalkkiin.

`HTMLMenu`-papuun lisättiin vakioita KOPPELO-sovelluksen fyysisen polun määräämiseksi. Lisäksi papuun luotiin metodit `createSupervisorAssignmentTree`, `createStudentAssignmentTree` ja `createAssignmentTree` ohjaajan, opiskelijan ja yleisen käyttäjän navigointipalkkien luomiseksi sekä metodiin `createTree` lisättiin kutsut edellä esiteltyihin metodeihin. `createTree` on näistä ainoa suoraan kutsuttava julkinen metodi. Se saa parametrikseen kutsuneen käyttäjän tiedot.

Muokatun pavun toiminta ei vieläkään ole täysin toiveiden mukainen. Navigointipalkki ei esimerkiksi muista viimeistä tilaa, jonka se viimeksi on aukaissut käyttäjälle. Jos käyttäjä etenee sovelluksessa sivulle, jota ei ole lisätty navigointipalkkiin, navigointipalkki “romahtaa kasaan” ja valittuna olevaa sivua tai sovelluksen osaa ei voi havaita navigointipalkista. Toivottavampaa olisi, että navigointipalkki säilyttäisi edellisen tilansa. KOPPELO-projektin puitteisssa tätä ominaisuutta ei kuitenkaan korjattu.

Muutetun pavun lähdekoodi on tulostettuna muiden projektiin liittyvien lähdekoodien yhteydessä.

6.6 Uudet Java-pavut

KOPPELO-sovelluksen tarpeisiin luotiin uusi Java-Papu `KoppeloRights`. Papu on apuna käyttäjän oikeuksien tarkastamisessa yksittäisen JSP-sivun suhteen. Pavun metodeilta kysytään, onko yksittäisellä käyttäjällä haluttua oikeutta joko henkilökohtaisesti tai rooli- tai käyttäjäryhmänsä perusteella. Pavun avulla JSP-sivulle tulevaa koodin määrää pystyttiin rajaamaan suunnattomasti. Lisäksi oikeuksiin liittyvä tieto on nyt kapseloitu yhden pavun sisälle. Jos käyttöoikeuden myöntämiset halutaan tehdä jollain toisella kriteerillä, riittää muutos tehdä yksittäisen pavun sisään. Muutos näkyy JSP-sivuilla automaattisesti.

`KoppeloRights`-papu tarvitsee toimiakseen KOTKA-järjestelmän `DB`-pavun instanssin tietokantayhteyden muodostamiseen. Oikeuspapua käytetään lähes kaikilla KOPPELO-sovelluksen sivuilla.

Uuden pavun lähdekoodit on tulostettuna muiden projektiin liittyvien lähdekoodien yhteydessä.

6.7 Sovelluksessa käytetyt JSP-Tagit

JSP-sivuilla on voitu vähentää suoria Java-luokkien kutsuja käyttämällä JSP-Tageja. Tageja käytetään kahdessa yhteydessä: tietokantayhteyksien hallintaan sekä staattisen tekstin kääntämiseen.

KOPPELO-sovellus käyttää Jakarta-projektin tuottamaa SQL-tagia tietokantayhteyden hallintaan. Tagi ottaa tietokantayhteyden kunkin sivun alussa ja vapauttaa yhteyden sivun lopussa. Projektiryhmä koki tietokantatagin käytön miellyttävämmäksi ja helppokäyttöisemmäksi kuin KORPPI-projektin tuottaman `DB`-pavun. Lisäksi JSP-sivujen koodi siistiytyi tagin ansiosta huomattavasti.

Lisätietoja Jakarta-projektista ja SQL-tagista saa osoitteesta `http://jakarta.apache.org/taglibs/doc/dbtags-doc/`

Staattisen tekstin kääntämiseen käytetään Heikki Uuksulaisen tekemää käännös-tagia. Tagin ansiosta projektiryhmän ei tarvinnut miettiä mm. staattisten otsikkoteksien monikielistämistä sovelluksen kehittämisen aikana, vaan tagin on tarkoitus palauttaa käyttöliittymän tekstit käyttäjälle hänen määräämällään kielellä. Käännöstagi on tarkoitus ottaa myöhemmin käyttöön koko KOTKA-järjestelmässä.

7 Sivustokartat

Luvussa esitellään KOPPELO-sovelluksen sivustorakennetta sekä sivujen keskinäisiä suhteita. KOPPELO-sovelluksessa ei ole toteutettu erillisiä JSP-sivuja opiskelijoita ja ohjaajia varten, vaan kutsut menevät yhteisille JSP-sivuille. Sivuilla oleva oikeuspapu rajaa sivulle pääsyä ja tiedon näyttämistä yksittäisen JSP-sivun sisällä tietokannassa määrättyjen oikeuksien pohjalta.

7.1 Sivustokarttojen tulkitsemisesta

Sivustokartoissa (kuvat 3–6) tummennetulla pohjalla esitetyt JSP-sivut ovat käyttäjälle näkyviä sivuja. Vaalealla pohjalla olevat sivut päivittävät tietokantaa, jonka jälkeen suoritus siirtyy automaattisesti seuraavalle sivulle.

Suorat nuolet sivukartoissa esitettyjen JSP-sivujen välillä kertovat, että sivulta on linkki nuolen päässä esitetylle sivulle. Kaarinuoli sivujen välillä kertoo, että sivulla esitetään käyttäjälle mahdollisuus palata takaisin edelliselle sivulle.

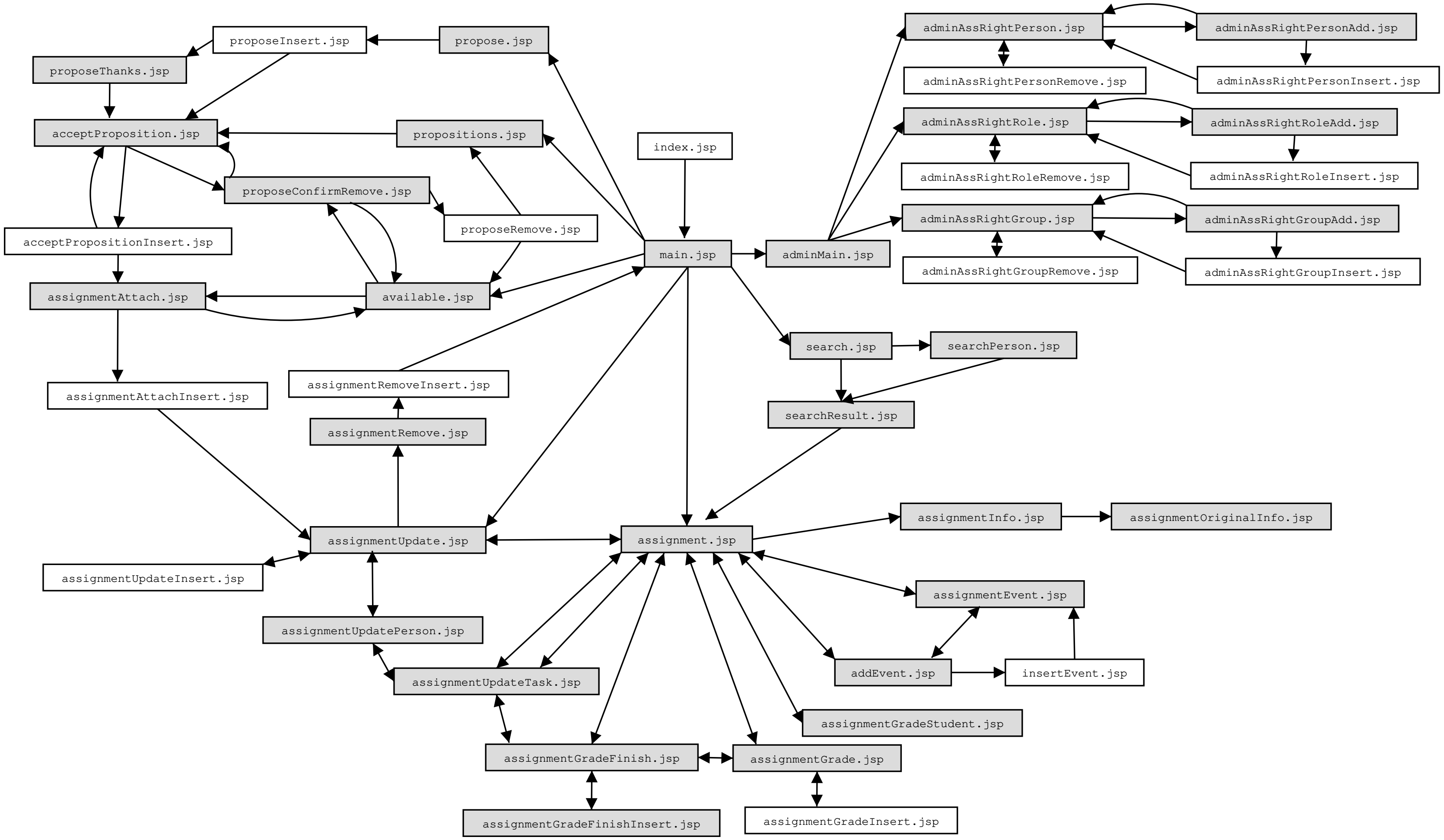
Opinnäytteen tietojen hallinnan kokonaisuuden muodostavien sivujen `assignment.jsp`, `assignmentUpdate.jsp`, `assignmentUpdatePerson.jsp`, `assignmentUpdateTask.jsp`, `assignmentGrade.jsp`, `assignmentGradeFinish.jsp` ja `addEvent.jsp` väleille ei ole piirretty kaikkia sivujen välisiä suhteita kaavion selkeyttämiseksi. Kaikilta edellä mainituilta sivuilta on siis linkit toisiinsa. Lisäksi missään sivukartoissa ei ole eritelty käyttäjän mahdollisuuksia siirtyä sivulta toiselle sovelluksen navigointipalkin avulla.

Sivustokartoista on jätetty piirtämättä suhteet virhesivuihin. Käytännössä jokaisen sivun suorituksessa voi tapahtua virheitä. Niistä yleisimmät lienevät tapaukset, joissa käyttäjä on syöttänyt tietoa virheellisessä muodossa tai yrittää päästä (esimerkiksi suoralla osoitteella) sellaiselle sivulle, johon hänellä ei ole käyttöoikeutta.

7.2 KOPPELO-sovelluksen sivustokartta

Kuvassa 3 on esitetty koko KOPPELO-sovelluksen sivustokartta. Sivustokartta esittää sivujen väliset suhteet eli kuinka sivuilta on linkkejä toisiinsa ja kuinka sivuilla voi liikkua kaikilla käyttöoikeustasoilla. Tilanpuutteen vuoksi kyseisessä sivustokartassa on esitetty ainoastaan JSP-tiedostojen nimet.

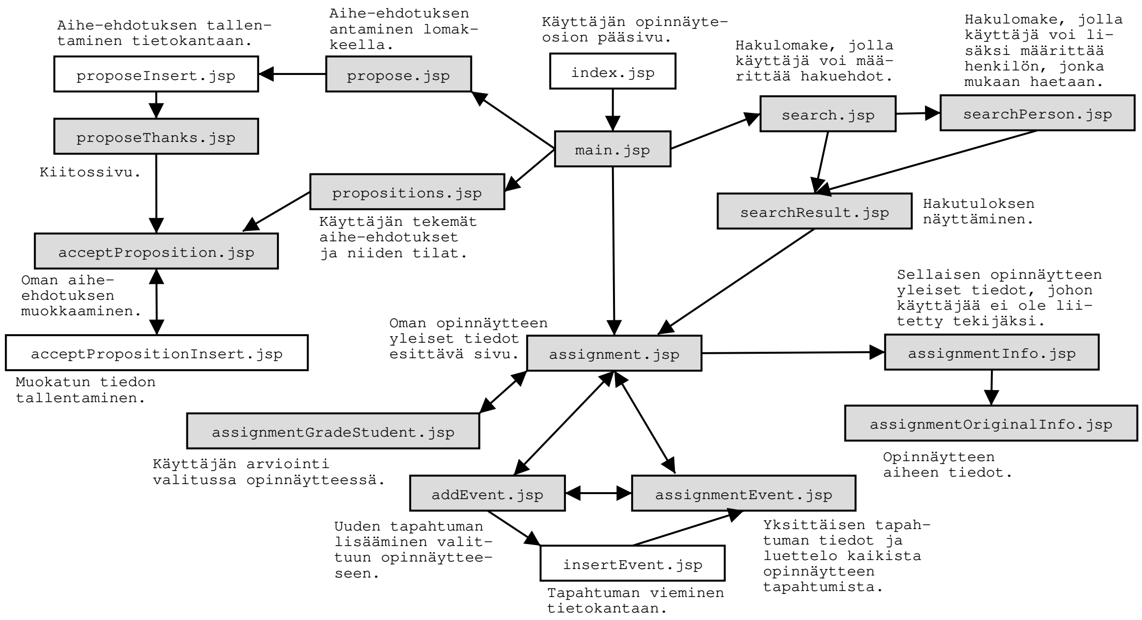
Sivustokartta on jaettu pienempiin kokonaisuuksiin lukujen 7.3–7.5 kuvissa 4–6. Näissä sivustokartoissa on kunkin tiedoston nimen yhteydessä myös lyhyt kuvaus sivun käyttötarkoituksesta. Kattavammat kuvaukset sivujen sisällöistä löytyvät luvuista 9 – 13.



Kuva 3: KOPPELO-sovelluksen sivustokartta.

7.3 Opiskelijan sivustokartta

Kuvan 4 opiskelijan sivustokartassa on esitetty vain ne sivut, joita opiskelijalla on oikeus käyttää KOPPELO-sovelluksessa. Sivustokartta perustuu siihen oletukseen, että käyttäjäryhmälle `opiskelija` on asetettu normaali opinnäytteen tietojen katseluoikeus ja käyttäjälle ei ole asetettu muita KOPPELO-sovelluksen erityisoikeuksia.

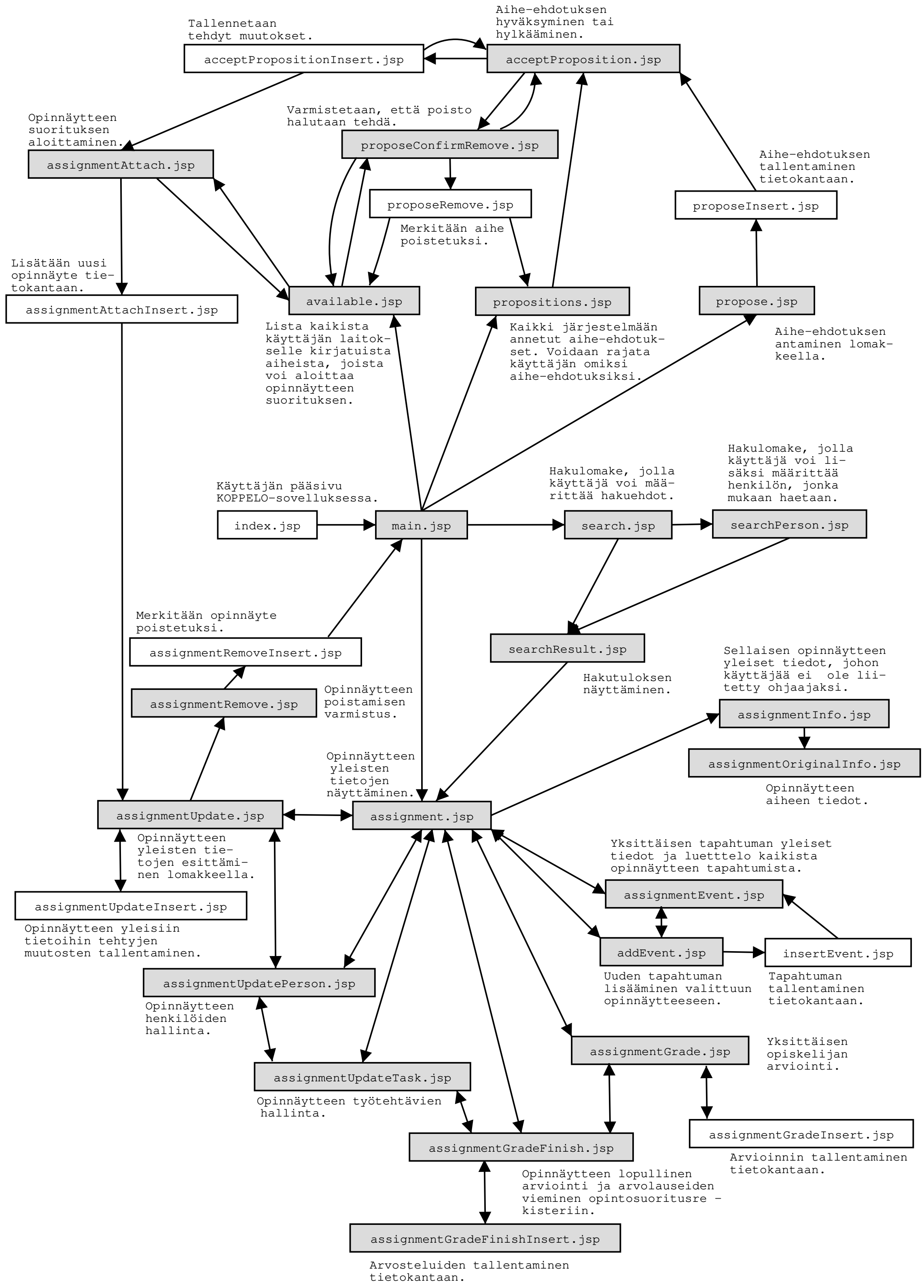


Kuva 4: Opiskelijan sivustokartta KOPPELO-sovelluksessa.

Opiskelijan sivustokartassa voidaan erottaa kolme osa-aluetta. Aiheen ehdottamiseen liittyvät sivut `propositions.jsp`, `propose.jsp`, `proposeInsert.jsp`, `proposeThanks.jsp`, `acceptProposition.jsp` ja `acceptPropositionInsert.jsp` esitellään luvuissa 9.1–9.6. Opinnäytteen tietoihin liittyvät sivut `assignment.jsp`, `addEvent.jsp`, `insertEvent.jsp`, `assignmentEvent.jsp` ja `assigmentGradeStudent.jsp` esitellään luvuissa 10.12–10.14, 10.11 ja 11.5. Hakusivut `search.jsp`, `searchPerson.jsp` ja `searchResult.jsp` esitellään luvuissa 11.2–11.4. Näiden osa-alueiden lisäksi on käyttäjän henkilökohtainen opinnäyteosan pääsivu (`index.jsp`, joka esitellään luvussa 11.1).

7.4 Ohjaajan sivustokartta

Kuvan 5 sivustokartassa ohjaaja käsitetään henkilöksi, jolle on asetettu opinnäytteeseen liittyen kaikki mahdolliset oikeudet. KOPPELO-sovellus mahdollistaa ohjaajan sivustokokonaisuuden jakamisen edelleen pienempiin osiin, mutta kokonaisuuden hahmottamiseksi ohjaajan sivustokartta on esitetty tässä yhteydessä täydellisenä.



Kuva 5: Ohjaajan sivustokartta KOPPELO-sovelluksessa.

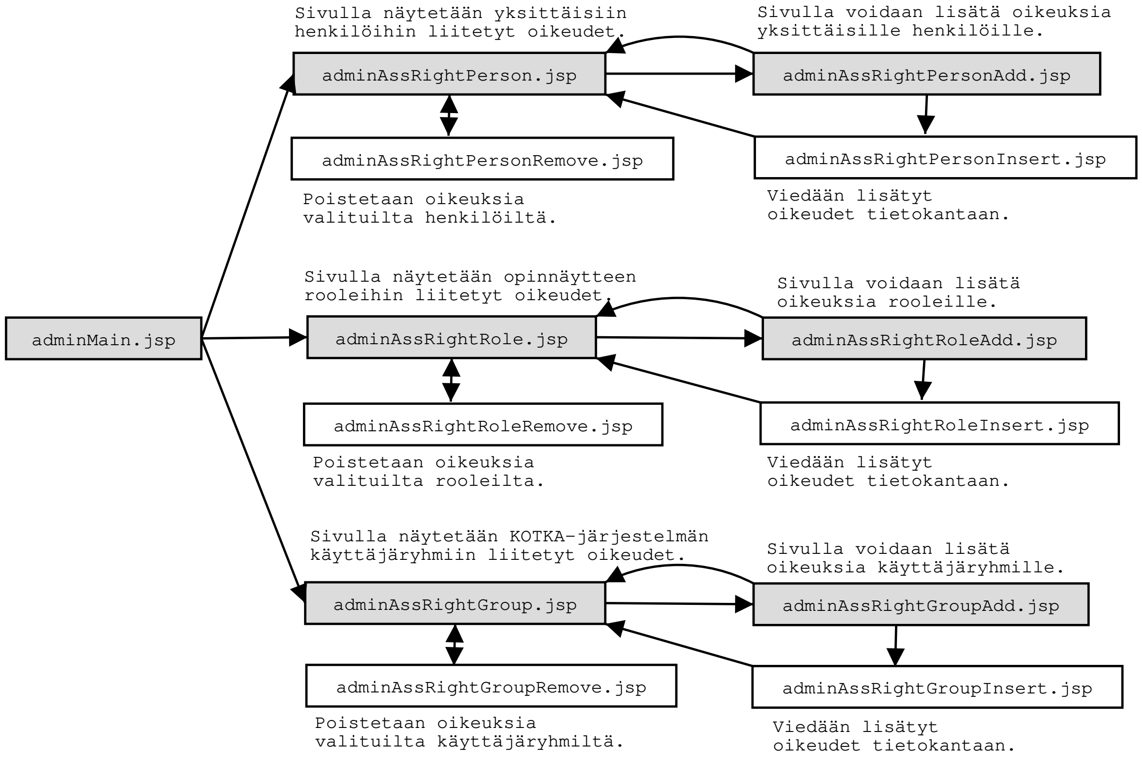
Oikeudet voitaisiin jakaa edelleen opinnäytteen aihe-ehdotusten hyväksyjään, opinnäytteen kiinnittäjään, vastaavaan ohjaajaan, henkilöiden kiinnittäjään, arvostelijaan ja opinnäytteen hyväksyjään. Tällöin kullekin oikeustasolle näkyisi vain muutama yksittäinen sivu kuvan 5 sivustokartasta.

Ohjaajan sivustokartta jakautuu kolmeen jollain tavalla toisistaan eroteltavaan osa-alueensa: aihe-ehdotusten käsittelyyn, opinnäytteen tietojen hallintaan ja hakuun. Osa-alueisiin jako vastaa luvuissa 9–11 esitettyä sivujakoa.

Sovelluksen sivuille on laadittu erillinen käyttöohje, joka on liitetty projektikansioon.

7.5 Ylläpidon sivustokartta

Kuvan 6 sivustokartassa on esitetty KOPPELO-projektin puitteissa ylläpidon käyttöliittymään toteutetut JSP-sivut ja niiden suhteet toisiinsa. Sivustokartan esittämät JSP-sivut on kuvattu tarkemmin luvuissa 13.1–13.13. Ylläpidon osaan kuuluvat sivut tunnistaa sivujen nimiin liittyvästä `admin`-tunnisteesta.



Kuva 6: Ylläpitäjän sivustokartta KOPPELO-sovelluksessa.

8 Sivujen käyttöliittymän yleisrakenne

Luvussa kuvataan yleisellä tasolla KOPPELO-sovellukseen toteutettua käyttöliittymää. Sovellukselle on kirjoitettu erillinen käyttöohje, jossa perehdytään tarkemmin yksittäisten sivujen sisältämiin tietoihin ja toiminnallisuuteen. Sivujen teknisiä toteutuksia on esitetty tämän dokumentin luvuissa 9–14.

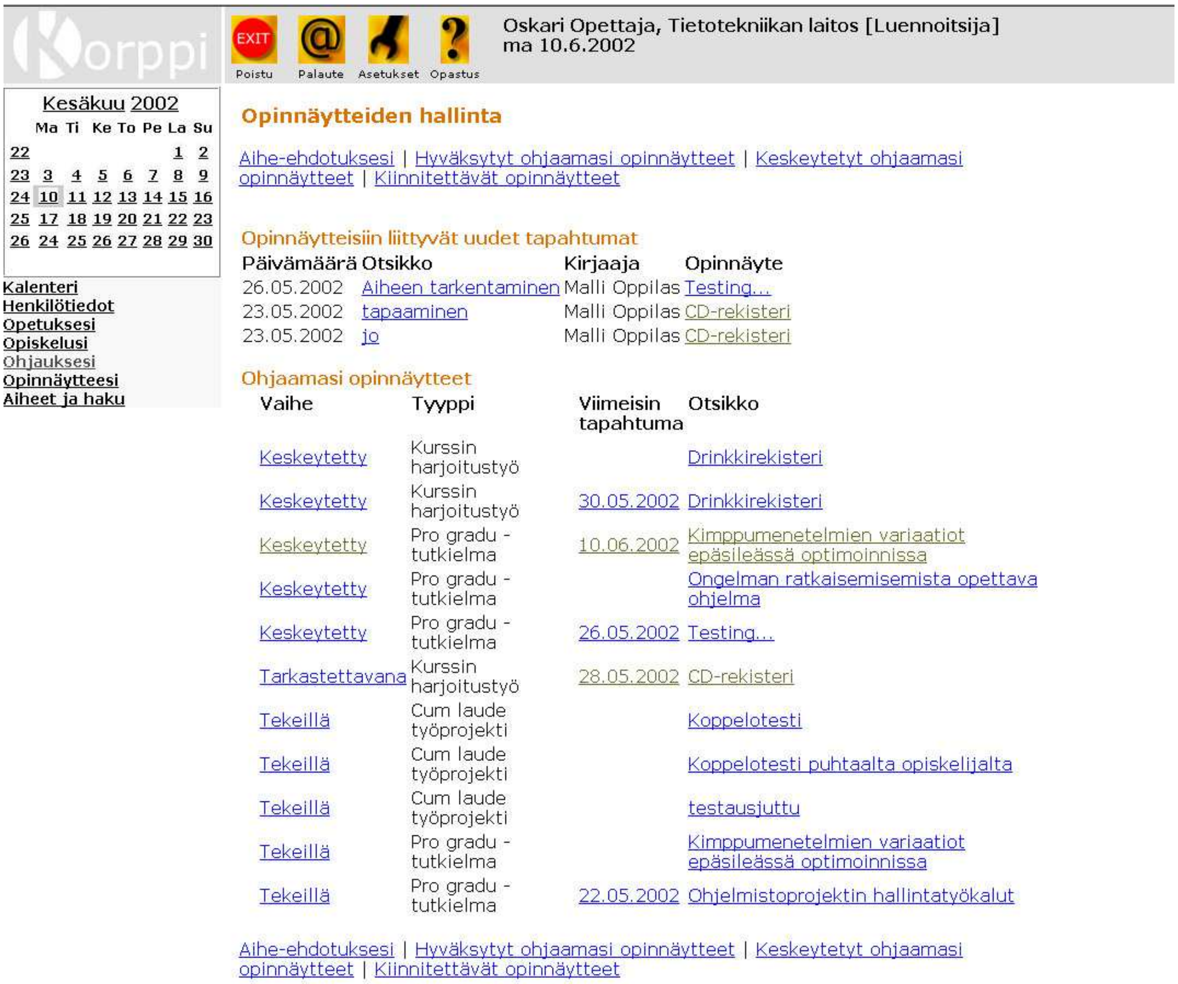
KOPPELO-sovellukseen toteutettiin käyttöliittymät opiskelijoille, ohjaajille ja ylläpidolle. Käyttöliittymien ulkoasut pidettiin yhtenäisinä KOTKA-järjestelmän ulkoasun kanssa. Sovelluksen käyttäjä ei siten tiedä käyttävänsä erillistä moduulia, vaan hänelle KOPPELO-sovellus on yhtenäinen osa KOTKA-järjestelmää.

8.1 Käyttöliittymän perusnäkymä

Kuvassa 7 on esitetty ohjaajan käyttöliittymän mukainen KOPPELO-sovelluksen pääsivu. Kuva antaa viitteitä siihen, millainen on KOTKA-järjestelmän ja samalla KOPPELO-sovelluksen perusrakenne. Sivun ylälaidassa on KORPPI-sovelluksen logo, jonka vasemmalla puolella on linkkejä yleisiin toimintoihin (järjestelmästä uloskirjautuminen, palautteen antaminen, omien asetusten muuttaminen ja avuste) sekä käyttäjän nimi ja käyttäjätaso KORPPI-sovelluksessa.

KORPPI-logon alapuolella näkyy kalenteri, joka on samalla linkki KOLIBRI-sovellukseen eli henkilökohtaiseen kalenteriin. Kalenterin alla on navigointipalkki, joka auttaa hahmottamaan avatun sivun sijoittumista järjestelmäkokonaisuuteen ja mahdollistaa sujuvamman siirtymisen järjestelmän sivuilta toiselle. Jäljelle jäävä sivun osa on varattu kunkin sivun omalle toiminnallisuudelle ja tietojen esittämiselle.

Lisää sovelluksen käyttöliittymän kuvia on esitetty lukujen 9–13 yhteydessä.



Kuva 7: Ohjaajan opinnäytteiden hallintasivu.

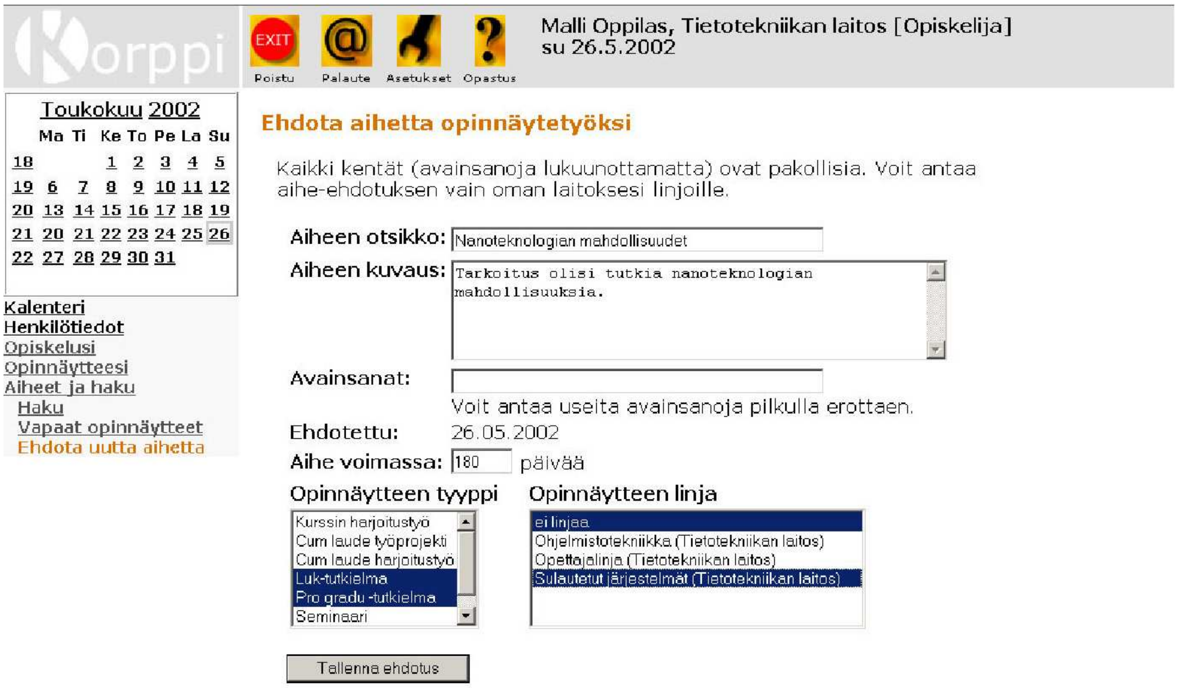
9 Sivut aihe-ehdotuksesta opinnäytteen aiheeksi

Luvussa kuvataan, kuinka opinnäytteen aihe-ehdotus käsitellään kaikkien mahdollisten välivaiheidensa kautta hyväksytyksi opinnäytteen aiheeksi. Osion JSP-sivut tunnistaa muista KOPPELO-sovelluksen sivuista siitä, että niiden nimen osana esiintyy sana `propose` jossain muodossa.

Sivujen näkyvyys riippuu käyttäjille asetetuista oikeuksista. Osa sivuista näkyy vain opiskelijan sivustokartassa (kuva 4), mutta lähes kaikki sivut on kuvattu ohjaajan sivustokartan yläosassa (kuva 5).

9.1 propose.jsp

Aihe-ehdotus annetaan sivulla `propose.jsp`. Sivun käyttöliittymä on esitetty kuvassa 8. Sivulla on lomake aihe-ehdotuksen tietojen täyttämiselle. Tietojen täyttämistä ei valvota tällä sivulla esimerkiksi JavaScriptien avulla, vaan tietojen oikeellisuuden tarkastus jätetään seuraavien sivujen huoleksi. Lomakkeen lähettäminen aiheuttaa sivun `proposeInsert.jsp` (luku 9.2) kutsun.



Kuva 8: Aiheen ehdottamisen sivunäkymä.

Sivu `propose.jsp` ei tarvitse parametreja. Ainoa käyttöoikeustarkastus on, että sivun käyttäjä on kirjautunut KOTKA-järjestelmään. Tämä vaatimus on välttämätön, sillä aiheen ehdottaja tunnistetaan KOTKA-järjestelmän antaman id-tunnuksen perusteella.

9.2 proposeInsert.jsp

Sivu `proposeInsert.jsp` ei näy käyttäjälle millään tavalla, sillä sivun ainoa tarkoitus on tallentaa käyttäjän antamat arvot tietokantaan. Epäonnistuessaan se vie sivulle `error.jsp` ja onnistuessaan se ohjaa käyttäjän oikeuksien mukaan joko sivulle `proposeThanks.jsp` (luku 9.3) tai `acceptProposition.jsp` (luku 9.5). Sivuille viedään parametrina tietokantaan lisätyn aihe-ehdotuksen id.

Sivun ainoa käyttöoikeustarkastus on, että käyttäjä on kirjautunut KOTKA-järjestelmään. Sivun vaatimat parametrit on esitelty alla:

`title`	on tallennettavan aiheen otsikko, joka ei saa olla `null`.
`description`	on tallennettavan aiheen kuvaus, joka ei saa olla `null`. Parametri voi kuitenkin sisältää tyhjän merkkijonon.
`proposeddate`	ilmoittaa päivämäärän, jolloin aihe-ehdotus on tehty. Arvo ei saa olla `null`.
`validdays`	ilmaisee, kuinka kauan aihe-ehdotusta säilytetään tietokannassa. Arvon täytyy olla numeerinen.
`line`	kertoo aihe-ehdotukseen liitetyt linjat. Parametri voi olla moniarvoinen tai se voi sisältää arvon `null`. Parametri ei saa kuitenkaan puuttua.
`type`	kertoo aihe-ehdotuksen mahdolliset opinnäytetyypit. Parametri voi olla moniarvoinen.
`keywords`	sisältää mahdolliset avainsanat. Parametri voi puuttua.

Sivun suorittaminen aiheuttaa automaattisen sähköpostiviestin lähettämisen aihe-ehdotuksen hyväksymisestä vastaaville henkilöille ehdotetuilla linjoilla ja opinnäytetyypeillä. Viestin sisältö kootaan parametrien arvoista sekä linkistä sivulle, jolla vastaanottaja voi käydä hyväksymässä aihe-ehdotuksen (`acceptProposition.jsp`, katso luku 9.5).

9.3 proposeThanks.jsp

Sivulla kiitetään käyttäjään hänen antamastaan aihe-ehdotuksesta ja tarjotaan linkki sivulle `acceptProposition.jsp` (luku 9.5), jossa hän voi vielä käydä muuttamassa aihe-ehdotuksensa tietoja.

Sivun käyttöoikeus on kaikilla KOTKA-järjestelmän käyttäjillä. Sivu vaatii parametrin `origid`, eli sen aihe-ehdotuksen id-numeron, jonka käyttäjä on juuri lisännyt tietokantaan.

9.4 propositions.jsp

Annetusta parametreista riippuen sivulla `propositions.jsp` näytetään joko kaikki tietokannassa olevat käyttäjän laitoksen aihe-ehdotukset tai vain käyttäjän itse tekemät aihe-ehdotukset (ei aiheeksi hyväksyttyjä ehdotuksia). Parametrin `own` asettaminen rajaa aihe-ehdotukset käyttäjän omiin aihe-ehdotuksiin. Sivulla ei käytetä erityisiä oikeustasotarkastuksia. Sivun käyttöliittymä on esitetty kuvassa 9.

Kuva 9: Omien aihe-ehdotusten sivunäkymä.

Sivulla luetellut aihe-ehdotukset ovat linkkejä sivulle `acceptProposition.jsp` (luku 9.5).

9.5 acceptProposition.jsp

Sivulla `acceptProposition.jsp` tarkastetaan, että sivun käyttäjä on joko aihe-ehdotuksen antaja tai henkilö, jolla on oikeus hyväksyä aihe-ehdotus jollekin aihe-ehdotukseen asetetuista mahdollisista linjoista ja opinnäytetyypeistä. Muutoin käyttäjä ohjataan sivulle `assignmentOriginalInfo.jsp` (luku 12.2). Sivun käyttöliittymä on esitetty kuvassa 10.

Kuva 10: Aihe-ehdotuksen muokkaamisen ja hyväksymisen sivunäkymä.

Jotta sovellus tietää, minkä aihe-ehdotuksen tietoja täytyy muokata, vaatii sivu parametrin `origid`, joka on käsiteltävän aihe-ehdotuksen id-numero.

Käyttäjälle näytetään eri toimintoja riippuen siitä, onko hän aihe-ehdotuksen antaja vai hyväksyjä. Aiheen ehdottaja voi ainoastaan muokata ehdotuksensa tietoja. Tällöin muutetut tiedot lähetetään sivulle `acceptPropositionInsert.jsp` (luku 9.6) parametrilla `update`. Hyväksyjälle näkyy kolme lisätoimintoa: sähköpostiviestin lähettäminen aiheen ehdottajalle, aiheen hylkääminen ja aiheen hyväksyminen. Nämä kaikki lähettävät viestin samalle sivulle `acceptPropositionInsert.jsp`. Toiminto erotellaan sivun lähettämällä parametrilla.

9.6 acceptPropositionInsert.jsp

Sivulla `acceptPropositionInsert.jsp` päivitetään aihe-ehdotuksen tietoja, lähetetään sähköpostiviesti aihe-ehdotuksen tekijälle sekä hyväksytään tai hylätään aihe-ehdotus. Tietojen päivitys tehdään aina sivua kutsuttaessa, mutta muu toiminta riippuu sivun saamista parametreista.

Sivun käyttöoikeus on vain aiheen ehdottajalla ja aihe-ehdotusten hyväksyjillä. Sivun tarvitsemat parametrit on selitetty alla. Puutteellinen parametrilista aiheuttaa sivun `error.jsp` kutsumisen.

`origid`	on muokattavan aihe-ehdotuksen id, joka ei saa olla `null`.
`title`	on tallennettava otsikko. Jos parametria ei ole annettu, tulkitaan otsikko tyhjäksi merkkijonoksi.
`description`	on tallennettava kuvaus. Jos parametria ei ole annettu, tulkitaan kuvaus tyhjäksi merkkijonoksi.
`validuntil`	ilmaisee päivämäärän, johon saakka aihe-ehdotus säilytetään tietokannassa. Arvon täytyy olla päivämäärätyyppinen. Annetun päivämäärän jälkeen aihe merkitään poistetuksi tietokannasta.
`line`	kertoo aihe-ehdotukseen liitetyt linjat. Parametri voi olla moniarvoinen tai se voi sisältää arvon `null`. Parametri ei saa kuitenkaan puuttua.
`type`	kertoo aihe-ehdotuksen mahdolliset opinnäytetyypit. Parametri voi olla moniarvoinen.
`keywords`	sisältää mahdolliset avainsanat. Parametri voi puuttua.
`message`	sisältää mahdollisesti lähetettävän sähköpostiviestin. Parametri voi puuttua.
`update`	ilmaisee, että tiedot halutaan vain tallentaa tietokantaan. Parametri ohjaa käyttäjän tietojen tallentamisen jälkeen takaisin sivulle `acceptProposition.jsp` (luku 9.5).
`email`	ilmaisee, että `message`-parametrissa oleva arvo pitää lähettää aiheen ehdottajalle. Toiminto aiheuttaa sähköpostiviestin lähettämisen.
`accept`	kertoo, että aihe-ehdotus hyväksytään eli tiedot tallennetaan ja aiheen hyväksyjäksi asetetaan käyttäjän id. Käyttäjä ohjataan sivulle `assignmentAttach.jsp` (luku 10.1).
`fail`	kertoo, että kyseinen aihe-ehdotus halutaan merkitä poistetuksi tietokannasta. Parametrin antaminen ohjaa käyttäjän sivulle `proposeConfirmRemove.jsp` (luku 9.7).

Neljä viimeistä parametria `update`, `email`, `accept` ja `fail` siis määräävät, millainen toiminto kyseisellä sivulla tehdään. Vain yksi näistä toiminnoista voidaan suorittaa kerrallaan, sillä kunkin toiminnon suorittaminen ohjaa käyttäjän uudelle KOPPELO-sovelluksen sivulle.

9.7 proposeConfirmRemove.jsp

Sivulla varmistetaan, haluaako käyttäjä merkitä yksittäisen aihe-ehdotuksen tai aiheen poistetuksi KOPPELO-sovelluksesta. Sivun käyttöoikeus on kaikilla niillä henkilöillä, joilla olisi oikeus hyväksyä kyseinen aihe-ehdotus opinnäytteen aiheeksi. Mikäli käyttäjä haluaa poistaa aihe-ehdotuksen, hänet ohjataan sivulle `proposeRemove` (luku 9.8). Käyttäjä voi myös valita edelliselle sivulle takaisin vievän toiminnon. Edellinen sivu määritellään sivulle annetusta parametrista.

Sivu vaatii parametrikseen poistettavan aiheen id-numeron nimellä `origid`. Lisäksi sivulle voi antaa parametrin, jota käytetään sivulla takaisin linkkinä. Tämä parametri annetaan nimellä `page`. Ellei parametria ole annettu, käytetään oletuksena sivua `acceptProposition.jsp` (luku 9.5).

9.8 proposeRemove.jsp

Sivulla merkitään aihe tai aihe-ehdotus poistetuksi tietokannasta. Sivun käyttöoikeus on niillä henkilöillä, joilla olisi oikeus hyväksyä vastaava aihe-ehdotus aiheeksi.

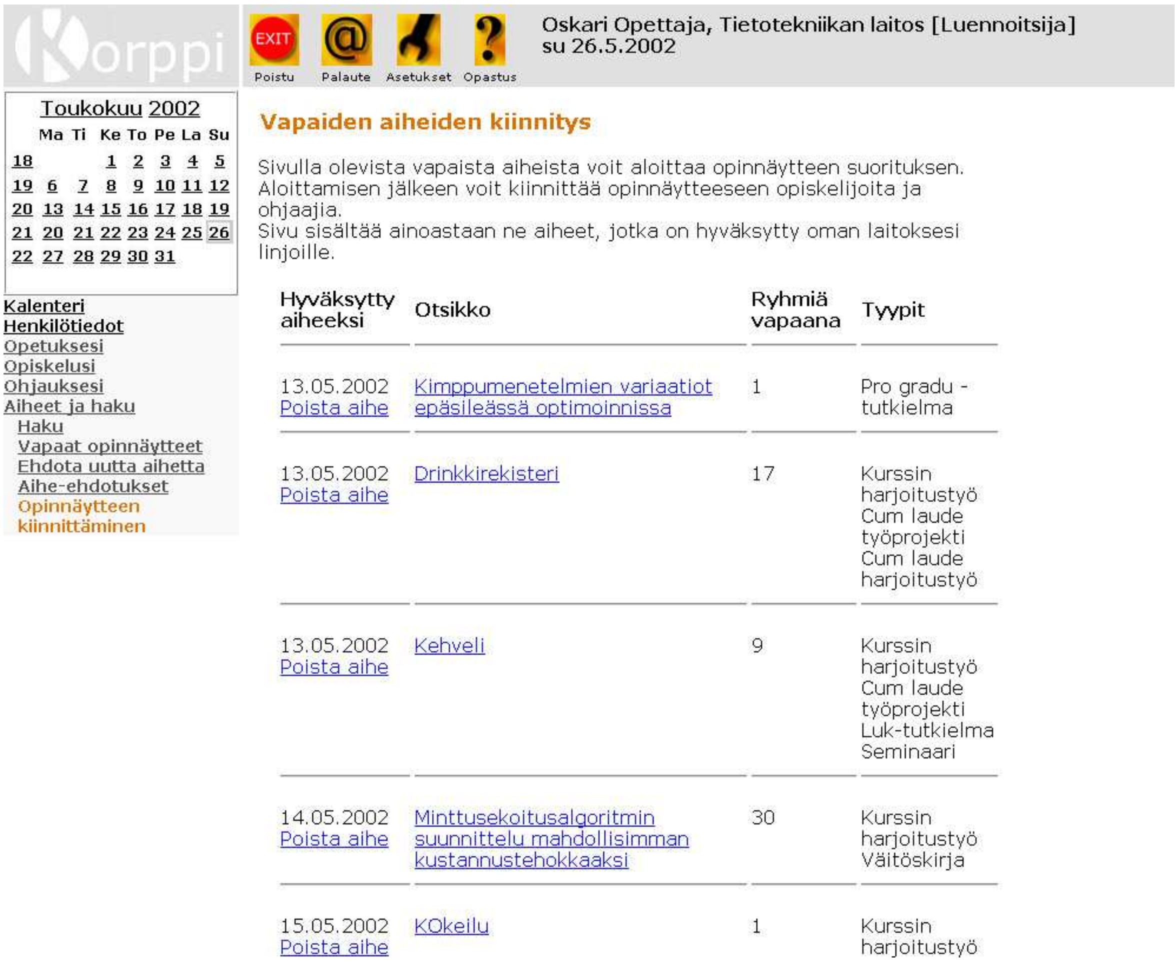
Sivun pakollisena parametrina on `origid`, joka ilmaisee poistettavan aiheen id:n. Lisäksi tarvitaan parametri `redirect`, joka ilmaisee mille sivulle käyttäjä palautetaan poistamisen jälkeen. Sivun kutsuminen lähettää automaattisen sähköpostiviestin aiheen ehdottajalle. Viestissä ilmoitetaan, että aihe tai aihe-ehdotus on hylätty.

9.9 available.jsp

Sivulla `available.jsp` esitetään kaikki ne opinnäytteiden aiheet, joista voi aloittaa opinnäytteen suorituksen. Sivun katseluoikeus on kaikilla KOTKA-järjestelmän opettaja-tasoisilla käyttäjillä, joilla on oikeus aloittaa yksittäisen opinnäytteen suoritus. Sivu ei vaadi parametreja. Sivun näkymä on esitetty kuvassa 11.

Sivulla esitetään kunkin aiheen yhteydessä linkki sivulle `assignmentAttach.jsp` (luku 10.1), jossa käyttäjä voi aloittaa opinnäytteen suorituksen. Lisäksi sivulla on linkit sivulle `proposeConfirmRemove.jsp` (luku 9.7), jossa käyttäjä voi poistaa aiheen sovelluksesta.

Sivulle `assignmentAttach.jsp` vievään linkkiin liittyvä parametri `origid` ilmoittaa, mistä aiheesta opinnäytteen suoritus halutaan aloittaa. Sama parametri viedään myös sivulle `proposeConfirmRemove.jsp`. Lisäksi poistamissivulle viedään parametrina `page` tämän JSP-sivun nimi.



Kuva 11: Varattavissa olevien aiheiden sivunäkymä.

10 Opinnäytteen tietojen hallinta

Luvussa esitetään opinnäytteen tietojen hallintaan liittyviä sivuja. Sivujen käyttöoikeudet tulisi olla liitetty lähinnä rooleihin tai yksittäisiin henkilöihin. KOTKA-järjestelmän käyttäjäryhmätasoihin liitetyt oikeudet ovat liian laajoja opinnäytteen tietojen hallintaan.

Luvussa esitetyt sivut näkyvät ohjaajan käyttöliittymän sivustokartan (kuva 5) alaosassa. Luvuissa 10.11–10.14 esitetyt sivut näkyvät myös opinnäytteeseen liitetyille opiskelijoille.

10.1 assignmentAttach.jsp

Sivulla `assignmentAttach.jsp` voidaan aloittaa opinnäytteen suoritus. Sivun näkymä on esitetty kuvassa 12. Kaikki lomakkeella kysytyt tiedot täytyy täyttää. Tiedot lähetetään edelleen sivulle `assignmentAttachInsert.jsp`.

Kuva 12: Opinnäytteen aloittamisen sivunäkymä KOPPELO-sovelluksessa.

Sivun vaatima parametri `origid` ilmaisee, aiheen, josta opinnäyte aiotaan aloittaa. Sivulle pääseminen edellyttää, että käyttäjälle on annettu oikeus kiinnittää opinnäytteitä.

10.2 assignmentAttachInsert.jsp

Opinnäytteen aloittaminen luo uuden rivin tietokannan tauluun `Assignment`. Tämä tehdään sivulla `assignmentAttachInsert.jsp`. Sivu vaatii seuraavia parametreja:

`origid`	on opinnäytteeksi aloitettavan aiheen id, joka ei saa olla `null`.
`lineid`	on aloitetun opinnäytteen linjan. Parametri voi sisältää arvon `null`. Parametri ei saa kuitenkaan puuttua.
`typeid`	kertoo aiheeseen liitettävän opinnäytetyypin id:n. Parametri ei saa puuttua.
`courseid`	sisältää aiheeseen liitettävän kurssin id:n. Parametri ei saa puuttua.
`begindate`	sisältää aiheen aloittamispäivämäärän. Parametri ei saa puuttua.

Sivun käyttäminen vaatii, että käyttäjällä on oikeus aloittaa opinnäyte parametrin mukaisella linjalla ja opinnäytetyyppillä. Lisäksi sivulla tarkastetaan, että parametrin mukaisesta opinnäytteen aiheesta saa todella aloittaa opinnäytteen suorituksen. Tarkastus on välttämätöntä tehdä mm. siksi, että muutoin käyttäjä voisi selaimen *Back*-toimintoa hyväksikäyttäen tahattomasti aloittaa lukuisia opinnäytteitä samasta aiheesta.

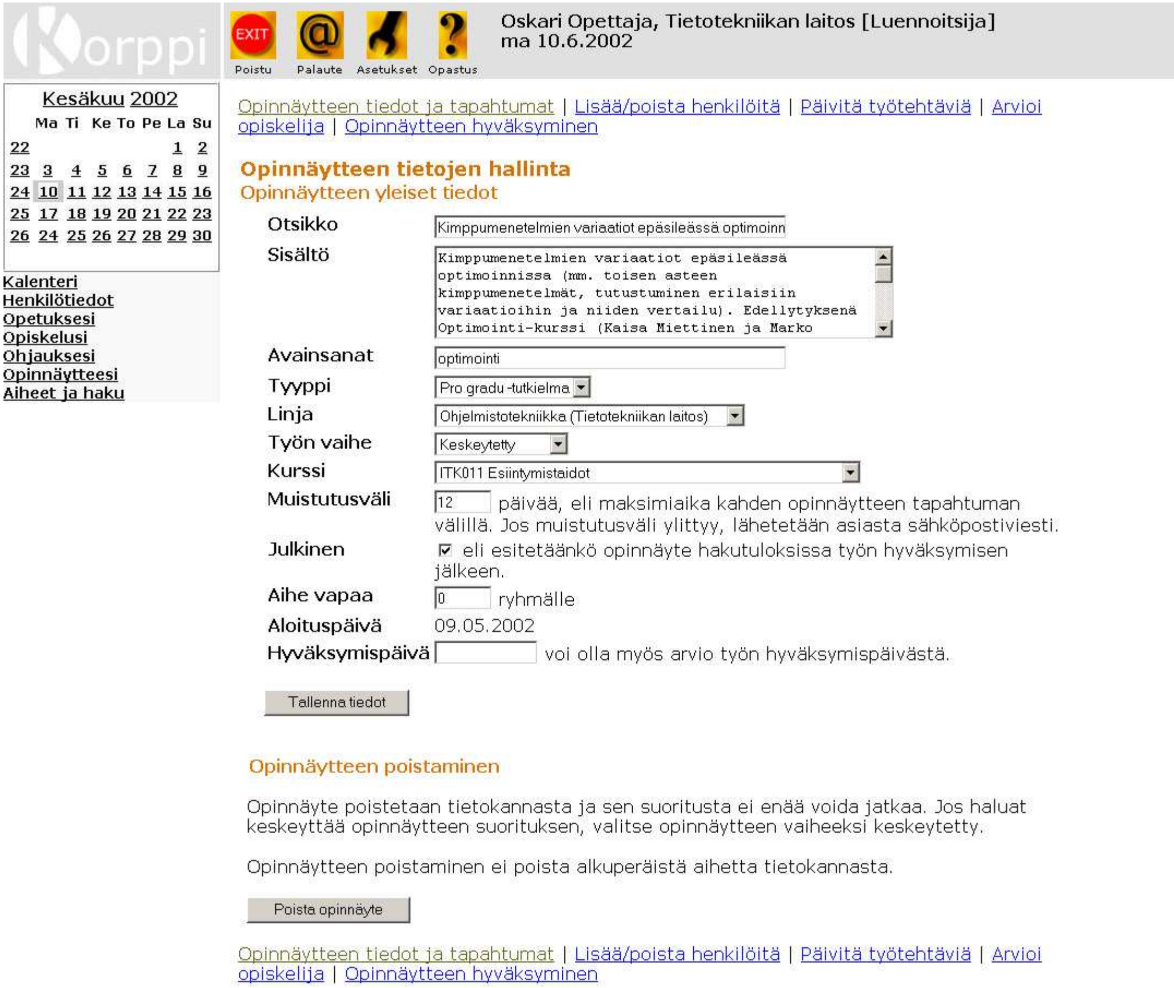
Tietokantaan viedään parametrien mukaiset arvot. Opinnäytteeseen liitetään aihetta vastanneet otsikko-, kuvaus- ja avainsanatiedot. Sivulle tullut käyttäjä liitetään opinnäytteeseen työn ohjaajana. Sivun suorituksen jälkeen käyttäjä ohjataan sivulle `assignmentUpdate.jsp`.

10.3 assignmentUpdate.jsp

Sivulla `assignmentUpdate.jsp` voidaan hallita opinnäytteen yleisiä tietoja. Lisäksi sivulla esitetään linkit muihin opinnäytteen hallintaan liittyviin sivuihin. Sivun näkymä on esitetty kuvassa 13.

Sivu vaatii parametrin `assid` eli käsiteltävän opinnäytteen id:n. Tämän parametrin perusteella tutkitaan, onko käyttäjällä oikeus muuttaa opinnäytteen tietoja. Sivun linkkilistan muodostamista varten tutkitaan myös, onko käyttäjällä oikeus arvioida työn tekijöitä tai hyväksyä opinnäyte. Oikeuksien mukaan linkkilistaa voidaan rajata. Kuvan 13 näkymän käyttäjällä on kaikki mahdolliset oikeudet opinnäytteeseen liittyen.

Tietojen muuttamista varten sivulla esitetään lomake. Lomakkeen lähettäminen vie sivulle `assignmentUpdateInsert.jsp` (luku 10.4). Lisäksi sivulla esitetään mahdollisuus poistaa opinnäyte sovelluksesta. Toiminnon suorittaminen vie käyttäjän sivulle `assignmentremove.jsp` (luku 10.15).



Kuva 13: Opinnäytteen yleisten tietojen hallinnan sivunäkymä.

10.4 assignmentUpdateInsert.jsp

Sivulla `assignmentUpdateInsert.jsp` suoritetaan parametrina saatujen opinnäytteen yleisten tietojen päivitys tietokantaan. Sivun vaatimat parametrit ovat seuraavat:

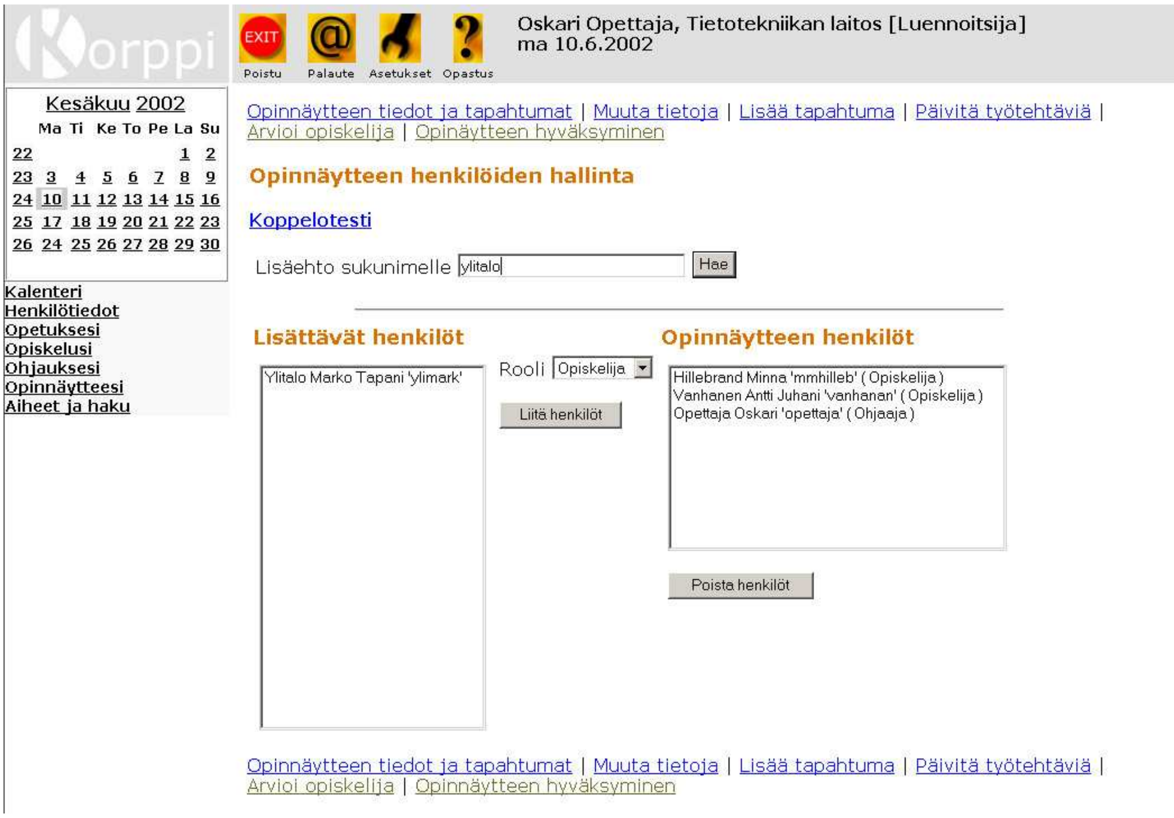
`assid`	on opinnäytteen id, joka ei saa olla `null`.
`title`	sisältää opinnäytteen otsikon. Parametri on pakollinen.
`description`	sisältää opinnäytteen kuvauksen. Parametri on pakollinen.
`keywords`	sisältää opinnäytteen avainsanat. Parametri on pakollinen.
`lineid`	kertoo aiheeseen liitettävän linjan id:n. Parametri on pakollinen.
`typeid`	kertoo aiheeseen liitettävän opinnäytetyypin id:n. Parametri on pakollinen.

`phaseid`	kertoo aiheeseen liitettävän vaiheen id:n. Parametri on pakollinen.
`courseid`	sisältää aiheeseen liitettävän kurssin id:n. Parametri on pakollinen.
`ispublic`	määrää, onko aihe julkinen vai ei. Parametri ei saa puuttua.
`finisheddate`	sisältää aiheen hyväksymispäivämäärän. Parametri ei saa puuttua, mutta sen arvona voi olla tyhjä merkkijono.

Sivun käyttöoikeus on niillä käyttäjillä, joille on annettu opinnäytteen tietojen päivitysoikeus parametrin `assid` mukaisessa opinnäytteessä. Sivun suorituksen jälkeen käyttäjä ohjataan sivulle `assignmentUpdate.jsp` (luku 10.3).

10.5 assignmentUpdatePerson.jsp

Sivulla `assignmentUpdatePerson.jsp` voidaan hallita opinnäytteeseen liittyviä henkilöitä. Lisäksi sivulla esitetään linkit muihin opinnäytteen hallintaan liittyviin sivuihin. Sivun näkymä on esitetty kuvassa 14.



Kuva 14: Opinnäytteen henkilöiden hallinnan sivunäkymä.

Sivu vaatii parametrin `assid` eli käsiteltävän opinnäytteen id:n. Tämän parametrin perusteella tutkitaan, onko käyttäjällä oikeus päivittää opinnäytteeseen liittyviä henkilöitä. Sivun linkkilistan muodostamista varten tutkitaan myös, onko käyttäjällä mm. oikeus arvioida työn tekijöitä tai antaa opinnäytteen lopullinen arviointi. Oikeuksien mukaan linkkilistaa voidaan rajata. Kuvan 14 näkymän käyttäjällä on kaikki mahdolliset oikeudet opinnäytteeseen liittyen.

Sama JSP-sivu hoitaa sekä tietojen päivittämisen että kannan tietojen näyttämisen käyttäjälle, joten tässä kohden sivun logiikka toimii eri tavalla kuin sovelluksen muut JSP-sivut. Jatkokehityksessä voitaisiin erottaa nämä toiminnot erillisiksi sivuiksi. Varsinainen toiminta tai ulkoasu ei kuitenkaan riipu tästä toteutusteknisestä yksityiskohdasta. Sivun painikkeet palauttavat käyttäjän samalle JSP-sivulle.

Tietojen ylläpitämisen vuoksi sivulle voi antaa moniarvoiset parametrit `addperson`, `role`, `removeperson` ja `condition`. Sivu lisää parametrin `addperson` mukaiset KOTKA-järjestelmän käyttäjät opinnäytteeseen parametrin `role` määräämässä roolissa. Vastaavasti se merkitsee poistetuksi parametrin `removeperson` mukaiset käyttäjät. Parametrissa `condition` säilytetään käyttäjän antama henkilöiden hakuehto.

10.6 assignmentUpdateTask.jsp

Sivulla `assignmentUpdateTask.jsp` voidaan hallita opinnäytteeseen liittyviä työtehtäviä. Lisäksi sivulla esitetään linkit muihin opinnäytteen hallintaan liittyviin sivuihin. Sivun näkymä on esitetty kuvassa 15.

Sivu vaatii parametrin `assid` eli käsiteltävän opinnäytteen id:n. Tämän parametrin perusteella tutkitaan, onko käyttäjällä oikeus muuttaa opinnäytteeseen liittyviä työtehtäviä. Sivun linkkilistan muodostamista varten tutkitaan myös, onko käyttäjällä mm. oikeus arvioida työn tekijöitä tai antaa opinnäytteen lopullinen arviointi. Oikeuksien mukaan linkkilistaa voidaan rajata. Kuvan 15 näkymän käyttäjällä on kaikki mahdolliset oikeudet opinnäytteeseen liittyen.

Sama JSP-sivu hoitaa sekä tietojen päivittämisen että kannan tietojen näyttämisen käyttäjälle, joten tässä kohden sivun logiikka toimii eri tavalla kuin sovelluksen muut JSP-sivut. Jatkokehityksessä voitaisiin erottaa nämä toiminnot erillisiksi sivuiksi. Varsinainen toiminta tai ulkoasu ei kuitenkaan riipu tästä toteutusteknisestä yksityiskohdasta. Sivun painikkeet palauttavat käyttäjän samalle JSP-sivulle.

Tietojen ylläpitämisen vuoksi sivulle voidaan antaa mahdollisesti moniarvoiset parametrit `templateid`, `addtask`, `removetask` ja `phaseid`. Parametrin `templateid` antaminen poistaa opinnäytteestä siihen aiemmin liittyvät työtehtävät ja liittää opinnäytteeseen mallin mukaiset työtehtävät. Parametrilla `addtask` voidaan lisätä opinnäytteeseen yksittäisiä työtehtäviä poistamatta opinnäytteeseen aiemmin liitettyjä tehtäviä. Vastaavasti parametrilla `removetask` voidaan poistaa yksittäisiä työtehtäviä.

Kuva 15: Opinnäytteen työtehtävien hallinnan sivunäkymä.

Parametrin `phaseid` avulla voidaan muuttaa yksittäisen työtehtävän vaihetta. Parametrin arvo sisältää tiedon siitä, mitä työtehtävää käsitellään ja mikä vaihe asetetaan uudeksi vaiheeksi.

10.7 assignmentGrade.jsp

Sivulla `assignmentGrade.jsp` voidaan arvostella yksittäisiä opinnäytteisiin liitettyjä opiskelijoita ja muokata aiemmin annettuja arviointeja. Sivun näkymä on esitetty kuvassa 16.

Sivu vaatii parametrin `assid` eli käsiteltävän opinnäytteen id:n. Tämän parametrin perusteella tutkitaan, onko käyttäjällä oikeus arvioida opinnäytteen opiskelijoita. Sivun linkkilistan muodostamista varten tutkitaan myös, onko käyttäjällä mm. oikeus muokata opinnäytteen yleisiä tietoja tai suorittaa opinnäytteen lopullinen arviointi. Oikeuksien mukaan linkkilistaa voidaan rajata. Kuvan 16 näkymän käyttäjällä on kaikki mahdolliset oikeudet opinnäytteeseen liittyen.

Kuva 16: Opiskelijoiden arvioinnin sivunäkymä.

Sivulle voidaan antaa parametrit `studentgrade`, `updated` ja `error`. Antamalla parametri `studentgrade` muokataan parametrin ilmaiseman opiskelijan arviointia. `studentgrade` viittaa taulun `assignmentFinished` kenttään `PersonToBeGradedID`. Parametri `updated` ilmaisee, että sivulle on tultu tietojen tallentamisen kautta, jolloin sivulla ilmoitetaan käyttäjälle tallentamisen onnistumisesta. Jos tallentaminen on epäonnistunut, voidaan virheilmoitus palauttaa käyttäjälle parametrilla `error`.

Arvioinnin tallentaminen vie käyttäjän sivulle `assignmentGradeInsert.jsp` (luku 10.8). Jos käyttäjä suorittaa uuden arvioinnin, välitetään sivulle parametri `new`. Jos käyttäjä haluaa tallentaa arviointiin tehdyt muutokset, välitetään sivulle parametri `update`.

10.8 assignmentGradeInsert.jsp

Opiskelijan arvioinnit tallennetaan tietokantaan sivulla `assignmentGradeInsert.jsp`. Samaa sivua käytetään sekä uuden arvioinnin tallentamiseen että vanhan arvioinnin tietojen muuttamiseen. Sivun käyttöoikeus on niillä henkilöillä, joilla on oikeus arvioida kyseisen opinnäytteen opiskelijoita.

Sivu vaatii seuraavat parametrit:

`assid`	on opinnäytteen id, joka ei saa olla `null`.
`studentid`	sisältää arvosteltavan opiskelijan relaatioid:n valitussa opinnäytteessä. Parametri on pakollinen.
`gradedate`	sisältää arviointipäivämäärän. Parametri on pakollinen.
`gradedby`	sisältää arvioitsijan relaatioid:n valitussa opinnäytteessä. Parametri on pakollinen.
`gradeid`	kertoo annetun arvosanan id:n. Parametri on pakollinen.
`memo`	sisältää arvostelun sanallisen kuvauksen. Parametri on pakollinen, mutta se voi sisältää esimerkiksi tyhjän merkkijonon.
`credits`	sisältää opinnäytteestä annettavat opintoviikot. Parametri on pakollinen.
`new`	kertoo, että käyttäjä haluaa lisätä uuden arvostelun tietokantaan. Jos parametri puuttuu, tulkitaan, että ollaan päivittämässä vanhoja tietokannan arvoja.

Sivun suorittamisen jälkeen käyttäjä ohjataan sivulle `assignmentGrade.jsp` (luku 10.7) parametrilla `updated` ja asettamalla parametrin `studentgrade` arvoksi arvostellun oppilaan relaation id (`studentid`).

10.9 assignmentGradeFinish.jsp

Sivulla `assignmentGradeFinish.jsp` suoritetaan opinnäytteen lopullinen hyväksyminen ja loppuarviointi. Sivun näkymä on esitetty kuvassa 17

Sivu vaatii parametrin `assid` eli käsiteltävän opinnäytteen id:n. Tämän parametrin perusteella tutkitaan, onko käyttäjällä oikeus hyväksyä opinnäyte ja suorittaa loppuarviointi. Sivun linkkilistan muodostamista varten tutkitaan myös, onko käyttäjällä mm. oikeus muokata opinnäytteen yleisiä tietoja tai arvioida opinnäytteen opiskelijoita. Oikeuksien mukaan linkkilistaa voidaan rajata. Kuvan 17 näkymän käyttäjällä on kaikki mahdolliset oikeudet opinnäytteeseen liittyen.

Arvioinnin tallentaminen vie käyttäjän sivulle `assignmentGradeFinishInsert.jsp` (luku 10.10). Sivulle välitetään opinnäytteeseen liitetyn kurssin mukainen kurssikoodi.

Kuva 17: Opinnäytteen hyväksymisen sivunäkymä.

10.10 assignmentGradeFinishInsert.jsp

Opiskelijoiden lopullinen arvolause tallennetaan tietokantaan sivulla `assignmentGradeFinishInsert.jsp`. Samaa sivua käytetään sekä uuden arvioinnin tallentamiseen että vanhan arvioinnin tietojen muuttamiseen. Sivun käyttöoikeus on niillä henkilöillä, joilla on oikeus hyväksyä opinnäyte.

Sivu vaatii seuraavat parametrit:

`assid`	on opinnäytteen id, joka ei saa olla `null`.
`coursecode`	sisältää opintosuoritusrekisteriin tallennettavan kurssikoodin.
`studentid`	sisältää arvosteltavan opiskelijan relaatioid:n valitussa opinnäytteessä. Parametri on pakollinen ja se voi olla moniarvoinen.
`gradedate`	sisältää arviointipäivämäärän. Parametri on pakollinen ja se voi olla moniarvoinen.
`gradeid`	kertoo annetun arvosanan id:n. Parametri on pakollinen ja se voi olla moniarvoinen.
`credits`	sisältää opinnäytteestä annettavat opintoviikot. Parametri on pakollinen ja se voi olla moniarvoinen.

Sivun suoritus perustuu siihen, että tietokannan taulussa `finish` aiemmin olleet opinnäytteen suoritukset merkitään poistetuiksi ja tämän jälkeen tietokantaan viedään uudet arvot uusina taulun riveinä. Tämä ei aiheuta ristiriitatilanteita tietokannassa, sillä taululle `finish` ei ole määrätty avainkenttiä.

Tietokannan päivittämisen jälkeen käyttäjä valitsee, siirtyykö hän takaisin opinnäytteen hyväksymissivulle `assignmentGradeFinish.jsp` (luku 10.9) vai opinnäytteen yleisten tietojen hallintaan sivulle `assignmentUpdate.jsp` (luku 10.3).

10.11 assignmentGradeStudent.jsp

Sivulla `assignmentGradeStudent.jsp` opiskelija pääsee näkemään itseensä liitetyn arvioinnin. Sivun näkymä on esitetty kuvassa 18.

Kuva 18: Opinnäytteen arvioinnin sivunäkymä opiskelijalle.

Sivu vaatii parametrin `assid`. Tämän opinnäytteen id:n sisältävän parametrin avulla tarkastetaan, että käyttäjä on opiskelijana kyseisessä opinnäytteessä. Näin ollen esimerkiksi opinnäytteen ohjaajat eivät pääse kyseiselle sivulle lukemaan arvosteluita. Myöskään muut opiskelijat eivät pääse näkemään kuin heitä koskevat arvioinnit.

10.12 addEvent.jsp

Opinnäytteeseen voidaan liittää tapahtumia sivulla `addEvent.jsp`. Sivun näkymä on esitetty kuvassa 19. Tapahtuman tietojen syöttämiseen käytetyn lomakkeen kentät ovat pakollisia. Lomakkeen lähettäminen ohjaa käyttäjän sivulle `insertEvent` (luku 10.13).

Kuva 19: Tapahtuman lisäämisen sivunäkymä.

Sivu vaatii parametrin `assid`, joka ilmaisee käsiteltävän opinnäytteen. Sivun käyttöoikeus on kaikilla opinnäytteeseen liitetyillä henkilöillä ja sivun sisältö on sama sekä opiskelijoille että ohjaajille. Ainoastaan tapahtuman tyyppi on rajattu siten, että ohjaajille esitetään kaikki tietokannan tapahtumatyypit, kun taas opiskelijalle esitetään vain '`Oppilaan raportti`'.

10.13 insertEvent.jsp

Sivulla `insertEvent.jsp` tallennetaan tietokantaan opinnäytteeseen lisättävän tapahtuman tiedot. Sivu vaatii seuraavat parametrit:

`assid`	on opinnäytteen id, joka ei saa olla `null`.
`regid`	on tapahtuman kirjaajan id. Parametri on pakollinen.
`date`	sisältää tapahtuman päivämäärän. Parametri on pakollinen.
`begintime`	sisältää tapahtuman aloitusajan. Parametri on pakollinen.
`endtime`	sisältää tapahtuman loputusajan. Parametri on pakollinen.

`title`	sisältää tapahtuman otsikon.
`memo`	sisältää tapahtuman kuvauksen.
`typeid`	ilmaisee tapahtuman tyypin id:n.
`taskid`	ilmaisee tapahtumaan liitetyn työtehtävän id:n. Parametri voi sisältää arvon `null`.

Virheellisten päivämäärä- ja kellonaikasyötteiden antaminen vie käyttäjän sivulle `error.jsp`.

Samalla kun tietokantaan tallennetaan tapahtuman tiedot, merkitään tapahtuma uudeksi tapahtumaksi kaikille opinnäytteeseen liitetyille henkilöille tapahtuman kirjaajaa lukuunottamatta. Sivun suorituksen jälkeen käyttäjä ohjataan sivulle `assignmentEvent.jsp` (luku 10.14) parametrinaan juuri lisätyn tapahtuman id.

10.14 assignmentEvent.jsp

Opinnäytteen tapahtumien tietoja pääsee tarkastelemaan sivulla `assignemntEvent.jsp`. Sivun näkymä on esitetty kuvassa 10.14.

Kuva 20: Tapahtuman tietojen katselemisen sivunäkymä.

Sivu vaatii parametrin `eventid`, jonka perusteella, päätellään minkä tapahtuman tiedot käyttäjälle esitetään. Parametrin perusteella saadaan myös selville, mihin opinnäytteeseen tapahtuma on liitetty. Sivulle pääsy edellyttää, että käyttäjällä on oikeus tarkastella opinnäytteen yksityiskohtaisia tietoja.

10.15 assignmentRemove.jsp

Sivulla `assignmentRemove.jsp` käyttäjältä varmistetaan, haluaako hän poistaa valitun opinnäytteen KOPPELO-sovelluksesta eli merkitä opinnäytteen poistetuksi tietokannasta.

Sivu vaatii parametrin `assid` eli opinnäytteen id:n. Tämän perusteella varmistetaan, että käyttäjällä on oikeus poistaa opinnäyte tietokannasta. Mikäli käyttäjä haluaa poistaa opinnäytteen, ohjataan hänet sivulle `assignmentRemoveInsert.jsp` (luku 10.16). *Takaisin*-linkistä käyttäjä pääsee opinnäytteen tietojen hallintasivulle `assignmentUpdate.jsp` (luku 10.3).

10.16 assignmentRemoveInsert.jsp

Opinnäytteen merkitseminen poistetuksi KOTKA-järjestelmän tietokannasta tehdään sivulla `assignmentRemoveInsert.jsp`. Sivu vaatii parametrin `assid`, joka ilmaisee poistettavan opinnäytteen id:n. Tämän perusteella myös varmistetaan, että käyttäjällä on oikeus suorittaa poistaminen.

Opinnäytteen poistamisen jälkeen käyttäjä ohjataan sivulle `main.jsp` parametrilla `teacher`.

11 Käyttäjän opinnäytteet, haku ja raportit

Luvussa esitellään opinnäytteiden hallintaan liittyviä käyttäjäkohtaisesti muuntuvia sivuja. Sivujen käyttöoikeus tulisi olla automaattisesti (luvun 11.5 sivua `assignment.jsp` lukuunottamatta) kaikilla KOTKA-järjestelmän käyttäjillä. Siten luvussa esitetyt sivut löytyvät myös luvun 7.3 opiskelijan sivustokartasta kuvasta 4.

11.1 main.jsp

Sivu `main.jsp` on KOPPELO-sovelluksen käyttäjäkohtaisen osan pääsivu. Kyseiseen sivuun osoittaa myös suora linkki KOPPELO-sovelluksen kotihakemistosta. Samalla sivulla esitetään sekä opiskelijan että ohjaajan linkit opinnäytesovelluksen keskeisimpiin osiin.

Näkymän rajaaminen tehdään sivun saamien parametrien avulla. Luvun 8 kuvassa 7 on esitetty ohjaajan perusnäkymä KOPPELO-sovellukseen. Opiskelijan näkymä on muutoin vastaava, mutta ohjattavien opinnäytteiden tilalla opiskelijalle näytetään tekeillä olevat opinnäytteet. Vastaavasti ylä- ja ala-reunan linkkilistat vaihtuvat “käyttäjän” mukaan.

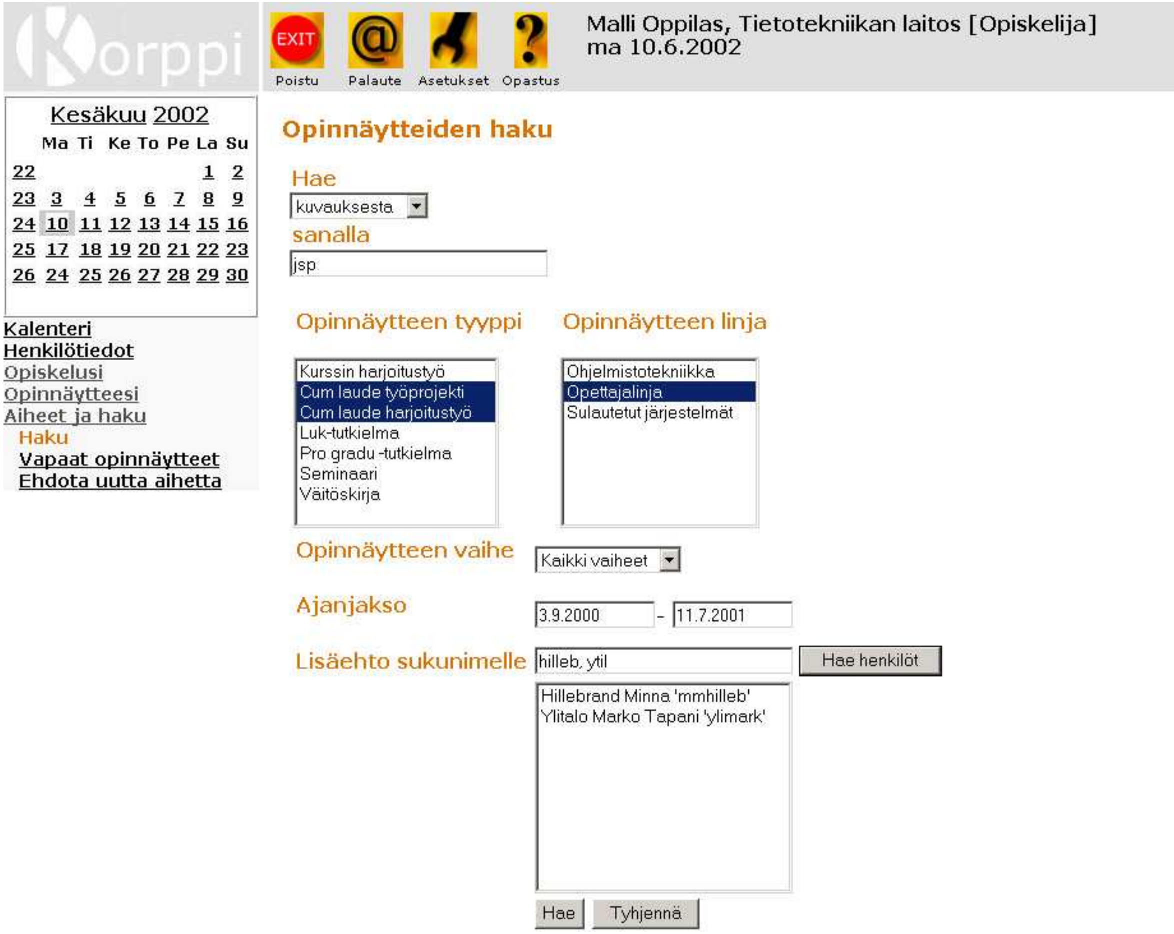
Sivulla vastaanotetaan parametrit `student` ja `teacher`. Mikäli parametri `student` on annettu, näytetään opiskelijan osa sivusta eli tekeillä olevat omat opinnäytteet. Mikäli parametri `teacher` on annettu, näytetään käyttäjän ohjattavana olevat opinnäytteet. Sivun yläosan uudet tapahtumat esitetään joka tapauksessa.

Varsinaista käyttöoikeustarkastusta sivulla ei siis tehdä, vaan jokainen KOTKA-järjestelmän käyttäjä pystyy kutsumaan kyseistä sivua suoralla WWW-osoitteella. Sen sijaan sivu ei esiinny käyttäjän navigointipalkissa ennen kuin käyttäjä on liitetty johonkin opinnäytteeseen joko opiskelijana tai ohjaajana.

Sivulla luetellut opinnäytteet ovat linkkejä vastaavien opinnäytteiden yleisiin tietoihin sivulla `assignment.jsp` (luku 11.5). Opinnäytteisiin liittyvät päivämäärät ovat linkkejä opinnäytteen tapahtumiin sivulle `assignmentEvent.jsp` (luku 10.14). Ohjaajan osassa opinnäytteen vaihe on linkki opinnäytteen tietojen hallinnointiin sivulle `assignmentUpdate.jsp` (luku 10.3).

11.2 search.jsp

Sivulla `search.jsp` määritellään käytetyt hakuehdot. Opinnäytteitä ja niiden aiheita voidaan hakea otsikon, kuvauksen, avainsanojen, tyypin, linjan, vaiheen, ajanjakson ja tekijän tai ohjaajan perusteella. Sivun näkymä on esitetty kuvassa 21.



Kuva 21: Opinnäytteiden hakusivun sivunäkymä.

Sivu vastaanottaa seuraavat parametrit:

`personid`	on henkilön id, joka saa olla myös `null`.
`searchText`	on vapaan tekstihaun syöte. Parametri ei ole pakollinen.
`searchTextMethod`	ilmaisee numerolla vapaan tekstihaun kohteen (otsikko, kuvaus tai avainsanat). Parametri ei ole pakollinen.
`phaseid`	on vaiheen id, joka saa olla myös `null`.

`timebegin`	on ajanjakson aloittamisajankohta. Käytettävä yhdessä `timeend`-parametrin kanssa. Parametri ei ole pakollinen.
`timeend`	on ajanjakson lopetusajankohta. Käytettävä yhdessä `timebegin`-parametrin kanssa. Parametri ei ole pakollinen.
`lineid`	on linjan id, joka saa olla myös `null`.
`typeid`	on opinnäytteen tyypin id, joka saa olla myös `null`.
`condition`	on henkilöhaun syöte, jonka perusteella henkilön valintalista rakentuu. Parametri ei ole pakollinen.

11.3 searchPerson.jsp

Sivu `searchPerson.jsp` on käyttäjälle näkymätön sivu. Sivu vastaanottaa samat parametrit kuin `search.jsp`-sivu (luku 11.2). Sivun tarkoitus on tarkistaa, onko `condition`-parametri annettu. Jos se on annettu, palataan sivulle `search.jsp`. Muutoin siirrytään eteenpäin sivulle `searchResult.jsp`.

11.4 searchResult.jsp

Sivu `searchResult.jsp` näyttää hakutuloksen annettujen parametrien perusteella. Sivu vastaanottaa samat parametrit kuin `search.jsp`-sivu (luku 11.2). Hakutulos järjestetään opinnäytteen tyypin ja otsikon mukaan. Kuvassa 22 on esitetty hakutuloksen näkymä. Kuvassa on saatu hakutuloksena kaikki yksittäisen opiskelijan hyväksytyt opinnäytteet.

Kuva 22: Hakutuloksen sivunäkymä.

Raportointi toteutettiin sivulla siten, että valitsemalla selaimen tulostustoiminnon käyttäjä saa tulostettua sellaisen version hakutuloksista, jossa ei näy sovelluksen käyttöliittymään kuuluvia navigointipalkkeja tai muita muotoiluja. Tämä ominaisuus toteutettiin erillisellä tyylitiedostolla `report.css`.

11.5 assignment.jsp

Sivulla `assignment.jsp` esitetään opinnäytteen yleisiä tietoja. Sivun näkymä opinnäytteeseen liitetylle opiskelijalle on esitetty kuvassa 23.

Kuva 23: Opinnäytteen tietojen sivunäkymä opiskelijalle.

Sivu vaatii parametrin `assid` eli tarkasteltavan opinnäytteen id:n. Mikäli parametria ei ole annettu, sivu pyytää parametria `origid`, jonka perusteella se lähettää sovelluksen suorituksen sivulle `assignmentInfo.jsp` (luku 12.1). Mikäli kumpaakaan parametria ei ole annettu, siirretään käyttäjä virhesivulle `error.jsp`. Parametrin `assid` perusteella tarkastetaan, että käyttäjä on liitetty opinnäytteeseen. Ellei näin ole, siirretään käyttäjä opinnäytteen yleisten tietojen sivulle `assignmentInfo.jsp`.

Sivulla näytetään opinnäytteen yleiset tiedot ja opinnäytteeseen liitetyt henkilöt, työtehtävät ja tapahtumat. Henkilöille, joilla on opinnäytteessä arviointioikeus, näytetään lisäksi opinnäytteen arvioinnit. Opiskelijoille näytetään heihin liittyvät henkilökohtaiset arvioinnit.

Sivun linkkilistoja varten tarkastetaan käyttäjän oikeudet opinnäytteessä siten, että linkkilista rajataan käyttäjän oikeustasoa vastaavaksi.

12 Yleiset sivut

Luvussa kuvataan sellaisia KOPPELO-sovelluksen sivuja, joille ei ole annettu erityisiä käyttöoikeusmäärityksiä. Lisäksi luvussa esitellään sovellukseen liittyvät virhesivut.

12.1 assignmentInfo.jsp

Sivulla `assignmentInfo.jsp` esitetään opinnäytteestä vain yleiset tiedot, kuten otsikko, kuvaus ja avainsanat. Sivun näkymä on esitetty kuvassa 24. Sivu näkyy samanlaisena kaikille KOTKA-järjestelmän käyttäjäryhmille.

Kuva 24: Opinnäytteen tietojen yleinen sivunäkymä.

Sivu vaatii parametrin `assid`, joka ilmaisee käsiteltävän opinnäytteen id:n. Mikäli parametria ei ole annettu, sivu pyytää parametria `origid` ja lähettää suorituksen edelleen opinnäytteen aiheen tietojen sivulle `assignmentOriginalInfo` (luku 12.2).

12.2 assignmentOriginalInfo.jsp

Sivulla `assignmentOriginalInfo.jsp` esitetään opinnäytteen aiheen tai aihe-ehdotuksen tiedot. Sivun näkymä on esitetty kuvassa 25. Sivu vaatii parametrin `origid` eli sen opinnäytteen aiheen id:n, jonka tietoja halutaan tarkastella.

Kuva 25: Opinnäytteen aiheen tietojen sivunäkymä.

12.3 error.jsp

Sivulla `error.jsp` viestitetään käyttäjälle virheellisestä toiminnasta KOPPELO-sovelluksen sivuilla. Sivu vastaanottaa parametrit `page` ja `message`. Parametri `page` ilmaisee, millä sovelluksen sivulla virhe on tapahtunut ja parametri `message` sisältää tiedon siitä, mikä saattaa olla virheen syy. Parametrien sisältö esitetään sivulla käyttäjälle. Jatkotoimenpiteet jäävät kuitenkin käyttäjän varaan.

12.4 errorNoAccess.jsp

Sivua `errorNoAccess.jsp` kutsutaan niissä tilanteissa, kun käyttäjä yrittää edetä KOPPELO-sovelluksessa (esimerkiksi suoran WWW-osoitteen kautta) sellaiselle sivulle, johon hänellä ei ole käyttöoikeutta. Sivulla tiedotetaan käyttäjälle virheellisestä etenemisestä, mutta jatkotoimenpiteet jäävät käyttäjän varaan.

13 Ylläpidon JSP-sivut

Luvussa esitellään KOPPELO-sovelluksen ylläpitäjälle tarkoitettuja sivuja. Näiden sivujen keskinäiset suhteet on esitetty luvun 7.5 kuvassa 6.

Ylläpidon sivuilla hallinnoidaan KOPPELO-sovellukseen liittyviä oikeuksia. Oikeudet jakautuvat käyttäjä-, rooli- ja käyttäjäryhmäoikeuksiin, joiden merkitys on hieman erilainen eri tilanteissa. Käyttäjäkohtaiset oikeudet otetaan huomioon kaikissa sovelluksen toiminnan tilanteissa. Roolikohtaisia oikeuksia tarkastellaan tilanteissa, joissa käsiteltävä opinnäyte voidaan tunnistaa yksikäsitteisesti. Käyttäjäryhmäkohtaisia oikeuksia tarkastellaan silloin, kun opinnäytettä ei voida tunnistaa tai käyttäjä ei liity opinnäytteeseen.

13.1 adminMain.jsp

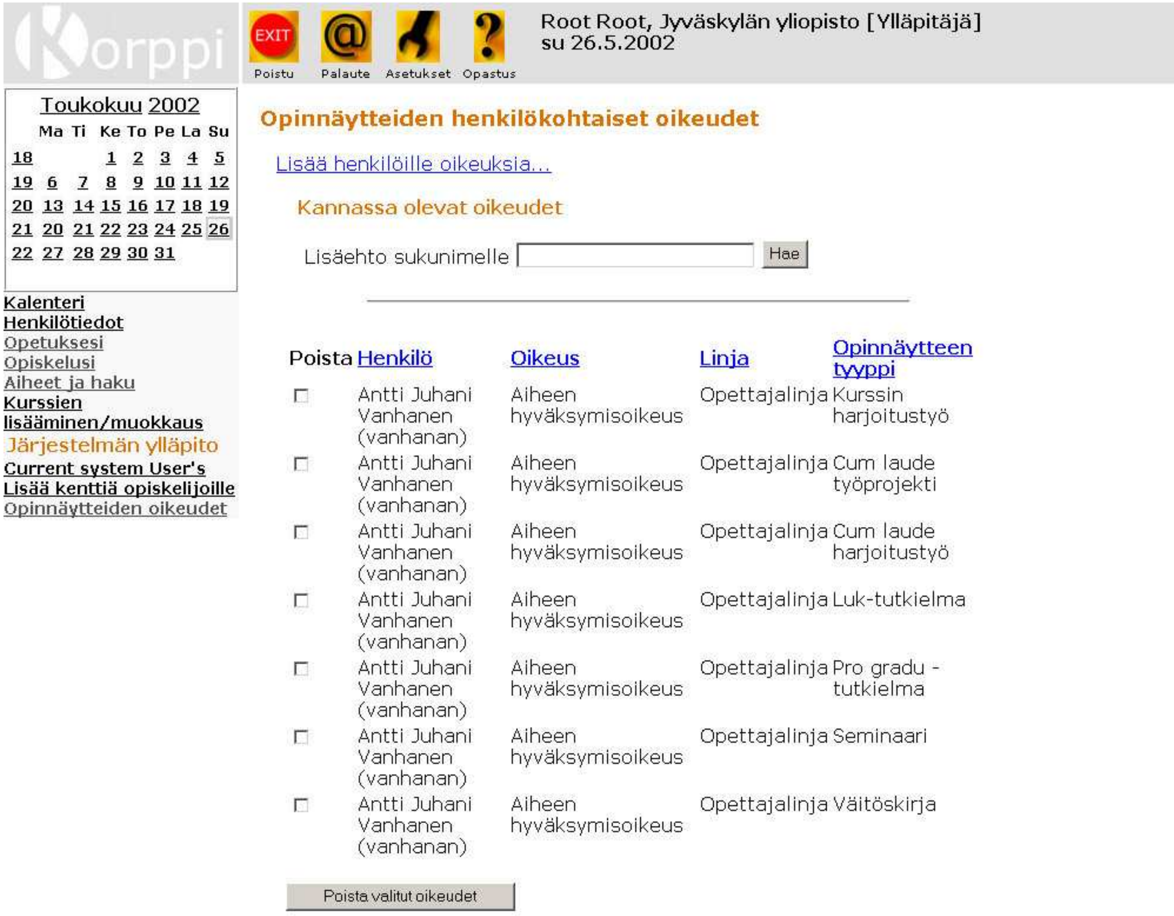
Sivu `adminMain.jsp` on ylläpidon KOPPELO-sovellukseen liittyvä pääsivu. Sivulle pääsy on rajoitettu ainoastaan järjestelmän ylläpitäjille, joille sivu näkyy navigointipalkissa erillisenä kohtana *Opinnäytteiden oikeudet*. Sivun näkymä on esitetty kuvassa 26.

Kuva 26: Ylläpitäjän pääsivun näkymä.

Sivu ei vaadi parametreja. Se sisältää linkit edelleen oikeuksien hallinnointisivuille `adminAssRightPerson.jsp` (luku 13.2), `adminAssRightRole.jsp` (luku 13.6) ja `adminAssRightGroup.jsp` (luku 13.10).

13.2 adminAssRightPerson.jsp

Sivu `adminAssRightPerson.jsp` on tarkoitettu käyttäjäkohtaisten oikeuksien tarkastelemiseen ja poistamiseen. Sivulla olevan hakutoiminnon avulla voidaan rajata tarkasteltavaa käyttäjäjoukkoa. Sivun näkymä on esitetty kuvassa 27.



Kuva 27: Käyttäjien oikeuksien tarkastelun näkymä.

Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla. Sivu ei vaadi parametreja, mutta parametrilla `condition` sille voidaan asettaa hakuehto käyttäjän nimen mukaan. Ellei hakuehtoa ole annettu, näytetään sivulla vain linkki oikeuksien asettamisen sivulle `adminAssRightPersonAdd.jsp` (luku 13.4) ja syöttökenttä nimellä hakemista varten. Haku palauttaa käyttäjän samalle sivulle.

Hakutuloksena löytyneet käyttäjät esitetään sivulla listana. Oikeuksia voidaan poistaa valitsemalla poistettavat oikeudet ja lähettämällä lomake sivulle `adminAssRightPersonRemove.jsp` (luku 13.3).

Hakutuloksen otsikkotekstit toimivat linkkeinä samalle sivulle, jolloin sivulle välitetään parametri `orderby` ilmaisemaan, minkä tiedon mukaisesti hakutulos halutaan järjestää (aakkosjärjestykseen).

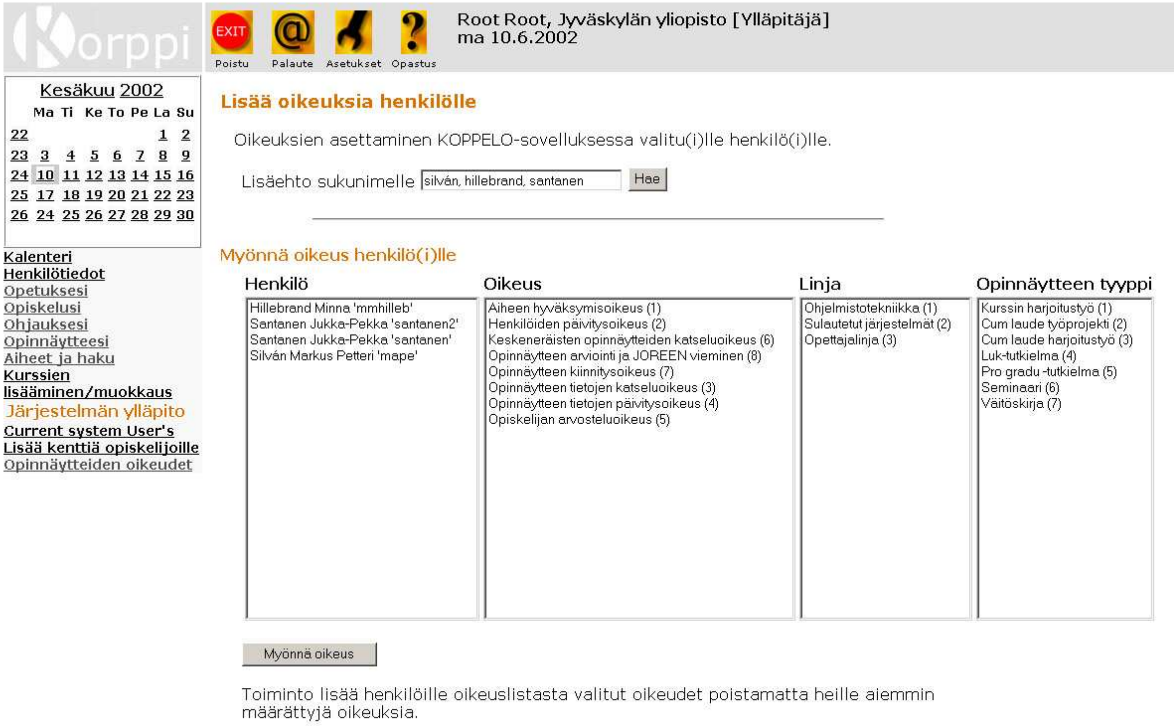
13.3 adminAssRightPersonRemove.jsp

Sivun `adminAssRightPersonRemove.jsp` tarkoitus on merkitä henkilöille myönnettyjä oikeuksia poistetuksi KOPPELO-sovelluksesta. Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla.

Sivu vaatii parametrit `personid`, `lineid`, `typeid` ja `rightid`, jotka ilmaisevat järjestelmään lisättävät oikeudet. Parametrit voivat olla moniarvoisia. Tietokannan päivityksen jälkeen käyttäjä ohjataan sivulle `adminAssRightPerson.jsp` (luku 13.2).

13.4 adminAssRightPersonAdd.jsp

Sivu `adminAssRightPersonAdd.jsp` on tarkoitettu käyttäjäkohtaisten oikeuksien lisäämiseen. Sivulla olevan hakutoiminnon avulla voidaan rajata niiden käyttäjien joukkoa, joille oikeudet aiotaan myöntää. Sivun näkymä on esitetty kuvassa 28.



Kuva 28: Käyttäjien oikeuksien lisäämisen näkymä.

Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla.

Sivu ei vaadi parametreja, mutta parametrilla `condition` sille voidaan asettaa hakuehto käyttäjän nimen mukaan. Ellei hakuehtoa ole annettu, näkyy käyttäjien valintalista tyhjänä. Hakulomake palauttaa käyttäjän samalle sivulle.

Lomakkeella olevat oikeuksien lisäämiseen liittyvät valintalistat täytyy täyttää siten, että jokaisesta kohdasta tulee vähintään yksi kohta valittua. Siten käyttäjiin tulee liitettyä valitut oikeudet määrätyllä laitoksella tietyissä opinnäytetyypeissä. Lomakkeen lähettäminen vie käyttäjän sivulle `adminAssRightPersonInsert.jsp` (luku 13.5).

13.5 adminAssRightPersonInsert.jsp

Sivulla `adminAssRightPersonInsert.jsp` lisätään KOTKA-järjestelmän tietokantaan parametrin mukaisille käyttäjille oikeuksia KOPPELO-sovelluksessa. Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla.

Sivu vaatii parametrit `personid`, `lineid`, `typeid` ja `rightid`, jotka voivat olla moniarvoisia. Oikeuksia yritetään lisätä tietokantaan. Lisäys voi epäonnistua esimerkiksi tilanteessa, jossa jollekin henkilölle on jo aiemmin merkitty samat oikeudet. Tätä virhetilannetta ei kuitenkaan ole tarvetta näyttää käyttäjälle millään tavalla.

Tietokannan päivityksen jälkeen käyttäjä ohjataan oikeuksien tarkastelusivulle `adminAssRightPerson.jsp` (luku 13.2).

13.6 adminAssRightRole.jsp

Sivu `adminAssRightRole.jsp` on tarkoitettu opinnäytteiden roolien mukaisten oikeuksien tarkastelemiseen ja poistamiseen. Sivun näkymä on esitetty kuvassa 29.

Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla. Sivu ei vaadi parametreja.

KOPPELO-sovellukseen liittyvät roolien oikeudet esitetään sivulla listana. Oikeuksien otsikkotekstit toimivat linkkeinä samalle sivulle. Tällöin sivulle välitetään parametri `orderby` ilmaisemaan, minkä tiedon mukaisesti hakutulos halutaan järjestää (aakkosjärjestykseen). Oikeuksia voidaan poistaa valitsemalla poistettavat oikeudet ja lähettämällä lomake sivulle `adminAssRightRoleRemove.jsp` (luku 13.7).

Kuva 29: Opinnäytteen roolien oikeuksien tarkastelun näkymä.

13.7 adminAssRightRoleRemove.jsp

Sivun `adminAssRightRoleRemove.jsp` tarkoitus on merkitä rooleille myönnettyjä oikeuksia poistetuksi KOPPELO-sovelluksesta. Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla.

Sivu vaatii parametrit `roleid` ja `rightid`, jotka ilmaisevat poistettavat oikeudet. Parametrit voivat olla moniarvoisia. Tietokannan päivityksen jälkeen käyttäjä ohjataan sivulle `adminAssRightRole.jsp` (luku 13.6).

13.8 adminAssRightRoleAdd.jsp

Sivu `adminAssRightRoleAdd.jsp` on tarkoitettu opinnäytteen roolien oikeuksien lisäämiseen. Sivun näkymä on esitetty kuvassa 30.

Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla. Sivu ei vaadi parametreja.

Lomakkeella olevat oikeuksien lisäämiseen liittyvät valintalistat täytyy täyttää siten, että molemmista kohdasta (rooli ja oikeus) tulee vähintään yksi kohta valittua. Lomakkeen lähettäminen vie käyttäjän sivulle `adminAssRightRoleInsert.jsp` (luku 13.9).

Kuva 30: Opinnäytteen roolien oikeuksien lisäämisen näkymä.

13.9 adminAssRightRoleInsert.jsp

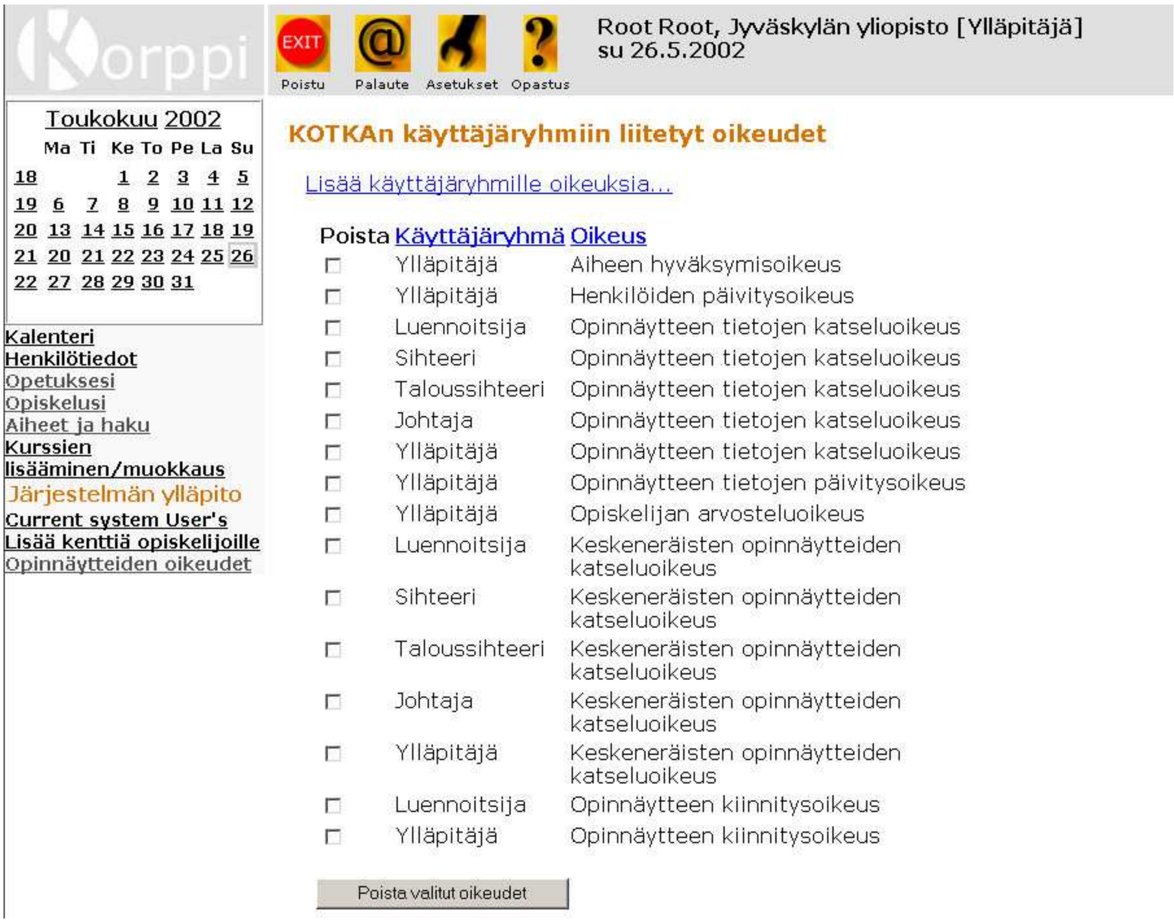
Sivulla `adminAssRightRoleInsert.jsp` lisätään KOTKA-järjestelmän tietokantaan parametrin mukaisille opinnäytteen rooleille oikeuksia KOPPELO-sovelluksessa. Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla.

Sivu vaatii parametrit `roleid` ja `rightid`, jotka ilmaisevat kantaan lisättävät oikeudet tietyille rooleille. Parametrit voivat olla moniarvoisia. Oikeuksia yritetään lisätä tietokantaan. Lisäys voi epäonnistua esimerkiksi tilanteessa, jossa jollekin roolille on jo aiemmin merkitty samat oikeudet. Tätä virhetilannetta ei kuitenkaan ole tarvetta näyttää käyttäjälle millään tavalla.

Tietokannan päivityksen jälkeen käyttäjä ohjataan oikeuksien tarkastelusivulle `adminAssRightRole.jsp` (luku 13.6).

13.10 adminAssRightGroup.jsp

Sivu `adminAssRightGroup.jsp` on tarkoitettu KOTKA-järjestelmän käyttäjäryhmäkohtaisten oikeuksien tarkastelemiseen ja poistamiseen. Sivun näkymä on esitetty kuvassa 31.



Kuva 31: Käyttäjäryhmien oikeuksien tarkastelun näkymä.

Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla. Sivu ei vaadi parametreja.

Tietokannan sisältämät käyttäjäryhmiin liitetyt oikeudet esitetään sivulla listana. Listan otsikkotekstit toimivat linkkeinä samalle sivulle. Linkeistä välitetään sivulle parametri `orderby` ilmaisemaan, minkä tiedon mukaisesti hakutulos halutaan järjestää (aakkosjärjestykseen). Oikeuksia voidaan poistaa valitsemalla poistettavat oikeudet ja lähettämällä lomake sivulle `adminAssRightGroupRemove.jsp` (luku 13.11).

13.11 adminAssRightGroupRemove.jsp

Sivun `adminAssRightGroupRemove.jsp` tarkoitus on merkitä käyttäjäryhmille myönnettyjä oikeuksia poistetuksi KOPPELO-sovelluksesta. Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla.

Sivu vaatii parametrit `groupid` ja `rightid`, jotka ilmaisevat kannasta poistettavat oikeudet. Parametrit voivat olla moniarvoisia. Tietokannan päivityksen jälkeen käyttäjä ohjataan sivulle `adminAssRightGroup.jsp` (luku 13.10).

13.12 adminAssRightGroupAdd.jsp

Sivu `adminAssRightGroupAdd.jsp` on tarkoitettu käyttäjäryhmäkohtaisten oikeuksien lisäämiseen. Sivun näkymä on esitetty kuvassa 32.

Kuva 32: Käyttäjäryhmien oikeuksien lisäämisen näkymä.

Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla. Sivu ei vaadi parametreja.

Lomakkeella olevat oikeuksien lisäämiseen liittyvät valintalistat täytyy täyttää siten, että molemmista kohdasta (käyttäjäryhmä ja oikeus) tulee vähintään yksi kohta valittua. Siten käyttäjäryhmiin tulee liitettyä valitut oikeudet. Lomakkeen lähettäminen vie käyttäjän sivulle `adminAssRightGroupInsert.jsp` (luku 13.13).

13.13 adminAssRightGroupInsert.jsp

Sivulla `adminAssRightPersonInsert.jsp` lisätään KOTKA-järjestelmän tietokantaan parametrin mukaisille käyttäjäryhmille oikeuksia KOPPELO-sovelluksessa. Sivun käyttö vaatii, että käyttäjä on kirjautunut järjestelmään ylläpitäjän oikeuksilla.

Sivu vaatii parametrit `groupid` ja `rightid`, jotka sisältävät tietokantaan tallennettavat oikeudet valituilla käyttäjäryhmillä. Parametrit voivat olla moniarvoisia. Oikeuksia yritetään lisätä tietokantaan. Lisäys voi epäonnistua esimerkiksi tilanteessa, jossa jollekin käyttäjäryhmälle on jo aiemmin merkitty samat oikeudet. Tätä virhetilannetta ei kuitenkaan ole tarvetta näyttää käyttäjälle millään tavalla.

Tietokannan päivityksen jälkeen käyttäjä ohjataan käyttäjäryhmien oikeuksien tarkastelemisen sivulle `adminAssRightGroup.jsp` (luku 13.10).

14 Automaattisten viestien lähettäminen

Luvussa esitetään, kuinka KOPPELO-sovelluksen automaattinen viestinlähetysjärjestelmä toimii. Automaattisilla viesteillä tarkoitetaan niitä tilanteita, kun viestejä ei voida lähettää käyttäjän aktiivisen toiminnan perusteella.

14.1 Muistutustilanteet ja toteutustapa

KOPPELO-sovellus lähettää käyttäjille muistutussähköpostiviestejä parissa erilaisessa erilaisessa tilanteessa. Muistustusviesti lähetetään mm. silloin, kun käyttäjän kirjaama aihe tai aihe-ehdotus on ollut riittävän kauan järjestelmässä ja aihe alkaa erääntyä.

Toisaalta muistustussähköpostiviesti lähetetään myös sellaisessa tilanteessa, kun ohjaaja on määrännyt opinnäytteeseen muistustusvälin. Muistutusväli ilmaisee ajanjakson, jonka kuluessa opinnäytteeseen täytyy liittää uusia tapahtumia. Muutoin asiasta ilmoitetaan opinnäytteen tekijöille.

Tämä automatiikka saadaan aikaiseksi luomalla erillinen JSP-sivu, joka suoritetaan päivittäin palvelimelta. Päivittäinen suoritus saadaan aikaiseksi määräämällä palvelimeen ajastettu toiminto *cron*. Siten automaattisesti suoritettava sivu on teknisesti samanlainen kuin mikä tahansa muu järjestelmän sivu. Tämän sivun rajoituksena on vain se, että se ei voi tarkastaa käyttäjän oikeellisuutta.

Muistutussähköposteihin liittyvät puutteet on kirjattu jatkokehitysideoihin lukuun 17.10.

14.2 autoMatik.jsp

KOPPELO-järjestelmän automaattiset muistutussähköpostiviestit lähetetään sivulta `autoMatik.jsp`. Sivu ei vaadi mitään parametreja, eikä se tarkasta käyttäjän henkilöllisyyttä.

Sivulla suoritetaan tarvittavat tietokantakyselyt haluttujen sähköpostiviestien selvittämiseksi ja lähetetään viestit. Sivu ei ilmaise millään tavalla onnistumistaan tai epäonnistumistaan. Tämä tieto voitaisiin tosin kirjata esimerkiksi lokitiedostoon.

Sivun suurin epäonnistumisen syy on se, että käyttäjät eivät ole kirjanneet järjestelmään sähköpostiosoitteitaan. Tähän tilanteeseen ei kuitenkaan ole muita mahdollisia ratkaisuja kuin vaatia kaikkia käyttäjiä ilmoittamaan sähköpostiosoitteensa.

15 Sovelluksen testaaminen

Luvussa tarkastellaan sovelluksen testaamisen yleisiä periaatteita ja esitellään käytetty testausympäristö. Testauksesta on lisäksi kirjoitettu erillinen testausraportti, jossa esitellään tarkemmin käytettyjä testausmenetelmiä ja -tuloksia. Eri käyttäjien täyttämistä testitapauslomakkeista on koottu sovelluksessa havaittuja virheellisiä toimintoja tämän dokumentin lukuun 16. Täytetyt testilomakkeet ovat testausraportin liitteenä.

15.1 Testausympäristöt

Sovelluksen testaamiseen käytettiin useita eri selaimia ja selainversioita. Tällä pyrittiin löytämään sovelluksesta selainriippuvaisia osia ja muokkaamaan sovelluksesta kaikissa ympäristöissä toimiva kokonaisuus.

Taulukossa 4 on esitelty sovelluksen testauksessa käytetyt käyttöjärjestelmät, selaimet ja selainversiot.

KÄYTTÖJÄRJESTELMÄ	**SELAIN**	**VERSIOT**
Windows 2000	Internet Explorer	5.0 ja 6.0
Windows XP	Netscape Navigator	4.7 ja 6.0
	Opera	5.12
	Mozilla	0.9.4
Debian Linux	Netscape Navigator	4.78
Red Hat Linux	Opera	5.0
	Mozilla	0.9.9
Sun Solaris	Netscape Navigator	4.76

Taulukko 4: Testauksessa käytetyt käyttöjärjestelmät ja selaimet.

Lopullinen sovellus toimii lähes samalla tavalla kaikissa taulukon 4 mukaisissa selaimissa. Eroja esiintyy lähinnä sovellusten tavoissa käsitellä tyylitiedostoja. Toiminnalliset erot on kuvattu luvussa 16.3.

15.2 Testausperiaatteita

Kaikki sovelluksen osat testattiin huolellisesti jo ohjelmointivaiheessa. Jokainen yksittäinen JSP-sivu testattiin heti sen valmistuttua, jotta myöhemmissä vaiheissa ei tarvinnut keskittyä etsimään piileviä virheitä koko ohjelmakoodista. Tämä testausperiaate toimikin hyvin, sillä lopullisen sovelluksen testauksessa löytyi hyvin vähän sivujen teknisestä toteutuksesta johtuvia virheitä.

Sovelluksen teknisestä toimivuudesta haettiin pienempiä osakokonaisuuksia, joita testattiin sovelluksen kehittämisen aikana yhtenäisinä osina. Tämä helpotti varsinaista sovelluksen kokonaisuuden testausta, sillä lopullisessa toteutuksessa yksittäisiä sivuja on paljon, ja niiden kaikkien järjestelmällinen läpikäyminen on erittäin aikaavievää ja työlästä. Koska sovelluksesta oli erotettu osakokonaisuuksia, voitiin sovelluksen testaamisessa luottaa pienempien palasten yhteentoimivuuteen. Tällöin koko sovelluksen testauksessa voitiin keskittyä siihen, että näiden osa-alueiden väliset rajapinnat toimivat luotettavasti.

Rajapintojen testaus tarkoittaa lähinnä oikeiden parametrien välittämistä sivujen välillä. Virheelliset parametrit aiheuttivat poikkeuksetta kyseisen JSP-sivun kaatumisen tai selkeästi virheellisen toiminnon. Aktiivisen testauksen tuloksena nämä rajapintojen väliset virheet saatiin selvitettyä ja korjattua projektin aikataulussa.

15.3 Testaustulokset

Testaustapausten perusteella sovellus näyttää täyttävän sille asetetut toiminnalliset vaatimukset lähes kokonaisuudessaan. Sovellukseen ei jäänyt lainkaan sellaisia virheitä, jotka selkeästi aiheuttaisivat sovelluksen kaatumisen. Virheellisiä toimintoja on hyvin vähän. Nämä virheet on kuvattu luvussa 16.

Sovelluksen käytettävyyttä tulisi vielä testata sellaisella käyttäjäjoukolla, joka ei ole ollut mukana sovelluksen kehittämisessä. Näin saataisiin esille käyttäjien kannalta loogiset virheet esimerkiksi lomakkeiden sijoittelussa. Erityisesti ohjaajan käyttöliittymän puolella ainakin opiskelijoiden arvostelusivujen ja työtehtävien liittämisen sivusta tulisi saada palautetta.

16 Ohjelman toiminnassa havaitut virheet

Luvussa kuvataan sovellukseen KOPPELO-projektin puitteissa jääneitä virheellisiä toimintoja. Suurin osa virheistä korjattiin jo sovelluksen kehittämisen aikana ja tämän vuoksi luvussa esitellyt virheet ovat suurimmaksi osaksi sellaisia ohjelman osia, joita ei ehditty toteuttaa lainkaan projektin aikataulun puitteissa.

Kehitysaikana havaittuja virheitä on lueteltu projektin aikana käydyissä sähköpostiviesteissä. Viestit on tulostettu projektikansioon.

16.1 Haku-toiminnon virheet

Hakusivun (`search.jsp`) avainsanahakuun voi syöttää vain yhden hakusanan. Useampien hakuehtojen syöttäminen pilkuilla erotettuina ei siis onnistu. Jatkokehityksen pohdittavaksi jää, onko kyseinen toiminto tarpeellinen.

Lisäksi mistään ei tule ilmi, että opiskelija voi hakea keskeneräisistä opinnäytteistä vain niitä, joihin hän itse liittyy. Tämä saattaa aiheuttaa sekaannusta.

16.2 Virheet käyttäjän syötteissä

Käyttäjän antamien syötteiden oikeellisuutta ei tarkasteta missään yhteydessä. Tietokantaan tallennettavien päivämäärien ja kellonaikojen tapauksessa tosin tarkastetaan, että syötteet ovat oikeassa mudossa, mutta arvojen järkevyyteen ei muuten puututa. Siten käyttäjä voi tallentaa esimerkiksi tapahtuman tuhannen vuoden päähän.

Käyttäjä ei voi antaa aihe-ehdotusta, jolla ei ole otsikkoa. Tosin myöhemmissä opinnäytteen vaiheissa, joissa käyttäjä pääsee muokkaamaan mm. otsikkotietoa, ei enää tarkasteta, että käyttäjä antaisi opinnäytteelle otsikon.

16.3 Selaimiin liittyvät virheet

Joillakin selaimilla, esimerkiksi Windows-käyttöjärjestelmän Operalla, käyttäjän jo tekemät valinnat ja syöttämät arvot häviävät, jos sivu ladataan uudelleen. Tämä on selainriippuvainen ongelma, eikä siihen ole löytynyt ratkaisua.

17 Jatkokehitykseen siirretyt ominaisuudet

Luvussa kuvataan KOPPELO-sovellukseen liittyviä parannusehdotuksia, joita projektin neljän kuukauden aikataulussa ei ehditty toteuttamaan. Osa jatkokehittäjille siirretyistä tehtävistä liittyy täysin sovelluksen ulkoasun muokkaamiseen (luvut 17.1–17.5), mutta joukossa on myös joitain sovelluksen toimintaan ja rakenteeseen liittyviä kehitysideoita (luvut 17.6–17.11). Lukuun 17.12 on kirjattu jo projektin alussa esitetyt toiveet, joita projektin aikataulussa ei ehditty toteuttamaan.

17.1 Syöttökentät ja painikkeet

Yleisenä ongelmana sovelluksen sivuilla on eri tyyppisissä **syöttökentissä käytetyt kirjasintyypit**. Kirjasintyypin yhdenmukaistaminen on jätetty jatkokehittäjän tehtäväksi. Tämä ongelma esiintyy kaikilla sivuilla, joilla esitetään käyttäjälle lomakkeita.

Joillain sovelluksen sivuilla olisi tarvetta sijoittaa **tilaa painikkeiden ympärille** siten, että ne eivät olisi aivan kiinni selaimen alareunan tilapalkissa tai toisissaan. Tämä ulkoasuun liittyvä hienosäätö on jätetty jatkokehittäjille.

17.2 Sivujen sisällön kehittäminen

Aihe-ehdotuksen hyväksymissivulta `acceptProposition.jsp` (luku 9.5) **lähetettävästä sähköpostiviestistä** näkyy käyttäjälle vain ilmoitus viestin lähettämisestä. Tämän tilalle voisi kehittää kattavamman sivun, jossa esitettäisiin ainakin lähetetyn viestin sisältö.

Aiheet ja haku -sivulla ei tällä hetkellä ole juurikaan sisältöä. Sivulle voisi esimerkiksi listata navigaatiopalkin sisältämät linkit ja yleisohjeita sivujen merkityksestä tai näyttää omien aihe-ehdotusten tilanne. Sivun varsinaisen sisällön miettiminen jätettiin kuitenkin jatkokehittäjän harteille.

Haun tulossivulla voisi **kehittää sivun otsikoita** vastaamaan annettuja hakuehtoja. Tämä loisi käyttäjälle mielikuvan eri sivuista ja auttaisi hahmottamaan sivun tarkoitusta. Lisäksi hakusivulla tai haun tulossivulla voitaisiin esittää linkki raportin näyttämiseksi selainikkunassa.

Varattavissa olevien opinnäytteiden aiheiden luettelossa (katso luku 9.9) pitäisi olla **erilliset linkit** aiheen tietojen katseluun ja opinnäytteen aloittamiseen. Linkki opinnäytteen aloittamiseen pitäisi olla myös aiheen tietojen tarkastelun sivulla (katso luku 12.1), mikäli käyttäjällä on oikeus aloittaa opinnäytteen suoritus. Varattavien aiheiden luettelossa esitetylle **poistolinkille** tulisi etsiä **loogisempi paikka**.

17.3 Opinnäytteiden tyyppien mukaan lajitteleminen

Ainakin sivuille *Omien opinnäytteiden hallinta* (`main.jsp`, luku 11.1) ja *Varattavissa olevat aiheet* (`available.jsp`, luku 9.9) tarvitaan opinnäytteiden lajitteleminen opinnäytteen tyypin mukaan erillisten otsikoiden alle. Sama ongelma esiintyy kaikissa yhteyksissä, joissa sivulla esitetään paljon tietoa eri opinnäytetyypeistä. Tämän toteuttaminen voi olla hankalaa joissain yhteyksissä ja ajanpuutteen vuoksi se jätettiin jatkokehittäjän huoleksi.

Pitkiin listauksiin liittyy myös tarve voida **rajata** esimerkiksi varattavissa olevien opinnäytteiden **luetteloa joillain hakuehdoilla** kyseisillä sivuilla. "Staattisilla" sivuilla (kuten varattavissa olevien aiheiden sivu `available.jsp` luvussa 9.9) tämä ei ole suuri ongelma, mutta esimerkiksi hakutuloksena saatuun hyväksyttyjen aiheiden sivuun voi olla todella työlästä lisätä uusia hakuehtoja. Sama ominaisuus saadaan sovelluksessa joka tapauksessa aikaiseksi suorittamalla uusi, erillinen haku.

Nykyisessä toteutuksessa sivulla *Varattavissa olevat aiheet* `available.jsp` (luku 9.9) tarvittaisiin **pilkku erottamaan opinnäytetyypit** listauksessa. Tämän toteuttamiseen tarvitaan kuitenkin jonkinlainen kikkailu apumuuttujien kanssa. Tällöin pitää nimittäin selvittää, onko halutuloksissa vielä lisää tuloksia, eli tulostetaanko pilkku yksittäisen opinnäytetyypin jälkeen vai ei. Ajanpuutteen vuoksi tämäkin yksityiskohta jätettiin toteuttamatta. Mikäli varattavien aiheiden sivun toteutuksessa käytetään opinnäytetyypin mukaista lajittelua, ei pilkulla erottelua luonnollisesti tarvita.

17.4 Opinnäytteeseen liittyvät henkilöt

Omien opinnäytteiden hallinta -sivulle (katso luku 11.1) olisi etenkin ohjaajan puolella tarvetta saada luettelo opinnäytteeseen liitetyistä opiskelijoista. Teknisesti tämä on helpostikin toteutettavissa, mutta ongelmana on sivun fyysisen tilan rajallisuus. Jo nyt listauksiin omista ohjattavista opinnäytteistä on otettu runsaasti sisältöä, joten henkilöluettelon paikan miettiminen on jätetty jatkokehittäjän tehtäväksi. Tilaa luetteloon saadaan esimerkiksi poistamalla opinnäytteen tyyppi ja siirtämällä tämä tieto otsikoksi.

Opinnäytteen tietojen sivulla (luku 11.5) harkittiin, pitäisikö opinnäytteeseen liittyvistä **henkilöistä** tarjota **muutakin tietoa kuin sähköpostiosoite**. Esimerkiksi puhelinnumerolla ja kotiosoitteella voisi joissain tilanteissa olla käyttöä ainakin opinnäytten ohjaajille. Toisaalta kaikki käyttäjät eivät varmasti halua, että heidän tietonsa on saatavilla sovelluksen kautta, joten tämän ominaisuuden toteuttaminen jätettiin jatkokehittäjän harkinnan varaan.

17.5 Opinnäytteen tiedot ja tapahtumat

Opinnäytteen tiedot ja tapahtumat -sivulla `assignment.jsp` (katso luku 11.5) esitetään tällä hetkellä hyvin paljon asiaa. Tilaaja ehdotti useasti sivun jakamista vähintään kahteen osaan. Tapahtumien hallintaanhan on jo olemassa erillinen sivu `assignmentEvent.jsp` (luku 10.14).

Sivun kahtiajakoon liittyy kuitenkin se ongelma, että opinnäytteen eri vaiheissa työn tekijä on kiinnostunut sivun eri osa-alueista. Opinnäytteen aloittamisvaiheessa otsikko- ja kuvaustiedot ovat kiinnostavia, kun taas opinnäytteen käynnistyttyä mielenkiinto kohdistuu tapahtumiin ja edelleen opinnäytteen valmistuessa opiskelija haluaa nähdä pääasiassa arviointitietoja. Tämän pohjalta sivun jakamista päätettiin lykätä, kunnes käyttäjiltä saadaan toivomuksia sivujen sijoittelun suhteen.

Opinnäytteeseen liittyvien **tapahtumien sivulla** (luku 10.14) tulisi näkyä esimerkiksi eri värillä tai vahvennetulla kirjasimella **korostettuna tarkasteltavana oleva tapahtuma** kaikkien tapahtumien luettelossa. Myös tapahtuman tietoihin vievä linkki voisi puuttua kyseisen tapahtuman kohdalta. Tämä auttaisi käyttäjää hahmottamaan kokonaisuutta paremmin. Edelleen samalla sivulla voisi päivämäärän ja kellonaikojen esityksen **tiivistää yhdelle riville**.

17.6 Syötteiden tarkastaminen

Päivämääriä syötettäessä käyttäjä voi antaa päivämäärän joko menneisyydestä tai tulevaisuudesta. Päivämäärän formaatin oikeellisuuden tarkastaa `time`-papu. Päivämäärän järkevyyttä ei kuitenkaan tarkasteta missään yhteydessä, joten tämä on jatkokehittäjille jätetty korjattava ominaisuus. Päivämääriin liittyviä rajoja täytyy tosin pohtia vielä sovellusta käytettäessä, sillä toisaalta on järkevää pystyä kirjaamaan sovellukseen mm. menneitä ja tulevia tapahtumia.

Päivämääriin liittyviä ongelmia on ainakin sovelluksen sivuilla `propose.jsp` (luku 9.1), `acceptProposition.jsp` (luku 9.5), `assignmentUpdate.jsp` (luku 10.3), `assignmentGrade.jsp` (luku 10.7) ja `addEvent.jsp` (luku 10.12).

Kaikkiin tekstikenttiin liittyy se ongelma, että käyttäjä voi kirjoittaa kenttiin haluamiaan arvoja. Ominaisuutta oikein käytettämällä käyttäjä voi lisätä esimerkiksi opinnäytteen kuvaustekstiin HTML-tageja korostamaan tiettyjä kohtia. Toisaalta käyttäjä voi kirjoittaa kenttiin esimerkiksi **JavaScript-komentoja**, jolloin opinnäytettä tarkastelevalta käyttäjältä vähintään sulkeutuu selainikkuna tai hänet ohjataan toiselle sivustolle. Tämä on sovelluksessa suuri tietoturva-aukko, jota ei kuitenkaan ehditty huomioida aikataulun puitteissa.

17.7 Tietokantaan vaikuttavat kehitysideat

Aihe-ehdotukselle olisi hyvä pystyä tallentamaan **useita ehdottajia**. Tällä hetkellä useat ehdottajat voidaan kirjoittaa aihe-ehdotuksen kuvaukseen, mutta tarkoitus olisi saada tämä tieto omaksi elementikseen. Tietokanta ei kuitenkaan mahdollista useiden ehdottajien tallentamista, joten tämän ominaisuuden toteuttaminen on jätetty sovelluksen jatkokehitykseen.

Opinnäytteen työtehtävien hallinnoimisen sivuilla (10.6) tulisi näkyä **opinnäytteeseen käytetyn työmallin nimi**. Nykyisen tietokantarakenteen perusteella on mahdotonta päätellä yksikäsitteisesti, mitä mallia työtehtävien asettamiseen on käytetty (vai onko käytetty lainkaan). Tämän ominaisuuden lisääminen jää jatkokehittäjille. Työmallien käyttämisen ideologiaa voidaan joutua myös kokonaan muokkaamaan, mutta sivun käytettävyydestä kannattanee odottaa käyttäjien palautetta.

Hyväksyttyjen opinnäytteiden sivulla eli haun tuloksena saaduissa omissa hyväksytyissä opinnäytteissä voisi hyväksymispäivämäärän lisäksi näkyä myös opinnäytteen hyväksyjä. Tämä vaatisi muutoksen sekä hakusivun `searchResult.jsp` (luku 11.4) toimintaan että tietokantaan. Tällä hetkellä tietokannassa ei ole mitään mahdollisuutta merkitä opinnäytteen hyväksyjää.

17.8 Sovellukseen tarvittavat lisäsivut

Hyväksytyn aihe-ehdotuksen pitäisi näkyä aiheen ehdottajalle myös hyväksymisen jälkeen KOPPELO-sovelluksen sivuilla. Tällä hetkellä käyttäjä näkee aihe-ehdotuksen listauksessa vain hyväksymistä vailla olevat aihe-ehdotuksensa, joten joko kyseisen sivun (`propositions.jsp`, luku 9.4) toimintaa tulee muuttaa tai sitten tiedon pitäisi näkyä käyttäjän pääsivulla `main.jsp` (luku 11.1).

Tapahtumaa lisättäessä tiedot tallennetaan kantaan ilman, että käyttäjällä on mahdollisuutta korjata syötettään lomakkeen lähettämisen jälkeen. Käyttäjälle tulisi kuitenkin tarjota mahdollisuus muuttaa tapahtuman tietoja ainakin tapahtuman kirjaamissession ajan ja mahdollisesti myös myöhemmin.

17.9 Sovelluksen toiminnan epäloogisuuksia ja kehitysideoita

Mikäli opiskelijat eivät ole lisänneet opinnäytteeseen tapahtumia, ohjaaja voisi lähettää heille muistutussähköpostiviestin, vaikka asetettu muistutusväli ei olisi vielä kulunutkaan loppuun.

Sovelluksessa voi **hyväksyä opinnäytteitä, joihin ei liity opiskelijoita**. Tämä ominaisuus tulisi poistaa sovelluksesta. Hyväksymispäivämäärän asettamisen kohta voitaisiin jopa poistaa kokonaan opinnäytteen tietojen muuttamisen lomakkeelta, kunnes työn vaiheeksi asetetaan `Hyväksytty`.

Opiskelijoiden opintoviikkomääräksi voi antaa kuinka suuren arvon tahansa. Järjestelmään voisi **määrätä opintoviikoille asetettavan ylärajan** esimerkiksi opinnäytteeseen liitetyn kurssin tiedoista.

Sovelluksessa tapahtuvista **automaattisista toiminnoista** (`autoMatik.jsp`, luku 14.2) voitaisiin kirjata erillinen **lokitiedosto**, jossa selvitettäisiin erilaisten tapahtumien onnistuminen ja epäonnistuminen.

17.10 Automaattisten sähköpostiviestien puutteista

Sovelluksen lähettämät automaattiset sähköpostiviestit sisältävät kaiken käyttäjän tarvitseman tiedon, mutta niiden ulkoasua voisi selkeyttää ja yhtenäistää.

Kun aihe-ehdotuksia ja aiheita aiotaan poistaa järjestelmästä vanhentuneina, lähetetään asiasta tieto viikon ajan aiheen ehdottajalle. **Ilmoitus varsinaisesta poistamisesta** tulisi lähettää erityisesti silloin, kun järjestelmästä poistetaan hyväksytty, erääntynyt aihe. Poistamissähköpostin lähettäminen vaatii jonkin verran muokkausta sivulla `autoMatik.jsp` (luku 14.2).

17.11 Navigaatiopalkit

KOTKA-järjestelmän yleinen ongelma on navigaatiopalkin toiminta niissä tilanteissa, kun käyttäjä etenee sovelluksessa sellaiselle sivulle, jota ei ole merkitty navigaatiopalkkiin. Tällöin navigaatiopalkki "romahtaa kasaan". Tällöin käyttäjä ei voi navigaatiopalkista havaita, missä osassa sovellusta hän työskentelee. Navigaatiopalkin toiminnan korjaaminen kuuluu ehdottomasti sovelluksen jatkokehittäjän tehtäviin.

Toisaalta myös KOPPELO-sovellus tarvitsisi oman **navigaatiolistan esittämään linkkejä** opinnäytteen **hallintaan liittyvien sivujen välillä**. Nykyisessä toteutuksessa navigaatiolista ja siihen liittyvät oikeellisuustarkastukset tehdään jokaisella listan tulostavalla sivulla erikseen sekä hoidetaan tulostus vieläpä kahteen kertaan (ylä- ja alapalkkia varten)! Tämä toiminta pitäisi ehdottomasti siirtää navigaatiolistan muodostavan pavun sisään.

17.12 Muita jatkokehitysideoita

KOPPPELO-projektin puitteissa ei luotu minkäänlaista yhteyttä **KOPPELO- ja KOLIBRI-sovellusten välille**. Järjestelmän kokonaistoiminnan kannalta olisi kuitenkin suotavaa, että KOPPELO-sovelluksessa ilmoitetut tapaamiset näkyisivät myös käyttäjien henkilökohtaisissa kalentereissa. Yhteyden toteuttaminen vaatinee sopivan hakuehdon lisäämistä KOLIBRIN näkymät hakevaan SQL-lauseeseen.

Projektin alkuvaiheessa pohdittiin **KOTKA-järjestelmän tietokannan hajauttamista** useille eri palvelimille. Tietokannan hajauttaminen huomioitiin KOPPELO-sovelluksen tietokantasuunnitelmissa siten, että opinnäytekokonaisuuden muodostaviin tauluihin pyrittiin asettamaan mahdollisimman vähän yhteyksiä muuhun KOTKA-järjestelmän tietokantaan.

Varsinaisiin tietokannan hajautustekniikoihin ei projektin puitteissa kuitenkaan ehditty tutustua ja siten varsinainen hajauttaminen jää järjestelmän jatkokehittäjän vastuulle.

18 Asennusohjeet

Luvussa kuvataan KOPPELO-sovelluksen asentamiseen liittyviä toimenpiteitä. Luvussa esitellyt tiedostot löytyvät KOPPELO-projektin CD-levyltä.

Koska sovellus on olennaisesti KOTKA-järjestelmään liitettävä moduuli, oletetaan, että KOTKA-järjestelmän kurssikirjanpito-osa KORPPI on jo asennettu. KORPPI-sovelluksen asennusohje on liitetty KORPPI-projektin projektikansioon. KOPPELO-sovellus ei kuitenkaan vaadi KOLIBRI-sovelluksen asentamista.

Luvussa esitellyt tiedostot löytyvät KOPPELO-projektin CD-levyltä.

18.1 Tietokannan päivitys

Tietokannan luomiseen tarvittavat SQL-lauseet on koottu yhteen tiedostoon `koppelo.sql`. Luontilauseet voidaan suorittaa periaatteessa millä tahansa tietokannan käsittelyyn tarkoitetulla hallintajärjestelmällä. KOPPELO-sovelluksen kehittämisen aikana kannan luomisen lauseet ajettiin PostgreSQL-sovelluksen komentorivikäyttöliittymällä.

Tietokantaan asetettavaa valikkomuotoista tietoa on koottu tiedostoon `aineisto.sql`. KOTKA-järjestelmän tietokannan taulujen sisältöön tarvittavat lisäykset on kirjoitettu tiedostoon `kotkaupdates.sql`. Nämä tiedostot voidaan myös ajaa sopivalla tietokannanhallintasovelluksella.

KOPPELO-sovelluksen käyttämät tietokannan taulut voidaan poistaa kannasta ajamalla tiedoston `koppelodrop.sql` sisältämät lauseet.

18.2 Tagien asentaminen

KOPPELO-sovellus käyttää Jakarta-projektin `dbtags`-kirjastoa. Se on SQL-kyselyjen suorittamiseen ja hakutulosten hallintaan tarkoitettu tag-kirjasto. Kyseinen kirjasto asennetaan kopioimalla kirjaston kuvaustiedosto sovelluksen `WEB-INF`-alihakemistoon ja kirjaston luokat sisältävä JAR-paketti sovelluksen `WEB-INF/lib`-alihakemistoon.

Tämän jälkeen tulee lisätä uusi `<taglib>`-elementti `WEB-INF/web.xml`-tiedostoon, jossa on kuvattu sovelluksen käyttämät tag-kirjastot. Seuraava esimerkki antaa käsityksen `<taglib>`-elementin määrittelystä:

```
<taglib>
  <taglib-uri>http://jakarta.apache.org/taglibs/dbtags</taglib-uri>
  <taglib-location>/WEB-INF/classes/koppelotags/koppelo.tld</taglib-location>
</taglib>
```

Kirjastoa voi käyttää myös viittaamalla JSP-sivuilla suoraan kirjaston kuvaustiedostoon, jolloin `<taglib>`-elementin määrittely on tarpeetonta.

Jakarta-projektin tuottama englanninkielinen asennusohje löytyy osoitteesta `http://jakarta.apache.org/taglibs/doc/dbtags-doc/`.

18.3 Papujen asentaminen

Sovelluksessa käytetyn oikeuspavun `KoppeloRights` lähdekoodi on kirjoitettu tiedostoon `KoppeloRights.java`. Tämä tiedosto kannattaa kääntää uudelleen asennusympäristössä esimerkiksi `javac`-kääntäjällä. Käännöksen tuloksena saatu `KoppeloRights.class`-tiedosto sisältää suoritettavan koodin. Tämä tiedosto tulee sijoittaa samaan paikkaan kuin muut KORPPI-sovelluksen pavut on sijoitettu. Papujen sijoituspaikkana on yleensä käytetty hakemistoa `/kotka/WEB-INF/classes/kotkabeans/`. Papujen lähdekoodit on sijoitettu alihakemistoon `src`.

Muokatun `HTMLMenu`-pavun lähdekoodi on tiedostossa `HTMLMenu.java`. Myös tämä papu kannattaa kääntää uudelleen asennusympäristössä. Käännöksen tuloksena saatu `HTMLMenu.class`-tiedosto korvaa KORPPI-projektin toteuttaman pavun käännöksen.

18.4 JSP-sivujen asentaminen

Sovellukset JSP-sivut tarvitsevat toimiakseen yhteisiä tiedostoja `beans.inc`, `constants.inc`, `header.inc`, `menus.inc`, `dbconnect.inc` ja `pageEnd.inc`. Nämä tiedostot tulee sijoittaa hakemistoon `/kotka/shared/`. Tiedostojen asentaminen ei vaadi erityisiä toimenpiteitä. Näiden tiedostojen kuvaukset löytyvät luvusta 6.1.

KOPPELO-sovelluksen suoritettavat JSP-sivut (luvut 9–14) asennetaan hakemistoon `/kotka/koppelo/`. Sivut ovat suorituskelpoisia sellaisenaan. TomCat-servlettimoottori kääntää JSP-sivut silloin, kun niihin kohdistuu pyyntöjä.

18.5 Muut toimenpiteet

Sovelluksen käyttämiseksi täytyy sovelluksella antaa joitain oikeuksia eri KOTKA-järjestelmän käyttäjäryhmille, opinnäytteen rooleille ja yksittäisille käyttäjille. Oikeuksien antamista on kuvattu tarkemmin luvussa 19.

Jotta sivujen käyttö nopeutuisi hieman, kannattaa sovelluksen asentajan käydä jokaisella sovelluksen JSP-sivulla. Tällöin TomCat-servlettimoottori kääntää sivut ja seuraavalle käyttäjälle ei tarvitse tehdä uutta käännöstä.

19 Ylläpitäjän ohjeet

Luvussa kuvataan KOPPELO-sovelluksen ylläpitäjän huomioitavia asioita. Erityisesti sovelluksen asennukseen liittyy joitain ylläpitäjän vastuulle asetettuja tehtäviä.

19.1 Oikeuksien asettaminen

Sovelluksen asentamisen jälkeen ylläpitäjän täytyy antaa käyttäjille, käyttäjäryhmille ja opinnäytteen rooleille oikeuksia toimia KOPPELO-sovelluksessa. Ilman oikeuksien asettamista sovelluksen käyttö on hyvin rajoitettua. Taulukoissa 5 ja 6 on kuvattu suositukset käyttäjäryhmille ja rooleille asetettavista oikeuksista.

Käyttäjäryhmä	**Oikeus**
Ylläpitäjä	Aiheen hyväksymisoikeus Henkilöiden päivitysoikeus Opinnäytteen tietojen katseluoikeus Opinnäytteen tietojen päivitysoikeus Opiskelijan arvosteluoikeus Keskeneräisten opinnäytteiden katseluoikeus Opinnäytteen kiinnitysoikeus
Johtaja ja sihteerit	Opinnäytteen tietojen katseluoikeus Keskeneräisten opinnäytteiden katseluoikeus
Luennoitsija	Opinnäytteen tietojen katseluoikeus Keskeneräisten opinnäytteiden katseluoikeus Opinnäytteen kiinnitysoikeus

Taulukko 5: Käyttäjäryhmien oikeudet.

Rooli	**Oikeus**
Opiskelija	Opinnäytteen tietojen katseluoikeus
Ohjaaja	Henkilöiden päivitysoikeus Opinnäytteen tietojen katseluoikeus Opinnäytteen tietojen päivitysoikeus Opiskelijan arvosteluoikeus
Tarkastaja	Opinnäytteen tietojen katseluoikeus Opiskelijan arvosteluoikeus Opinnäytteen arviointioikeus

Taulukko 6: Opinnäytteen roolien oikeudet.

Oikeusjakoa voi muokata ylläpidon käyttöliittymän sivuilla, joiden tekninen toteutus on kuvattu luvussa 13. Sivujen käyttöohje on erillisenä dokumenttina projektikansiossa.

19.2 Valikkotekstien ylläpito

Sovelluksen ylläpitäjä määrää sovelluksen valikoissa esiintyvien tietojen sisällöstä ja monikielistämisestä. Näiden tietojen hallinnoimiseen ei ole tehty erillistä käyttöliittymää, vaan ylläpitäjän on tarkoitus syöttää haluamansa tieto suoraan tietokantaan.

Valikkotekstejä on taulukossa 7 kuvatuissa tietokannan tauluissa. Tietoihin liittyvät käännökset ovat erillisissä käännöstauluissa. Tietokannan rakenteen hahmottamisessa auttaa luvun 5 kuvassa 2 esitetty ER-kaavio.

Taulu	**Kuvaus**
`AssignmentType`	sisältää luettelon opinnäytetyypeistä.
`StudyLine`	sisältää luettelon eri laitosten opintolinjoista.
`AssignmentRole`	sisältää luettelon opinnäytteiden rooleista.
`AssignmentPhase`	sisältää luettelon opinnäytteiden ja niiden työtehtävien vaiheista.
`AssEventType`	sisältää luettelon opinnäytteiden tapahtumien tyypeistä.
`AssignmentDetail`	sisältää opinnäytteiden, aiheiden ja aihe-ehdotusten tiedot.
`AssignmentKeyword`	sisältää sovelluksessa opinnäytteisiin, aiheisiin ja aihe-ehdotuksiin liitetyt avainsanat.
`AssignmentTask`	sisältää luettelon opinnäytteiden työtehtävistä.
`AssTaskCategory`	sisältää luettelon työtehtävien luokitteluista.
`AssTaskTemplate`	sisältää luettelon työtehtävien mallipohjien nimikkeistä.

Taulukko 7: Valikkotekstit tietokannan tauluissa.

Ylläpitäjän vastuulla on asettaa tietokantaan valikkomuotoisten tekstien käännökset asianmukaisiin käännöstauluihin.

20 Yhteenveto

KOPPELO-projekti toteutti Jyväskylän yliopiston tietotekniikan laitokselle KOPPELO-nimisen opinnäytteiden hallintasovelluksen, jolla pystytään keskitetysti hallitsemaan laitoksella tehtävien opinnäytteiden vaiheita. KOPPELOn sovellusraportissa kuvataan sovelluksessa toteutetut ominaisuudet. Sovelluksen rakenne kuvataan tarkasti sekä laitteisto-, ohjelmisto- että toteutustasoilla. Sovellusraporttiin on lisäksi koottu toteutetussa sovelluksessa havaitut virheet. Toteuttamatta jääneet osat on kuvattu sovellusraportissa jatkokehitysideoina.

KOPPELO-sovelluksessa on toteutettu kaikki opiskelijan ja ohjaajan toiminnalliset vaatimukset. Sovelluksella voi esittää ja hyväksyä aihe-ehdotuksia, aloittaa opinnäytteen suorituksen ja seurata opinnäytteen etenemisestä. Sovelluksella voi hakea useilla eri hakukriteereillä opinnäytteitä ja aiheita, joten sovelluksella voidaan tuottaa erilaisia raportteja laitokselle tehdyistä opinnäytteistä. Sovelluksen oikeuksien hallintaan toteutettiin ylläpidolle erillinen käyttöliittymä. KOTKA-järjestelmän ulkopuolisille käyttäjille ei toteutettu mahdollisuutta tutkia laitoksilla valmistuneita opinnäytteitä.

Jatkokehitykseen siirrettiin useita sovelluksen ulkoasua koskevia parannusehdotuksia. Haku- ja raporttitoimintoa sekä automaattisia sähköpostiviestejä voitaisiin edelleen kehittää.