

KIURU

Tietotekniikan sovellusprojekti

Toni Hilpinen
Marko Koivuniemi
Jussi Mäkinen
Miika Nurminen

Sovellusraportti
28.1.2003

Jyväskylän yliopisto
Tietotekniikan laitos

Kiuru-projektin tietoja

Tekijät:

Toni Hilpinen (hilpinen@cc.jyu.fi)
Marko Koivuniemi (koimark@cc.jyu.fi)
Jussi Mäkinen (jusmaki@cc.jyu.fi)
Miika Nurminen (minurmin@cc.jyu.fi)

Yhteystiedot: Työtila AgC222.2, sähköpostilistojen osoitteet
kiuru@korppi.jyu.fi ja kiurulaaja@korppi.jyu.fi.

Työn nimi: Sovellusraportti, Kiuru-projekti.

Työ: Sovellusraportti tietotekniikan sovellusprojektiin.

Tiivistelmä: Kiuru-projekti toteutti Jyväskylän yliopiston tietotekniikan laitoksella kehitettyyn Korppi-järjestelmään WWW-pohjaisen varausosion sekä varaus-tietojen synkronoinnin Kotka- ja Timmi-järjestelmien välillä. Sovellusraportissa kuvataan määriteltyjen vaatimusten toteutumista sekä sovelluksen toteutusta ja toimintaa.

Avainsanat: Salivaraus, kurssien opetustapahtumat, resurssivaraus, Korppi-järjestelmä, Timmi-järjestelmä, WWW-sovellus.

Versionhallinta

Versio	Päiväys	Tehnyt	Muutokset
0.1	8.1.2003	Jussi Mäkinen	Ensimmäinen versio
0.2	13.1.2003	Miika Nurminen	Käyttöliittymäkuvaus
0.3	14.1.2003	Marko Koivuniemi	Sovelluksen puutteet ja jatkokehitys
0.4	16.1.2003	Miika Nurminen	Java-pavut
0.5	17.1.2003	Marko Koivuniemi	Lokitietoja, viimeistelyä.
0.6	21.1.2003	Marko Koivuniemi	Korjauksia JP Santasen kommenttien pohjalta
0.7	22.1.2003	Jussi Mäkinen	Pieniä korjauksia
0.8	22.1.2003	Marko Koivuniemi	Viimeistelyä
0.9	23.1.2003	Jussi Mäkinen	Viimeistelyä
0.95	23.1.2003	Marko Koivuniemi	Viimeistelyä
1.0	28.1.2003	Marko Koivuniemi	Viimeistelyä

Sisältö

1	Johdanto	1
2	Termejä	2
2.1	Aihepiirin yleisiä termejä	2
2.2	Teknisiä termejä	3
3	Asetettujen tavoitteiden toteutuminen	4
3.1	Timmi-rajapinta ja tietojen synkronointi	4
3.2	Varausmoduulin käyttöliittymä	5
3.3	Ylläpitäjän ja päävahvistajan käyttöliittymä	5
3.4	Resurssipyynnöt	6
3.5	Sovelluksen monikielisyys	6
3.6	Yhteenveto sovelluksen ominaisuuksista	6
4	Palvelinohjelmistot, toteutustekniikat ja työkalut	7
4.1	JSP-tekniikka	7
4.2	JSP-sivujen käsittelylogiikka	8
4.3	Java-pavut	9
4.4	Tietokanta	9
5	Sivujen käyttöliittymän yleisrakenne	11
6	JSP-sivut	12
6.1	Kalenteri	12
6.1.1	Salin viikkokalenteri spaceweek.jsp	12
6.1.2	Muutokset tapahtuman lisäyssivulle addAppointment.jsp	13
6.2	Opettajan käyttöliittymä	14
6.2.1	Muutokset sivulle modifyGroup.jsp	14
6.3	Varausjärjestelmä	15
6.3.1	myReservations.jsp ja deleteReservations.jsp	15
6.3.2	searchFreeSpace.jsp ja newSpace.jsp	16
6.3.3	searchResults.jsp	17
6.3.4	makeReservation.jsp ja commitReservation.jsp	18
6.3.5	show- ja modifyReservationRequests.jsp	19
6.3.6	reportInit.jsp ja reportHandler.jsp	20
6.4	Ylläpitäjän käyttöliittymä	22
6.4.1	Muutokset sivulle admin.jsp	22
6.4.2	searchSpace.jsp ja searchSpaceHandler.jsp	23
6.4.3	modifySpace.jsp ja spaceHandler.jsp	24
6.4.4	addSpaceGroupToSpace.jsp	25

6.4.5	searchSpaceGroup.jsp	26
6.4.6	modifySpaceGroup.jsp ja spaceGroupHandler.jsp	27
6.4.7	addPersonToSpaceGroup.jsp	29
6.4.8	addSpaceToSpaceGroup.jsp	30
7	Java-pavut	31
7.1	kotkabeans-pakettiin tehdyt muutokset	31
7.1.1	cSQL.java	32
7.1.2	DB.java	32
7.1.3	Encoder.java	33
7.1.4	HTMLMenu.java	33
7.1.5	User.java	33
7.1.6	Tools.java	34
7.2	kolibribeans-pakettiin tehdyt muutokset	35
7.2.1	EventBean.java	35
7.2.2	EventBeanContainer.java	35
7.2.3	QueryContainer.java	35
7.2.4	Week.java	36
7.3	Kiurubeans-paketti	36
7.3.1	HtmlBean.java	37
7.3.2	StringPair.java	37
7.3.3	KiuruString.java	38
7.3.4	SimpleDb.java	39
7.3.5	StatementContainer.java	40
7.3.6	StatementItem.java	40
7.3.7	IntItem.java	40
7.3.8	StringItem.java	40
7.3.9	KiuruHandler.java	40
7.3.10	KiuruSpace.java	41
7.3.11	SpaceGroup.java	42
7.3.12	KiuruReservation.java	42
7.3.13	SearchHandler.java	43
7.3.14	SpaceSearch.java	43
7.3.15	PersonSearch.java	43
7.3.16	SpaceGroupSearch.java	44
7.3.17	TimmiParser2.java	44
7.3.18	Timmi2KiuruSynchronizer.java	45
7.3.19	TimmiLog2Kiuru.java	46
7.3.20	TimmiObject.java	46
7.3.21	TimmiContactPerson.java	47
7.3.22	TimmiCustomer.java	47

7.3.23	TimmiReservation.java	47
7.3.24	TimmiBuilding.java	47
7.3.25	TimmiSpace.java	48
7.3.26	TimmiUseType.java	48
8	Sovelluksen testaaminen	49
8.1	Testausympäristöt	49
8.2	Testausperiaatteita	49
8.3	Testaustulokset	50
9	Ohjelman virheet ja toiminnan puutteet	51
9.1	Navigointi	51
9.2	Varausmoduuli	51
9.3	Synkronointi ja lokitiedostojen käsittely	52
9.4	Virheet käyttäjän syötteissä	52
9.5	Selaimiin liittyvät virheet	52
10	Jatkokehitykseen siirretyt ominaisuudet	53
10.1	Tietokannan rakenteeseen liittyvät kehitysideat	53
10.2	Kehitysideoita sovelluksen toiminnasta	53
10.3	Automaattisten sähköpostiviestien puutteista	54
10.4	Muita jatkokehitysideoita	55
11	Asennusohjeet	56
11.1	Tietokannan päivitys	56
11.2	Papujen asentaminen	56
11.3	JSP-sivujen asentaminen	57
11.4	Muut toimenpiteet	57
12	Ylläpitäjän ohjeet	58
12.1	Oikeuksien asettaminen	58
12.2	Ristiriitaiset varaukset	58
12.3	Monikielisyys	58
13	Yhteenveto	59
	Lähteet	58

1 Johdanto

Kiuru-niminen tietotekniikan sovellusprojekti on jatkoa Kotka-, Korppi-, Kolibri- ja Koppelo-projekteilte. Kiuru toteutti Korppi-järjestelmään tilanvarausmoduulin ja rajapintayhteyden Yomi Oyj:n Timmi-järjestelmään. Kiuru-ryhmän kirjoittama koodi on julkaistu GPL-lisenssin alaisuudessa.

Syksyn 2002 Kiuru-projektiryhmään kuuluivat tietotekniikan opiskelijat Toni Hilpinen, Marko Koivuniemi, Jussi Mäkinen ja Miika Nurminen. Tilaajana toimi Jyväskylän yliopistolta hallintovirasto ja tietotekniikan laitos. Muut projektiorganisaatioon kuuluneet henkilöt esitellään Projektiraportissa.

Projektiryhmä on laatinut jo aiemmin Projektisuunnitelman, Sovellussuunnitelman, Vaatimusmäärittelyn, Haastatteluraportin ja Projektiraportin. Projektiraportissa kuvataan projektin taustoja ja projektille asetettuja tavoitteita, projektiorganisaatiota sekä suunnitelmien toteutumista ja projektin onnistumista.

Raportissa kuvataan sovelluksen arkkitehtuuria ja toteutettuja ominaisuuksia. Lisäksi esitetään sovelluksen toimintaa myös yksittäisten JSP-sivujen ja Java-papujen tasolla. Sovelluksen tarvitsemat tietokantamuutokset on kirjattu Kotka-tietokantaraporttiin.

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

Käyttöliittymä sekä tekniset kuvaukset yksittäisistä JSP-sivuista on esitetty luvuissa 5-6. Java-pavut esitellään luvussa 7. Sovelluksen testausta on lyhyesti kuvattu luvussa 8. Sovelluksesta havaitut virheet ja jatkokehitysideat on koottu lukuihin 9 ja 10. Sovelluksen asennusohjeet ja ylläpitäjän huomioita on kirjattu lukuihin 11 ja 12.

2 Termejä

Luvussa on kuvataan termejä, jotka auttavat ymmärtämään dokumentteja paremmin. Tekniset termit on pääosin listattu sellaisia henkilöitä varten, joilla ei ole ohjelmointitaustaa.

2.1 Aihepiirin yleisiä termejä

Seuraavat termit liittyvät salinvarausprosessiin:

Korppi	on Jyväskylän yliopistossa joidenkin laitoksien käyttämä opetus- ja tutkimushallintajärjestelmä.
Kotka	on Korppi-järjestelmän tietokanta.
Maksaja	on henkilö tai organisaatio, joka maksaa varauksen.
Oheisvarattava	on liikuteltava esine, jonka vahtimestarit toimittavat tilaisuuden järjestäjän toiveiden mukaisesti haluttuun saliin haluttuna ajankohtana.
Resurssi	on johonkin saliin liittyvä ominaisuus tai kiinteästi saliin asennettu laite tai esine. Esimerkiksi salin paikkamäärä tai piirtoheittimen olemassaolo voivat olla resursseja.
Resurssipyyntö	kuten varauspyyntö, kuitenkin sillä erotuksella, että salin sijaan ilmoitetaan resurssien tarve.
Sali	on varattavissa oleva paikka, johon ihmiset voivat kokoontua (esimerkiksi luentosali tai kokoustila).
Sali-järjestelmä	on Jyväskylän yliopistossa aiemmin käytössä ollut tekstipohjainen salienvarausjärjestelmä.
Timmi	on Yomi Applications Oy:n kehittämä tilanvaraussovellus.
Vakiovaraus	on sarja varauksia, jotka liittyvät toisiinsa ja toistuvat esimerkiksi viikoittain samana viikonpäivänä.
Varauksen pyytäjä	on henkilö, joka ehdottaa varattavaksi jotakin tilaa jonakin ajankohtana.
Varauksen vahvistaja	on henkilö, jolla on oikeus vahvistaa varaus tiettyyn tilaan.
Varaus	on käyttöoikeus tiettyyn tilaan tiettyinä ajankohtana.
Varauspyyntö	on varauksen pyytäjän tekemä ehdotus varaukseksi koskien tiettyä tilaa tiettyinä ajankohtana.

2.2 Teknisiä termejä

Seuraavat ohjelmointitekniset termit liittyvät sovelluksen taustalla olevaan järjestelmään:

AJP12	on protokolla tiedonsiirtoon Apache- ja Tomcatpalvelimien välillä.
Apache CSS	on ilmainen HTTP-palvelinohjelmisto. eli Cascading Style Sheets on WWW-sivujen ulkoasua kuvaava kieli.
HTML	on WWW-sivujen sisältöä kuvaava kieli.
HTTP	on WWW-arkkitehtuurin käyttämä tiedonsiirtoprotokolla.
Jakarta-Tomcat	on ilmainen servletti- ja JSP-moottori.
Java	on Sunin kehittämä laitteistoriippumaton olio-ohjelmointikieli.
JavaDoc	on Java-kehitysympäristön mukana tuleva dokumentointityökalu.
Java-pavut	(engl. <i>JavaBeans</i>) ovat Java-ohjelmointikielellä luotuja komponentteja, joita voidaan kutsua JSP-sivuilla.
JDBC	eli Java Database Connectivity on Java-teknologian käyttämä rajapinta erilaisiin tietokantoihin.
JSP	eli Java Server Pages on skriptaustyylinen ohjelmointikieli.
Keksi	(engl. <i>cookie</i>) on menetelmä saada tallennettua tietoja WWW-sivuilta käyttäjien koneille. Palvelin lähettää pieniä tietopaketteja selaimelle, jotka tallennetaan käyttäjän koneelle. Myöhemmin selain lähettää tiedot palvelimelle ja palvelin voi käyttää niitä esimerkiksi istuntojen tallentamiseen.
Käyttötapaus	(engl. <i>use case</i>) on käyttäjän tai sovelluksen toimintoa tietyn tehtävän suorittamiseksi kuvaava dokumentti.
Moduuli	on tarkkaan rajattu toiminnallinen ohjelmakokonaisuus.
Poolman	on ohjelma, joka kontrolloi tietokantayhteyksiä.
PostgreSQL	on ilmainen tietokannanhallintajärjestelmä.
Skripti	on yksinkertainen ohjelmanpätkä, joka useimmiten on tehty tulkittavalla kielellä.
Selain	on ohjelma, joka käyttäjän koneella tulkaa HTML-kieliset sivut esitettävään muotoon.
Servletti	(engl. <i>servlet</i>) on palvelimella sijaitseva sovellus, joka toteuttaa HTTP-palvelimen pyynnöstä tietyn toiminnon.
SQL	eli Structured Query Language on relaatiotietokantojen hallintaan kehitetty standardi kieli.

3 Asetettujen tavoitteiden toteutuminen

Kiuru-sovelluksen ensisijaisena vaatimuksena oli määrittellä ja toteuttaa liittymä Timmi-salivarausjärjestelmään Korppi-järjestelmästä. Lisäksi yleisenä vaatimuksena oli suunnitella ja toteuttaa helppokäyttöinen salien ja muiden tilojen varaussovellus Korppi-järjestelmän osaksi. Sovellus on tarkoitettu ensisijaisesti Jyväskylän yliopiston henkilökunnan käyttöön. Luvussa on esitetty sovellukselle määriteltyjen vaatimusten toteutuminen.

Projektin alussa ryhmälle annettiin yleisluontoinen kuvaus ohjelmalta toivotavista ominaisuuksista ja listaa tarkennettiin sovelluksen potentiaalisten käyttäjien haastattelujen pohjalta. Lisäksi Timmi-järjestelmän tuottaneen Yomi Oyj:n edustajien kanssa sovittiin Timmi-järjestelmän tarjoamasta rajapinnasta ja tietojen synkronoinnista järjestelmien välillä. Kiuru-projektin toteuttamista toiminnoista ja niiden suhteesta Timmi-järjestelmään saa käsityksen kuvasta 3 luvussa 5.

3.1 Timmi-rajapinta ja tietojen synkronointi

Tietojen synkronointi Timmi- ja Korppi-järjestelmien välillä oli yksi Kiuru-projektin tärkeimmistä tavoitteista. Kiuru-projektiin sisältyi Timmi-järjestelmän kehittäjän Yomi Oyj:n kanssa toteutettu Timmi-rajapinnan määrittelyvaihe. Määrittelyvaiheen aikataulua jouduttiin kuitenkin useasti korjaamaan ja lopullinen Yomi Oyj:n toteuttama rajapintamäärittely saatiin hyväksytyksi vasta projektin viimeisimmällä kolmanneksella.

Määrittelyn pohjalta projektiryhmä aloitti tarvittavien Korppi-järjestelmän komponenttien kehittämisen ja eri toteutusvaihtoehtojen läpikäynnin. Komponentit valmistuivat vaiheeseen, jossa niiden testausta rajapinnan kanssa voisi aloittaa.

Yomi Oyj toimitti osittain toimivan ja testattavan rajapinnan projektiryhmän käyttöön vasta viimeisten viikkojen aikana tammikuun alussa, jolloin kehityksen painopiste oli jo projektin viimeistelyssä ja loppudokumentoinnissa. Testauksessa tarvittavaa WinTimmi 2.0 -ohjelmaa projektiryhmä ei koskaan saanut käyttöönsä.

Kiuru-projekti toteutti kuitenkin rajapinnan käyttämiseen ja tietojen synkronointiin tarvittavat peruskomponentit. Sen sijaan testaus- ja viimeistelyvaihe jäi valitettavan lyhyeksi ja käytännössä kesken.

Tammikuussa 2003 projektiryhmän viimeistellessä projektikansioitaan yliopisto ilmoitti luopuvansa Timmi-järjestelmän käytöstä.

3.2 Varausmoduulin käyttöliittymä

Varausmoduuli käsittää Kiuru-projektin toteuttamat osiot, jotka liittyvät salivarausten tekemiseen, näyttämiseen ja käsittelyyn. Esimerkkeinä tästä ovat mm. varauspyynnön tekeminen, vahvistus ja poisto.

Kiuru-projekti sai toteutettua varausmoduuliin kaikki vaatimusmäärittelyssä määritellyt toiminnot lukuunottamatta varausprosessin virhetarkistuksia sekä mahdollisuutta selailla varauksia käyttäjien mukaan.

Varausprosessin virhetarkistukset ovat varausmoduulin suurin ongelma. Sovellus toimii, jos käyttäjä ei tee mitään väärin, vaan etenee lineaarisesti varausprosessin alusta loppuun saakka. Varausmoduulin saa kaatumaan antamalla väärän muotoisia syötteitä HTML-sivujen lomakkeisiin. Myös radiopainikeryhmän jättäminen kokonaan tyhjäksi kaataa sovelluksen. Sovellus kuitenkin toipuu virheestä painamalla selaimen takaisin-painiketta ja korjaamalla virheellisen syötteen.

Toteutetussa versiossa varauksien selailu onnistuu vain omien varausten osalta, jolloin käyttäjä voi nähdä omat varauspyyntönsä ja varauksensa sekä niiden käsittelytilanteen. Käyttäjäkohtainen varausten seuranta olisi hyödyllinen lähinnä tilastointi- ja laskutusoperaatioissa. Vastaavia ominaisuuksia ei tällä hetkellä kuitenkaan ole suunniteltu Korppi-järjestelmään, joten tämä ei liene suuri puute.

Varausmoduulissa olevat virheet estävät toistaiseksi sovelluksen laajamittaisen käytön yliopistossa ja senpä vuoksi sitä tuleekin testata ja jatkokehittää.

3.3 Ylläpitäjän ja päävahvistajan käyttöliittymä

Ylläpitäjän ja päävahvistajan käyttöliittymään saatiin toteutettua salien ja saliryhmien hallinta. Timmi-järjestelmän yhteyshenkilöiden hallintaa ei ehditty toteuttaa, mutta toiminto ei ole välttämätön järjestelmän käyttämiseen. Jos varaukseen halutaan merkitä uusia yhteyshenkilöitä, ne täytyy lisätä manuaalisesti Timmi-järjestelmään.

3.4 Resurssipyynnöt

Resurssipyynnöllä tarkoitetaan salin yksilöimisen sijaan haluttuun saliin liittyvien vaatimusten määrittelyä varauspyyntöä tehtäessä. Pyynnössä voidaan määritellä esimerkiksi tarpeet videoprojektorille ja tietokoneelle. Resurssipyyntöjä ei vielä pysty tekemään Kiuru-sovelluksella. Tietokannassa on tuki resurssipyynnöille, mutta käyttöliittymää niille ei ole toteutettu.

3.5 Sovelluksen monikielisyys

Sovelluksen tukea eri kieliversioille ei toteutettu Kiuru-projektin aikana. Osa-syynä tähän oli vastaavien toimintojen keskeneräisyys olemassaolevassa Korppi-järjestelmässä. Tämän aiheuttaa aikaisempien projektien käyttämät erilaiset käännöstekniikat, joita ollaan kehittämässä yhdenmukaiseksi. Toinen syy oli ajanpuute.

Projektiryhmä kuitenkin selvitti eri tekniikoita ja päätyi lopputulokseen, että lupaavin ratkaisu monikielisyyteen on Javan I18N. Java I18N on monipuolinen ja ennen kaikkea Sunin kehittämä ja Javaan sisäänrakennettu kielenkäännösteknologia, jota käyttämällä vältytään omatekoisilta ratkaisuilta.

Sovelluksen taustalle toimivista tietokannan tauluista osa on toteutettu siten, että niihin voidaan tallentaa tieto monikielisesti. Tästä tavasta ollaan kuitenkin osittain pyrkimässä eroon Korppi-järjestelmässä.

3.6 Yhteenveto sovelluksen ominaisuuksista

Ryhmä sai auttavasti kokoon lähes kaikki projektin alkupuolella sille asetetut vaatimukset.

Sovelluksessa havaitut virheet on koottu lukuun [9](#) ja jatkokehitysideoita on esitelty lukuun [10](#).

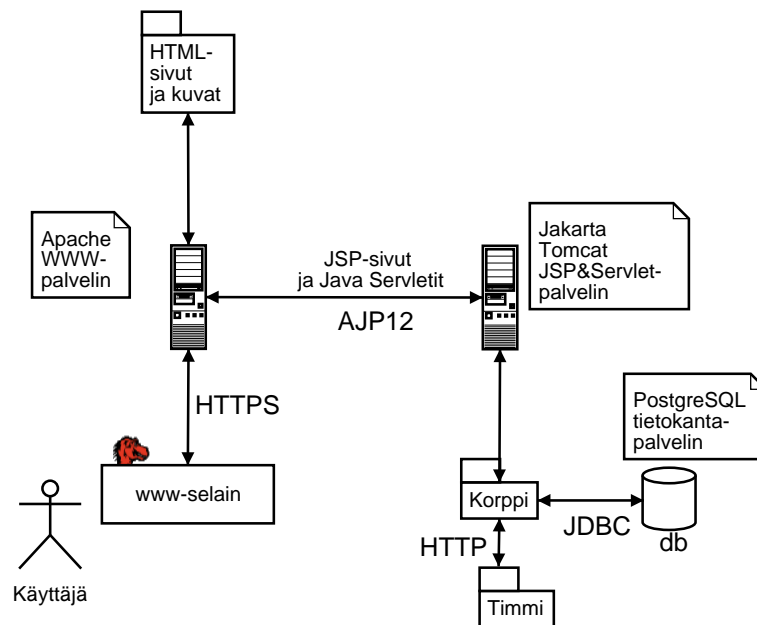
4 Palvelinohjelmistot, toteutustekniikat ja työkalut

Luvussa esitellään projektissa käytetyt työkalut ja ohjelmointitekniikat. Koska Korppi-järjestelmä on useiden aikaisempien projektien tulos, käytti Kiuru pitkälti samoja menetelmiä ja tekniikoita yhtenäisyyden säilyttämiseksi. Kehitysympäristön käyttöjärjestelmänä oli Redhat Linux 7.3 ja sovelluskehittimenä NetBeans IDE 3.4.

4.1 JSP-tekniikka

Korppi-järjestelmän rakennetta ja JSP-tekniikan toimintaa on havainnollistettu kuvassa 1. Kun WWW-palvelin vastaanottaa pyynnön selaimelta, se päättää, onko kyseessä staattinen HTML-sivu vai JSP-sivu. Apache palauttaa staattiset sivut suoraan takaisin selaimen.

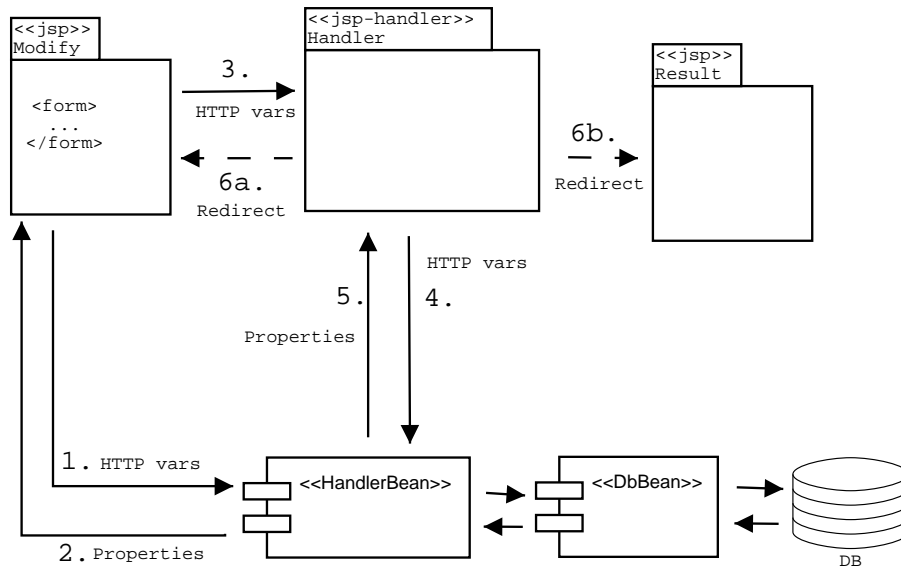
JSP-sivun pyyntö välittyy servlettimoottorille, joka Korppi-järjestelmässä on Tomcat. Se tutkii, onko kyselyä vastaava servletti olemassa. Olemassaoleva servletti suoritetaan ja se palauttaa selaimelle tuloksen HTML-muodossa. Jos servlettiä ei löydy, pyyntö ohjautuu JSP-moottorille, joka generoi servletin.



Kuva 1: Korppi-järjestelmän rakenne.

4.2 JSP-sivujen käsittelylogiikka

Kuvan 2 kaaviossa esitetään vaiheittain tietojen käsittelylogiikka johonkin tietokohteeseen tehtävien pyyntöjen ja muokkausten osalta.



Kuva 2: JSP-sivujen käsittely.

Oletetaan, että Modify-sivu on aluksi saanut HTTP-parametrina tiedon, mitä sivulla tehdään (esim. lisäys tai muokkaus) ja käsiteltävän tietokohteen tunnisteiden.

Pyynnön käsittelyvaiheet ovat seuraavat:

1. JSP-sivun HTTP-parametrit asettavat Handler-pavun vastaavat ominaisuudet (tieto siitä, mitä sivulla tehdään on myös ominaisuutena). HandlerBean hakee tarvittaessa tiedot tietokannasta Dbbean-pavun avulla.
2. Sivun tiedot generoidaan HandlerBean-pavun ominaisuuksien pohjalta.
3. Käyttäjän syöttämät Modify-lomakkeen tiedot lähetetään HTTP-muuttujina JSP-Handler-sivulle.
4. HTTP-parametrit asettavat HandlerBean-pavun vastaavat ominaisuudet. Jos syötteet ovat kelvollisia, HandlerBean päivittää tarvittaessa tietokannan tiedot.
5. HandlerBean palauttaa ominaisuuksissaan tiedon, onko päivitys onnistunut ja ovatko syötteet kelvollisia.

6. a) Jos tietojen päivitys ei onnistunut, `JSP-Handler` ohjaa kutsun takaisin `Modify`-sivulle ja ilmoittaa käyttäjälle virheistä.
- b) Muuten `JSP-Handler` ohjaa käyttäjän uudelle sivulle, jossa käyttäjä voi jatkaa.

Projektiryhmä noudatti seuraavia periaatteita JSP-sivuja ja Java-papuja toteuttaessaan:

- JSP-sivut eivät käsittele tietokantaa suoraan. Tarvittaessa tietokannan tietoja voidaan lukea Java-papujen palauttamista `RS2`-tietojoukkokomponentista.
- Java-pavut eivät ota kantaa tiedon esitystapaan. Poikkeuksena `HTMLBean`, jonka tarkoituksena on muotoilla `HTML`-muotoisia merkkijonoja annetusta tietojoukosta.
- `JSP-handler` -sivuilla ei yleensä tulosteta tietoja, vaan ohjataan käyttäjä eri sivuille riippuen sivun saamista `HTTP`-parametreista.

4.3 Java-pavut

Java-pavut (engl. *JavaBeans*) ovat Javalla toteutettuja luokkia. Java-koodi keskitettiin papuihin siten, että varsinaisten JSP-sivujen ylläpidettävyys säilyy. Lisäksi Kiuru-projekti toteutti yleiskäyttöisiä papuja, joita tulevat projektit ja jatkokehittäjät voivat käyttää varausten suunnitteluohjelman toteutuksessa.

Timmi-rajapinnan käsittely ja rajapinnasta saatavien `XML`-muotoisten vastausten käsittely sekä muut erikoistoiminnot hoidetaan omissa pavuissaan. Näin järjestelmän ylläpito ja muokkaaminen helpottuu.

4.4 Tietokanta

Aiempien Korppi-järjestelmään liittyvien sovellusprojektien tapaan Kiuru-projektissa ei käytetty `PostgreSQL`-tietokannan hallintajärjestelmän erikoispiirteitä, vaan noudatettiin `SQL92`-standardia. Ainoana poikkeuksena standardin käytöstä on tietokantaraportissa mainittu `EventSpace`- ja `Reservation`-taulujen synkronointi säännöillä. Tämä ratkaisu mahdollistaa järjestelmän siirrettävyyden alustalta toiselle.

Kiuru-projektin toteuttamat Kotka-tietokannan muutokset ja lisäykset suunniteltiin ER-kaavioiden avulla ja toteutettiin SQL-lauseilla. Muutokset olemassaolevaan tietokantaratkaisuun pyrittiin minimoimaan, mutta toisaalta pyrittiin ottamaan huomioon tulevien projektien tarpeet. Muutokset on kuvattu taulujen ja kenttien tasolla Kotka-tietokantaraportissa.

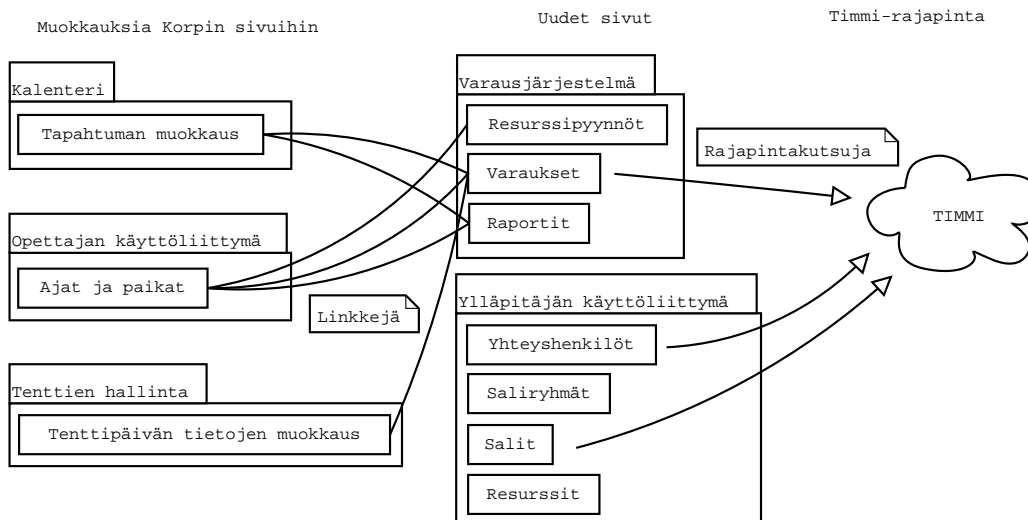
5 Sivujen käyttöliittymän yleisrakenne

Luvussa kuvataan yleisellä tasolla Kiuru-sovellukseen toteutettua käyttöliittymää sekä sen suhdetta Timmi-järjestelmään. Sivujen teknisiä toteutuksia on esitetty tämän dokumentin luvuissa 4 ja 6.

Kiuru-sovellukseen toteutettiin varausjärjestelmän käyttöliittymät salien varauspyyntöjen tekijöille ja vahvistajille sekä ylläpitäjän toiminnot salien ja saliryhmien hallintaan. Resurssipyyntöjen ja yhteyshenkilöiden käsittely sovittiin jätettäväksi jatkokehitykseen.

Käyttöliittymien ulkoasut pidettiin yhtenäisinä Korppi-järjestelmän ulkoasun kanssa. Sovelluksen käyttäjä ei siten tiedä käyttävänsä erillistä moduulia, vaan hänelle Kiuru-sovellus on yhtenäinen osa Korppi-järjestelmää.

Kuvassa 3 on kuvattu käyttöliittymäsivujen jakautuminen järjestelmän moduuleissa. Olemassaolevia sivuja on muokattu kalenterimodulin ja opettajan käyttöliittymän osalta. Tenttien hallinnan muokkaus jäi jatkokehitykseen. Kiuru-projektissa uutena toteutettu varausjärjestelmä sisältää sivut varausten listaukseen, käsittelyyn sekä raporttien generointiin. Varausten, yhteyshenkilöiden ja salien tietoja muokkaavat sivut käyttävät Timmi-rajapintakomponenttia.



Kuva 3: Sivustokartta.

6 JSP-sivut

Luvussa kuvataan kaikki Kiuru-projektin toteuttamat ja muokkaamat JSP-sivut. JSP-sivujen toimintaperiaate on luvussa 4.2 määritellyn periaatteen mukainen. Käsittelijäsivut on yleensä nimetty päätteellä `handler`. Toiminnot on kapseloitu mahdollisimman pitkälle Java-papuihin, jotka on kuvattu luvussa 7.

6.1 Kalenteri

Kalenterimoduuliin tehdyt muutokset liittyvät ensisijaisesti salivarausmahdollisuuden lisäämiseen tapahtuman määrittelyyn. Tapahtumia voi tarkastella henkilökohtaisen tai ryhmäkohtaisen näkymän lisäksi salikohtaisesti.

6.1.1 Salin viikkokalenteri `spaceweek.jsp`

Sivulla `spaceweek.jsp` käyttäjä voi tarkastella yksittäisen salin tapahtumia määrättyllä viikolla. Sivun vastaa ulkoasultaan ja toimintalogiikaltaan henkilön kalenterimoduulin `week.jsp`-sivua. Sivun näkymä on esitetty kuvassa 4.

The screenshot shows the Korppi web application interface. At the top, there is a header with the Korppi logo and user information: "Vesa Lappalainen, Tietotekniikan laitos [Luennoitsija] ma 13.1.2003". Below the header, there are navigation links: "Poistu", "Palaute", "asetukset", and "Omatuut".

On the left side, there is a sidebar menu with the following items: "Opetuskeski", "Opiskelusi", "Kalenteri", "Ilmoitustaulu", "Hankinnat", "Aiheet ja haku", "Salivaraukset", "Asetuksista voit vaihtaa kuukausikalenterin näkymän", and "Testikone!".

The main content area displays a calendar for "Tammikuu 2003". Below the calendar, there is a section titled "Viikkonäkymä, Belfast" and "Viikko 3". A date selector shows "Hyppää päivämäärään: 13.1.2003" with a "Käytä" button. Below this, there are links for "viikko 1", "viikko 2", "viikko 3", "viikko 4", and "viikko 5".

The central part of the page is a table showing events for each hour of the week. The table has columns for the days of the week (Ma 13.1, Ti 14.1, Ke 15.1, To 16.1, Pe 17.1, La 18.1, Su 19.1) and rows for the hours (0:00, 10:00, 12:00, 14:00, 16:00, 18:00). Each cell in the table contains a "[+]" symbol, indicating an event.

At the bottom of the page, there are controls for "Tarkkuus: 120 minuttia" and "Näytä väliltä: 08 - 20", along with a "Käytä" button and a "Tee raportti" button.

Kuva 4: Salin tapahtumatietojen viikkonäyttö.

Parametrin `spaceID` avulla määritellään sali, jonka tietoja tarkastellaan. Lisäksi sali käyttää muiden kalentersivujen tapaan parametreja `weekStartDate`, `clockMinuteAddition`, `firstHour` ja `lastHour` sivun ulkoasun määrittämiseen. *Tee raportti* -painikkeella käyttäjä voi luoda viikkonäkymästä PDF-muotoisen raportin.

6.1.2 Muutokset tapahtuman lisäyssivulle addAppointment.jsp

Käyttäjä voi määrittellä tapahtuman tiedot sivulla addAppointment. Kiuru-projekti lisäsi sivulle mahdollisuuden tehdä tapahtumasta salivaraus ja muokkasi käyttöliittymää salin valintaan. Sivun näkymä on esitetty kuvassa 5.

The screenshot shows a web application interface for editing an event. At the top, there is a header with the logo 'Korppi' and navigation links: 'Poistu', 'Palautte', 'Asetukset', and 'Opastus'. The user is identified as 'Vesa Lappalainen, Tietotekniikan laitos [Luennoitsija] ma 13.1.2003'. On the left, there is a calendar for January 2003, with the 13th selected. Below the calendar is a sidebar menu with links: 'Opetuksesi', 'Opiskelusi', 'Kalenteri', 'Päivä', 'Viikko', 'Kuukausi', 'Vuosi', 'Fyhmän luonti', 'Fyhmän tapahtumat', 'Lisää tapahtuma', 'Ohjausejat', 'Ilmoitustaulu', 'Henkilötiedot', 'Aiheet ja haku', 'Salivaraukset', 'Asetuksista voit vaihtaa kuukausikalenterin paikkaa', and 'Testikoneli'. The main content area is titled 'Muokkaa tapahtumaa'. It contains several form fields: 'Kuvaus *:' with the value 'Menonen'; 'Paikka:' with a dropdown menu showing 'A-rakennuksen studio' and a 'Hae Salia' button; 'Päiväys:' with date pickers for '14', '1', and '2003', and a checkbox for 'Koko päivä' with a link '[Useampi vuorokausi]'; 'Alkaa:' with time pickers for '8' and ':00'; 'Päättyy:' with time pickers for '10' and ':00'; 'Lisätiedot:' with a text area; 'Tärkeysluokka:' with a dropdown menu showing 'normaali'; and 'Suojaus:' with a dropdown menu showing 'julkinen'. Below these fields is a section titled 'Toistot' with the text 'Tällä tapahtumalla ei ole toistokertoja.' and a link 'Lisää viesti'. At the bottom, there are four buttons: 'Tallenna ja poistu', 'Tallenna ja lisää uusi', 'Poista', and 'Peruuta', along with a 'Tallenna ja tee salivaraus' button.

Kuva 5: Tapahtuman lisäys.

Parametri eventID määrittää tapahtuman tunnusteen. Paikka-tekstikenttä vaihdettiin alasvetovalikoksi, jossa näkyvät käyttäjän mielisalit ja tapahtumaa varten haetut salit. Näin järjestelmään ei lisätä vahingossa samaa salia moneen kertaan nimen kirjoitusasua vaihtamalla. Sama hakutoiminto on käytössä opettajan käyttöliittymässä sivulla modifyGroup.jsp (Katso luku 6.2.1).

6.2 Opettajan käyttöliittymä

Kalenterimoduulin tapaan myös opettajan käyttöliittymään tehdyt muutokset liittyvät salivarausmahdollisuuden lisäämiseen kurssin tapahtumien määrittelyyn.

6.2.1 Muutokset sivulle `modifyGroup.jsp`

`modifyGroup.jsp`-sivulla opettaja voi määrittellä kurssin tapahtumaryhmän ajat ja paikat. Kiuru-projekti lisäsi sivulle mahdollisuuden tehdä tapahtumista varauspyyntö ja muokkasi käyttöliittymää salin valinnan osalta. Sivun näkymä on esitetty kuvassa 6.

The screenshot shows the Korppi system interface. At the top, there is a header with the Korppi logo, user information (Vesa Lappalainen, Tietotekniikan laitos [Luennoitsija]), and the date (ma 13.1.2003). Below the header, there is a calendar for January 2003. The main content area is titled "OMA001 Omatöimikkurssi" and "Opetusryhmätyyppi: Luento". It contains a form for editing group details, including fields for "Yhteiset tiedot", "Kertakohtaiset tiedot", and "Ryhmakohtaiset tiedot". The form includes checkboxes for "Ryhmät", "Kerrat", "Automaattinen ilmoittautuminen", and "Ilmoittautuminen". There are also buttons for "Kasitttele kaikkia" and "Näytä valittujen tiedot".

Kuva 6: Opetusryhmien tapahtumien määrittäminen.

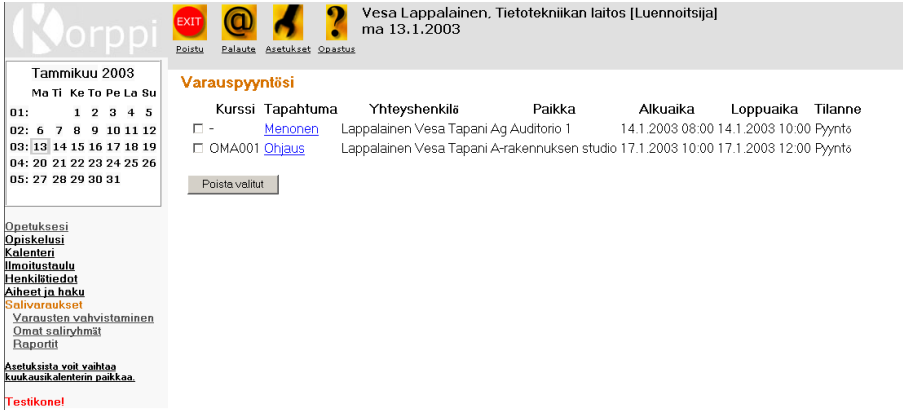
Sivulla käytetään uutta parametria `space` haetun salin tunnisteiden välittämiseen. `Paikka`-alavetovalikossa näkyvät kaikkien salien listan sijaan käyttäjän mielivalittavat ja tapahtumia varten haetut salit. Sama hakutoiminto on käytössä kalenterimoduulin sivulla `addAppointment.jsp`, joka kuvataan luvussa 6.1.2.

6.3 Varausjärjestelmä

Varausjärjestelmän kehittäminen Korppi-järjestelmään oli Kiuru-projektin merkittävin osuus. Kokonaisuuteen kuuluu sekä varauksien tietoja näyttäviä sivuja, että varausten hallintaan ja hyväksyntään liittyviä sivuja.

6.3.1 myReservations.jsp ja deleteReservations.jsp

Sivu `myReservations.jsp` on varausmoduulin aloitussivu. Sivulla on listattuna varauspyynnöt, joiden tekijänä tai yhteyshenkilönä käyttäjä on. Sivun näkyvä on esitetty kuvassa 7. `deleteReservations.jsp` on käsittelijä varauspyyntöjen poistamiseen.



The screenshot shows the Korppi reservation system interface. At the top, there is a navigation bar with the Korppi logo, an EXIT button, and icons for Poistu, Palauta, Asetukset, and Opastus. The user is identified as Vesa Lappalainen, Tietotekniikan laitos [Luennoitsija] on 13.1.2003. Below the navigation bar, there is a calendar for Tammikuu 2003. To the right of the calendar, the title 'Varauspyyntösi' is displayed. Below the title, there is a table of reservations with columns for Kurssi, Tapahtuma, Yhteyshenkilö, Paikka, Alkuaika, Loppuaika, and Tilanne. Two reservations are listed: one for 'Menonen' and one for 'Ohjaus'. A 'Poista valitut' button is located below the table. On the left side of the interface, there is a sidebar with various navigation links such as Opetuksesi, Opiskelusi, Kalenteri, Ilmoitustaulu, Henkilötiedot, Aiheet ja haku, and Salivaraukset.

Kurssi	Tapahtuma	Yhteyshenkilö	Paikka	Alkuaika	Loppuaika	Tilanne
<input type="checkbox"/>	Menonen	Lappalainen Vesa Tapani Ag	Auditorio 1	14.1.2003 08:00	14.1.2003 10:00	Pyyntö
<input type="checkbox"/>	OMA001 Ohjaus	Lappalainen Vesa Tapani	A-rakennuksen studio	17.1.2003 10:00	17.1.2003 12:00	Pyyntö

Kuva 7: Omat varaukset.

Käyttäjä näkee sivulta nopeasti varauspyyntöjen tilanteen. Jatkokehityksessä toistuvia tapahtumia on tarkoitus pystyä ryhmittelemään sekä liittämään yhteen. Tällöin tilanteesta saa yleiskuvan, vaikka tapahtumia olisi paljon.

6.3.2 searchFreeSpace.jsp ja newSpace.jsp

Sivua `searchFreeSpace.jsp` käytetään vapaan salin hakuun tapahtuman määrittelyn aikana. Sivulle on linkki kalenterin ja opettajan käyttöliittymän sivuilta (katso luvut 6.1 ja 6.2). Sivun näkymä on esitetty kuvassa 8. Haun tulokset on esitetty aliluvun 6.3.3 sivulla `searchResult.jsp`. Sivu `newSpace.jsp` on käsittelijä uuden salin lisäämiseen.

The screenshot shows the Korppi web application interface. At the top, there is a navigation bar with the Korppi logo and user information: "Vesa Lappalainen, Tietotekniikan laitos [Luennoitsija] ma 13.1.2003". Below the navigation bar, there are several icons: EXIT, @, a hand, and a question mark. The main content area is divided into several sections:

- Tammikuu 2003**: A calendar grid for January 2003, with the 13th highlighted.
- Etsi sali**: A search section with a form. It includes a "Ajat:" field with two date pickers (Vko 51/2002 and Vko 52/2002) and time ranges (Ma 08:00-10:00). Below this is a "Hakusana:" input field and a "Hae" button.
- Hae vain rakennuksista:** A section with checkboxes for "Agora" and "Mattianniemi".
- Koko:** A field with a dropdown menu set to "10".
- Tyyppi:** A section with checkboxes for "Luentosali", "Mikroluokka", and "Kokoustila".
- Lisää uusi sali**: A section with a "Salin nimi:" input field and a "Lisää" button.

On the left side, there is a sidebar with navigation links: "Opetuksesi", "Opiskelusi", "Kalenteri", "Ilmoitustaulu", "Henkilötiedot", "Aiheet ja haku", "Salivaraukset", "Asetuksista voit vaihtaa kuukausikalenterin paikkaa.", and "Testikone!".

Kuva 8: Vapaan salin haku.

Hakusanaksi voi antaa salin nimen alkuosan tai haluttaessa useampia sanoja pilkuilla tai kauttaviivoilla erotettuina. Sivulla voi myös lisätä nopeasti uuden salin tai paikan käyttöön, joka normaalin käyttäjän tapauksessa kirjautuu hänen lempisaliensa joukkoon. Varsinaisten salien lisäysoikeudet ovat vain ylläpitäjällä. Toiminto vastaa aiemman tapahtuman syöttö- ja muokkaussivujen tekstiruutua, johon käyttäjä saattoi vapaasti kirjoittaa salin nimen. Sali lisättiin automaattisesti järjestelmään, jos sitä ei ollut olemassa.

Sivun parametrit ovat samat kuin kutsuvalla sivulla. Valitun salin tunniste on parametrissa `space`.

6.3.3 searchResults.jsp

Sivulla näkyvät vapaan salin haun tulokset. Itse hakusivu on esitetty aliluvussa 6.3.2. Sivun näkymä on esitetty kuvassa 9.

The screenshot shows a web page with a header for 'Korppi' and user information for Vesa Lappalainen. The main content area is titled 'Hakutulokset' (Search Results) and displays a list of available rooms with radio buttons for selection. On the left, there is a calendar for January 2003 and a sidebar with navigation links.

Korppi EXIT @ ? Vesa Lappalainen, Tietotekniikan laitos [Luennoitsija]
ma 13.1.2003
Poistu Palaute Asetukset Opastus

Tammikuu 2003
Ma Ti Ke To Pe La Su
01: 1 2 3 4 5
02: 6 7 8 9 10 11 12
03: 13 14 15 16 17 18 19
04: 20 21 22 23 24 25 26
05: 27 28 29 30 31

Hakutulokset
Seuraaviin saleihin sopivat kaikki annetut ajat:
Varaus Sali Lisää mielisaleihin

- Ag B111.1 (Asia)
- Ag B112.1 (Africa)
- Ag B113.1 (Europe)
- Ag B211.1 (Sovjet)
- Ag B212.1 (Finland)
- Ag B213.1 (Lakes)
- AgA1
- Ag???
- Ag B112.2 (Latin)
- AgCOma
- AgCOma2

Valitse Lisää mielisaleihin

Opetuksesi
Opiskelusi
Kalenteri
Ilmoitustaulu
Henkilötiedot
Aiheet ja haku
Salivaraukset
Asetuksista voit vaihtaa
kuukausikalenterin paikkaa.
Testikone!

Kuva 9: Hakutulokset vapaista saleista.

Hakutuloksena saatuja saleja voi lisätä suoraan henkilökohtaiseen mielisali-ryhmään, jolloin sali näkyy jatkossa sivujen `addAppointment.jsp` ja `modifyGroup.jsp` alavetovalikoissa. Käyttäjän valitsema sali liitetään määriteltävään tapahtumaan *Valitse*-painikkeella.

Salin parametrit ovat samat kuin kutsuvalla sivulla. Salin hakuehto tuodaan parametrissa `searchText`.

6.3.4 makeReservation.jsp ja commitReservation.jsp

makeReservation.jsp-sivua käytetään opettajan käyttöliittymässä sivulla modifyGroup.jsp, kun käyttäjä tekee varauspyynnön määrittelemilleen tapahtumille. Määritellyistä tapahtumista muodostetaan osa sivun tiedoista. Sivun näkymä on esitetty kuvassa 10. commitReservation.jsp on käsittelijä varauksen tekemiseen.

The screenshot shows the Korppi web application interface. At the top, there is a navigation bar with the Korppi logo and user information: Miika Nurminen, Tietotekniikan laitos [Ylläpitäjä], ti 14.1.2003. Below the navigation bar, there are several icons: EXIT, @, a hand, and a question mark. A sidebar on the left contains a calendar for Tammikuu 2003 (January 2003) and a list of navigation links: Opetuksesi, Opiskelusi, Kalenteri, Ilmoitustaulu, Henkilötiedot, Aiheet ja haku, Selvitykset, Kurssinimikkeiden muokkaus, Ylläpito, Käännökset, Asetuksista voit vaihtaa kuukausikalenterin paikkaa, and Testikone! The main content area is titled 'Varauksen/varauspyynnön muokkaus' and contains a 'Tapahtumat' section with two entries: 'Ryhma1, AG C 222.1' with time 'Ajankohta: Vko 51 Ma 08:00:00.0 - 10:00:00.0' and 'Ryhma1, AG C 222.1' with time 'Ajankohta: Vko 52 Ma 08:00:00.0 - 10:00:00.0'. Below this is the 'Varauksen tiedot:' section with fields for 'Varauksen tila:' (dropdown menu), 'Varauspyynnön tekijä:' (Miika Nurminen), 'Tapahtuman kuvaus:' (KIURU-projektipalaveri), 'Yhteyshenkilö:' (radio button next to 'Lappalainen Vesa Tapani'), 'Laitos:' (dropdown menu), and 'Maksajaorganisaatio:' (dropdown menu). At the bottom of the form are two buttons: 'Tallenna varauspyyntö' and 'Poista'.

Kuva 10: Varauspyynnön teko.

Varauspyynnön tekijä on oletuksena Korppi-järjestelmän tämänhetkinen käyttäjä. Yhteyshenkilöt valitaan kurssin (tunti)opettajista tai tapahtumaryhmän jäsenistä.

Sivun parametrit ovat samat kuin kutsuvalla sivulla. Käsittelijän parametrit ovat seuraavat:

event	on varauksen tunniste,
requestState	on varauksen tilanne,
cPerson	on varaajan yhteyshenkilö,
organisation	on varaajan organisaatio,
payOrganisation	on maksajan organisaatio,
space	on salin tunniste ja
teeVarauspyynto	on sivun tallennustoiminto.

6.3.5 show- ja modifyReservationRequests.jsp

Sivu `showReservationRequests.jsp` listaa varauspyynnöt saleihin, joihin käyttäjällä on vahvistusoikeus. Varauspöyynnöt voi ryhmitellä käyttäjän määrittelemän kentän mukaan otsikkorivin linkeillä. Sivun näkymä on esitetty kuvassa 11. `modifyReservationRequests.jsp` on käsittelijä varausten vahvistukseen tai hylkäämiseen.

	Tapahtuma	Kurssi	Opettaja	Sali	Alkuaika	Loppuaika	Laitos
<input type="checkbox"/>	Vesa Lappalainen	-		AgCOma2	7.1.2003 10:00	7.1.2003 12:00	Tietotekniikan laitos
<input type="checkbox"/>	Vesa Lappalainen	-		A-rakennuksen studio	9.1.2003 08:00	9.1.2003 10:00	Tietotekniikan laitos
<input type="checkbox"/>	Vesa Lappalainen	-		Uimahalli	9.1.2003 08:00	9.1.2003 10:00	Tietotekniikan laitos
<input type="checkbox"/>	Vesa Lappalainen	-		Uimahalli	10.1.2003 10:00	10.1.2003 12:00	Tietotekniikan laitos
<input type="checkbox"/>	Vesa Lappalainen	-		Ag Auditorio 1	14.1.2003 08:00	14.1.2003 10:00	Tietotekniikan laitos
<input type="checkbox"/>	Ryhmä1	OMA001		A-rakennuksen studio	17.1.2003 10:00	17.1.2003 12:00	Tietotekniikan laitos
<input type="checkbox"/>	Ryhmä1	TIE305		Ag Auditorio 1	22.1.2003 10:00	22.1.2003 12:00	Informaatioteknologian tiedekunta

Kuva 11: Varauspöyynnöjen listaus.

`showReservationRequests.jsp`-sivun ainoa parametri on `order`, jolla määritetään varausten järjestys.

Käsittelijän parametrit ovat seuraavat:

<code>reservation</code>	määrittää vahvistettavat tai hylättävät varaukset,
<code>accept</code>	määrittää hyväksymistoiminnon ja
<code>reject</code>	määrittää hylkäämistoiminnon.

6.3.6 reportInit.jsp ja reportHandler.jsp

reportInit.jsp on raporttien luontisivu. Järjestelmän raportit rajoittuvat viikkolukujärjestysten luomiseen, joita tosin voi luoda samalla kertaa useammille saleille ja saliryhmille kerrallaan. Sivun näkymä on esitetty kuvassa 12.

Sivu reportHandler.jsp on raporttien käsittelijä. Sivulta pääsee myös spaceweek.jsp-sivulle (katso luku 6.1.1) tarkastelemaan salin viikkonäkymää.

The screenshot shows the 'Korppi' web application interface. At the top, there is a navigation bar with icons for 'EXIT', '@', a hand, and a question mark, along with the user information: 'Toni Hilpinen, Tietotekniikan laitos [Opiskelija] ma 13.1.2003'. Below the navigation bar, there are links for 'Poistu', 'Palautte', 'Asetukset', and 'Opastus'. The main content area is titled 'Raportin luonti' (Report Creation) and includes the instruction: 'Valitse sali tai saliryhmä, jonka tapahtumista raportti luodaan.' (Select a room or group from which the report will be generated). There is a search box labeled 'Hakuehto:' with the text 'ago' entered and a 'Hae' button. Below the search box, there is a section titled 'Hakutulokset' (Search Results) with a table showing a list of rooms and their types. The table has columns for 'Nimi' (Name) and 'Tyyppi' (Type). The results are as follows:

Nimi	Tyyppi
<input type="checkbox"/> Agora	Rakennus
<input type="checkbox"/> Agora	-
<input type="checkbox"/> Agora 2	-
<input type="checkbox"/> Agora 431.2	-
<input type="checkbox"/> Agora Beta	-
<input type="checkbox"/> Agora C132.1	-
<input type="checkbox"/> Agora C434.2	-
<input type="checkbox"/> Agora Delta	-
<input type="checkbox"/> Agora Takkahuone	-
<input type="checkbox"/> Agora Alfa-sali	-
<input type="checkbox"/> agoran gamma sali	-
<input type="checkbox"/> agoran saleissa	-

At the bottom of the search results, there is a 'Tee raportti valituista' (Generate report from selected) button. On the left side of the page, there is a calendar for 'Tammikuu 2003' (January 2003) and a sidebar with links for 'Opiskelusi', 'Kalenteri', 'Ilmoitustaulu', 'Henkilötiedot', 'Aiheet ja haku', 'Asetuksista voit vaihtaa kuukausikalenteriin paikkaa.', and 'Testikone!'.

Kuva 12: Raportin luonti.

Sivun parametrit ovat seuraavat:

code	on sivun hakuehto,
order	on salien ja saliryhmien listausjärjestys,
submitSearch	on toiminto uudelle haulle ja
continueSearch	on toiminto haun jatkamiselle.

Käsittelijän parametrit ovat seuraavat:

<code>weekStartDate</code>	määrittää aloitusviikon,
<code>clockMinuteAddition</code>	määrittää kellon asetukset,
<code>firstHour</code>	määrittää aloitustunnin,
<code>lastHour</code>	määrittää päättymistunnin,
<code>spaceId</code>	sisältää salit, jotka sisällytetään raporttiin (voi olla useita) ja
<code>spaceGroupId</code>	sisältää saliryhmät, joita raporttiin tulee (voi olla useita).

6.4 Ylläpitäjän käyttöliittymä

Ylläpitäjän käyttöliittymään tehtiin lisäyksiä salien ja saliryhmien hallinnan osalta. Tämä siksi, että salien varauspyyntö- ja vahvistusoikeudet määritellään saliryhmien avulla. Vain ylläpitäjä voi määrittellä uuden saliryhmän, johon voidaan tehdä varauksia.

Saliryhmien osalta sama käyttöliittymä on rajoitetusti käytössä myös tavallisilla käyttäjillä omien melisalien ryhmän muokkaukseen.

6.4.1 Muutokset sivulle admin.jsp

admin.jsp on ylläpitäjän aloitussivu. Sivun näkymä on esitetty kuvassa 13.



The screenshot shows the Korppi administration interface. At the top, there is a navigation bar with the Korppi logo and several icons: EXIT, @, a hand, and a question mark. Below the icons are links for Poistu, Palaute, Asetukset, and Opastus. The user information 'Miika Nurminen, Tietotekniikan laitos [Ylläpitäjä] ti 14.1.2003' is displayed on the right.

On the left side, there is a calendar for Tammikuu 2003 (January 2003) with a grid showing days of the week (Ma, Ti, Ke, To, Pe, La, Su) and dates from 01 to 31.

Below the calendar is a list of links for various administrative functions: Opetuksesi, Opiskelusi, Kalenteri, Ilmoitustaulu, Henkilötiedot, Aiheet ja haku, Salivaraukset, Kurssinimikkeiden muokkaus, Ylläpito, Kirjautuneet käyttäjät, Opinnäyteoikeudet, Kantapäivityksiä, JORE-tietoja, Postilistat, Salit, Saliryhmät, Yhteyshenkilöt, and Käännekieliset.

At the bottom left, there is a note: 'Asetuksista voit vaihtaa kuukausikalenterin paikkaa.' and a 'Testikone!' link.

On the right side, under the heading 'Ylläpitoliittymä', there is a list of links: Kirjautuneet käyttäjät, Opinnäyteoikeudet, Kantapäivityksiä, JORE-tietoja käyttäjille, rank-laskentaa, Postilistat, Salit, Saliryhmät, and Yhteyshenkilöt.

Kuva 13: Ylläpitäjän aloitussivu.

Kiuru-projekti lisäsi sivulle linkit *Salit* ja *Saliryhmät* uusille sivuille. Muita muutoksia ei tehty.

6.4.2 searchSpace.jsp ja searchSpaceHandler.jsp

Sivua `searchSpace.jsp` käytetään salien hakemiseen tietojen muuttamista tai poistamista varten. Sivun näkymä on esitetty kuvassa 14. `searchSpaceHandler.jsp` on sivun käsittelijä.



Kuva 14: Salin haku.

Hakusanaksi voi antaa salin nimen alkuosan tai haluttaessa useampia sanoja pilkuilla tai kauttaviivoilla erotettuna. Ylläpitäjä voi tarvittaessa poistaa useamman salin kerralla (poistamisesta katso luku 6.4.3).

Sivun parametrit ovat seuraavat:

<code>code</code>	on hakuehto,
<code>order</code>	on salien järjestys,
<code>submitSearch</code>	on toiminto uudelle haulle ja
<code>continueSearch</code>	on toiminto haun jatkamiselle.

Käsittelijän parametrit ovat seuraavat:

<code>submitAdd</code>	on salin lisäystoiminto,
<code>submitDelete</code>	on salin poistotoiminto ja
<code>selectedSpaces</code>	sisältää poistettavaksi valitut salit.

6.4.3 modifySpace.jsp ja spaceHandler.jsp

modifySpace.jsp-sivu on tarkoitettu salin tietojen muokkaukseen. Lisäksi salin voi liittää yhteen tai useampaan saliryhmään. Sivun näkymä on esitetty kuvassa 15.

Korppi

EXIT @ ? Toni Hilpinen, Tietotekniikan laitos [Opiskelija]
ma 13.1.2003

Poistu Palaute Asetukset Opastus

Tammikuu 2003
Ma Ti Ke To Pe La Su
01: 1 2 3 4 5
02: 6 7 8 9 10 11 12
03: 13 14 15 16 17 18 19
04: 20 21 22 23 24 25 26
05: 27 28 29 30 31

Opiskelusi
Kalenteri
Ilmoitustaulu
Henkilötiedot
Aiheet ja haku

Asetuksista voit vaihtaa
kuukausikalenterin paikkaa.

Testikone!

Muokkaa salin tietoja

Salin nimi:

Vastuuorganisaatio:

Salin tyyppi:
 Ei-tyyppiä
 Opetustila
 Mikroluokka
 Kokoustila
 Tyshuone
 Yleistila
 Oheisvarattava
 Muu tila

Salin koko:

Saliryhmat, joihin sali kuuluu: (ei saliryhmiä)

[Salit](#) | [Saliryhmat](#) | [Raportit](#)

Kuva 15: Salin tietojen muokkaus.

Viite-ehyden säilyttämiseksi salia poistettaessa merkitään poistetuksi myös salin liitokset saliryhmiin, resursseihin ja tapahtumiin.

Sivun parametrit ovat seuraavat:

spaceId on käsiteltävän salin tunniste ja
submitModify on salin muokkaustoiminto.

Käsittelijän parametrit ovat seuraavat:

spaceId	on käsiteltävän salin tunniste,
submitModify	on salin muokkaustoiminto,
name	on salin nimi,
organisationId	on salista vastaavan organisaation tunniste,
type	on salityypin tunniste,
size	on salin paikkamäärä,
submitPost	on tallennustoiminto,
submitDeleteRecord	on poistotoiminto,
submitAddSpaceGroup	on saliryhmän lisäys,
submitAddSpaceGroup-Confirm	on saliryhmän lisäyksen vahvistus ja
selectedSpaceGroups	on lisättävien saliryhmien tunniste.

6.4.4 addSpaceGroupToSpace.jsp

Sivu `addSpaceGroupToSpace.jsp` on käyttöliittymä salin liittämiseen yhteen tai useampaan olemassaolevaan saliryhmään. Sivun näkymä on esitetty kuvassa 16.

The screenshot shows a web application interface. At the top, there is a header with the logo 'Korppi' and navigation links: EXIT, @, a hand icon, and a question mark icon. The user is identified as 'Miika Nurminen, Tietotekniikan laitos [Ylläpitäjä]' with the date 'ti 14.1.2003'. Below the header, there is a calendar for 'Tammikuu 2003' and a list of links: Opetuksesi, Opiskelusi, Kalenteri, Ilmoitustaulu, Henkilötiedot, Aiheet ja haku, Salvaraukset, Kurssinimikkeiden muokkaus, Ylläpito, Käännökset, Asetuksista voit vaihtaa kuukausikalenterin paikkaa, and Testikone!. The main content area is titled 'Salin MaD 202 tiedot päivitetty!' and 'Salin liittäminen saliryhmään'. It states 'Olet kasittelemassa salia MaD 202.' and shows a table of groups to be added to the room. The table has columns 'Nimi' and 'Tyyppi' and a checkbox for each group.

Lisää sali valittuihin	
Nimi	Tyyppi
<input type="checkbox"/> Agora	Rakennus
<input type="checkbox"/> Ikioma	Henkilökohtainen saliryhma
<input type="checkbox"/> Mielisalit_jusmaki	Henkilökohtainen mielisaliryhma
<input type="checkbox"/> Mielisalit_vesal	Henkilökohtainen mielisaliryhma
<input type="checkbox"/> Santasen juhlasalit	Henkilökohtainen mielisaliryhma
<input type="checkbox"/> Santasen juhlasalit	Henkilökohtainen mielisaliryhma
<input type="checkbox"/> Seminaarinmaki	Alue
<input type="checkbox"/> Vesa salit	Henkilökohtainen saliryhma
<input type="checkbox"/> Ylistonmaki	Oheisarattavan toimituspaikat
<input type="checkbox"/> testi	Henkilökohtainen saliryhma

Lisää sali valittuihin

Kuva 16: Saliryhmän liittäminen saliin.

Sivua kutsutaan sivulta `modifySpace.jsp` (luku 6.4.3) ja se käyttää samaa käsittelijää. Sivun ainoa parametri on saliryhmien järjestyksen määrittävä `order`.

6.4.5 searchSpaceGroup.jsp

Sivua `searchSpaceGroup.jsp` käytetään saliryhmien hakemiseen tietojen muuttamista tai poistamista varten. Ylläpitäjälle näkyvät kaikki saliryhmät, kun muille näkyvät ainoastaan henkilökohtaiset hakuryhmät, mielisaliryhmä ja muut saliryhmät, joihin henkilöllä on vahvistus- tai muokkausoikeus. Sivun näkymä on esitetty kuvassa 17.



The screenshot shows a web application interface for managing saliryhmät. At the top, there is a header with the Korppi logo, navigation icons (EXIT, @, a hand, a question mark), and the user information: Vesa Lappalainen, Tietotekniikan laitos [Luennoitsija] ma 13.1.2003. Below the header, there is a calendar for January 2003. On the left, there is a vertical menu with links to various services like Opetuksesi, Opiskelusi, Kalenteri, Ilmoitustaulu, Henkilötiedot, Aiheet ja haku, Salivaraukset, Varausten vahvistaminen, Omat saliryhmät, Raportit, Asetuksista voit vaihtaa kuukausikalenterin paikkaa, and Testikonel. The main content area is titled 'Saliryhmät' and includes a 'Tallennettu!' message. Below this, there is a text prompt: 'Valitse saliryhma, jonka tietoja haluat muokata.' A table lists the saliryhmät with columns for 'Nimi' and 'Tyyppi'. The table contains four entries: 'Agora' (Rakennus), 'Mielisalit_vesal' (Henkilökohtainen mielisaliryhmä), 'Santasen juhlasalit' (Henkilökohtainen mielisaliryhmä), and 'Vesa salit' (Henkilökohtainen saliryhmä). There is a 'Lisää uusi' button below the table. At the bottom of the main content area, there are links for 'Salit | Saliryhmät | Raportit'.

Nimi	Tyyppi
Agora	Rakennus
Mielisalit_vesal	Henkilökohtainen mielisaliryhmä
Santasen juhlasalit	Henkilökohtainen mielisaliryhmä
Vesa salit	Henkilökohtainen saliryhmä

Kuva 17: Saliryhmän haku.

Jatkokehityksessä sivuun on lisättävä hakutoiminto. Sivun ainoa parametri on saliryhmien järjestyksen määrittävä `order`.

6.4.6 modifySpaceGroup.jsp ja spaceGroupHandler.jsp

Sivulla `modifySpaceGroup.jsp` voi muokata saliryhmän tietoja. Ylläpitäjällä on oikeus kaikkien tietojen muokkaamiseen, kun muiden oikeudet määräytyvät vahvistus- ja muokkaus-oikeuden mukaan. Sivun näkymä on esitetty kuvassa 18. `spaceGroupHandler.jsp` on sivun käsittelijä.



The screenshot shows the Korppi web application interface. At the top, there is a navigation bar with the Korppi logo and user information: Toni Hilpinen, Tietotekniikan laitos [Opiskelija], ma 13.1.2003. Below the navigation bar, there is a calendar for Tammikuu 2003. The main content area is titled 'Muokkaa salin tietoja' (Edit room information). It contains several form fields: 'Salin nimi:' (Room name) with the value 'BaD 120', 'Vastuuorganisaatio:' (Responsible organization) with a dropdown menu showing 'Terveystieteiden laitos', 'Salin tyyppi:' (Room type) with radio buttons for 'Ei tyyppiä', 'Opetustila', 'Mikroluokka', 'Kokoustila', 'Tyshuone', 'Yleistila', 'Oheisvarattava', and 'Muu tila', and 'Salin koko:' (Room size) with the value '15'. There are also buttons for 'Tallenna' (Save) and 'Poista sali' (Delete room). At the bottom, there are links for 'Salit', 'Saliryhmät', and 'Raportit'.

Kuva 18: Saliryhmän tietojen muokkaus.

Jos henkilöllä on muokkaus-oikeus, hän saa lisätä tai poistaa henkilöitä ja sa-
leja saliryhmästä. Jos henkilöllä on lisäksi päävahvistajan oikeudet, hän saa muo-
kata muiden saliryhmään liittyvien henkilöiden vahvistusoikeuksia. Muut saavat
ainoastaa luoda uusia hakuryhmiä tai mielisaliryhmiä. Vain ylläpitäjä saa luoda
saliryhmän, johon kuuluvilla henkilöillä on varausoikeudet.

Sivun parametrit ovat seuraavat:

<code>spaceGroupid</code>	on saliryhmän tunniste,
<code>submitModify</code>	on salin muokkaustoiminto ja
<code>submitAdd</code>	on lisäystoiminto.

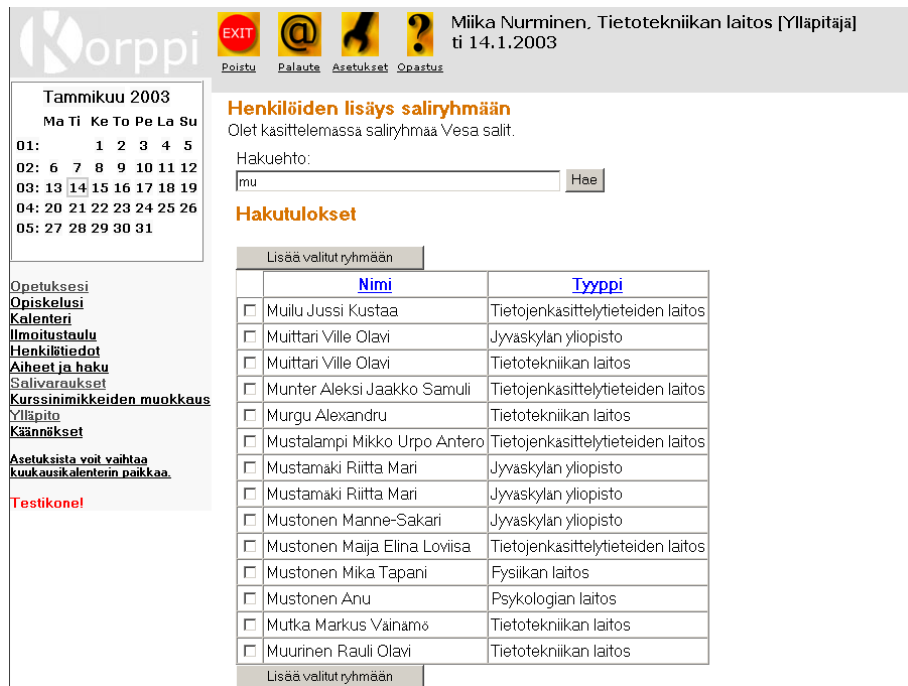
Käsittelijän parametrit ovat seuraavat:

<code>spaceGroupid</code>	on saliryhmän tunniste,
<code>submitModify</code>	on salin muokkaustoiminto,
<code>submitAdd</code>	on lisäystoiminto,
<code>name</code>	on saliryhmän nimi,
<code>type</code>	on saliryhmätyypin tunniste,
<code>accessRightLevelId</code>	on varauspyyntöjen pienin käyttöoikeustaso,
<code>description</code>	on vapaamuotoinen kuvaus,
<code>submitPost</code>	on tallennustoiminto,
<code>submitDelete</code>	on poistotoiminto,
<code>submitAddSpace</code>	on salin lisäys saliryhmään,
<code>submitDeleteSpace</code>	on salin poisto saliryhmästä,
<code>submitAddSpaceConfirm</code>	on salin lisäyksen vahvistus,
<code>selectedSpaces</code>	sisältää valittujen salien tunnisteet,
<code>submitAddPersons</code>	on henkilön lisäys saliryhmään,
<code>submitDeletePersons</code>	on henkilön poisto saliryhmästä,
<code>submitAddPersonConfirm</code>	on henkilön lisäyksen vahvistus ja
<code>selectedPersons</code>	sisältää valittujen henkilöiden tunnisteet.

Käsittelijä käyttää myös numeroituja parametreja `accessLevelN` ja `modifyRightN`, missä N on henkilön tunnistenumero ja parametrien arvot ovat kyseisen henkilön vahvistusoikeus ja muokkausoikeus.

6.4.7 addPersonToSpaceGroup.jsp

Sivu `addPersonToSpaceGroup.jsp` on käyttöliittymä henkilöiden lisäämiseen saliryhmään jossakin roolissa. Hakusanaksi voi antaa henkilön sukunimen alkuosan tai haluttaessa useampia sanoja pilkulla tai kauttaviivalla erotettuina. Sivun näkymä on esitetty kuvassa 19.



The screenshot shows the Korppi web application interface. At the top, there is a navigation bar with icons for 'EXIT', '@', a hand, and a question mark, along with the user's name 'Miika Nurminen, Tietotekniikan laitos [Ylläpitäjä]' and the date 'ti 14.1.2003'. Below the navigation bar, there are links for 'Poistu', 'Palautte', 'Asetukset', and 'Opastus'. The main content area is titled 'Henkilöiden lisäys saliryhmään' and includes a search form with a 'Hae' button. Below the search form, there is a table of search results with checkboxes for selection. The table has columns for 'Nimi' and 'Tyyppi'. The search results are as follows:

Lisää valitut ryhmään		
	Nimi	Tyyppi
<input type="checkbox"/>	Muilu Jussi Kustaa	Tietojenkäsittelytieteiden laitos
<input type="checkbox"/>	Muittari Ville Olavi	Jyväskylän yliopisto
<input type="checkbox"/>	Muittari Ville Olavi	Tietotekniikan laitos
<input type="checkbox"/>	Munter Aleksi Jaakko Samuli	Tietojenkäsittelytieteiden laitos
<input type="checkbox"/>	Murgu Alexandru	Tietotekniikan laitos
<input type="checkbox"/>	Mustalampi Mikko Urpo Antero	Tietojenkäsittelytieteiden laitos
<input type="checkbox"/>	Mustamaki Riitta Mari	Jyväskylän yliopisto
<input type="checkbox"/>	Mustamaki Riitta Mari	Jyväskylän yliopisto
<input type="checkbox"/>	Mustonen Manne-Sakari	Jyväskylän yliopisto
<input type="checkbox"/>	Mustonen Maija Elina Loviisa	Tietojenkäsittelytieteiden laitos
<input type="checkbox"/>	Mustonen Mika Tapani	Fysiikan laitos
<input type="checkbox"/>	Mustonen Anu	Psykologian laitos
<input type="checkbox"/>	Mutka Markus Vainams	Tietotekniikan laitos
<input type="checkbox"/>	Muurinen Rauli Olavi	Tietotekniikan laitos

At the bottom of the table, there is a button 'Lisää valitut ryhmään'.

Kuva 19: Henkilön lisäys saliryhmään.

Sivua kutsutaan sivulta `modifySpaceGroup.jsp` (luku 6.4.6) ja se käyttää samaa käsittelijää.

Sivun parametrit ovat seuraavat:

<code>personName</code>	on hakuehto,
<code>order</code>	on järjestyksen määrittely
<code>submitSearch</code>	on toiminto uudelle haulle ja
<code>continueSearch</code>	toiminto haun jatkamiselle.

6.4.8 addSpaceToSpaceGroup.jsp

Sivu `addSpaceToSpaceGroup.jsp` on käyttöliittymä salien lisäämiseen saliryhmään. Hakusanaksi voi antaa salin koodin alkuosan tai haluttaessa useampia sanoja pilkulla tai kauttaviivalla erotettuina. Sivun näkymä on esitetty kuvassa 20.

The screenshot shows the Korppi web application interface. At the top, there is a header with the Korppi logo and navigation links: Poistu, Palaute, Asetukset, and Opastus. The user is identified as Miika Nurminen, Tietotekniikan laitos [Ylläpitäjä], logged in on 14.1.2003.

On the left side, there is a calendar for Tammikuu 2003 (January 2003) with days of the week (Ma, Ti, Ke, To, Pe, La, Su) and dates from 01 to 31.

The main content area is titled "Salien lisäys saliryhmään" (Add course to group). Below the title, it says "Olet käsittelemässä saliryhmää Vesa salit." (You are processing the course group Vesa salit). There is a search bar with the input "agc" and a "Hae" (Search) button.

Below the search bar, there is a section titled "Hakutulokset" (Search results). It contains a table with the following data:

Lisää valitut ryhmään		
	Nimi	Tyyppi
<input type="checkbox"/>	AgC 222.1	-
<input type="checkbox"/>	AgC 413.1	-
<input type="checkbox"/>	AgC 434.2	-
<input type="checkbox"/>	AgC 515.3	-
<input type="checkbox"/>	AgC132.1	-
<input type="checkbox"/>	AgC133.1	-
<input type="checkbox"/>	AgC223.1	-
<input type="checkbox"/>	AgC231.1	-
<input type="checkbox"/>	AgC321.1	-
<input type="checkbox"/>	AgC434.2	-
<input type="checkbox"/>	AgC00ma	Opetustila
<input type="checkbox"/>	Agc223	-

At the bottom of the table, there is a button labeled "Lisää valitut ryhmään" (Add selected groups).

Kuva 20: Salin lisäys saliryhmään.

Sivua kutsutaan sivulta `modifySpaceGroup.jsp` (luku 6.4.6) ja se käyttää samaa käsittelijää.

Sivun parametrit ovat seuraavat:

<code>code</code>	on hakuehto,
<code>order</code>	on järjestyksen määrittäminen,
<code>submitSearch</code>	on toiminto uudelle haulle ja
<code>continueSearch</code>	on toiminto haun jatkamiselle.

7 Java-pavut

Kiuru-projektissa yleiskäyttöiset aliohjelmat on ohjelmoitu Java-papuna. Aiemmat projektit ovat toteuttaneet töidensä yhteydessä niistä osan, mutta Kiuru-projekti joutui uudistamaan osaa niistä. Varsinaisten opiskelijaprojektien lisäksi papuja on toteutettu myös Korppi-järjestelmän jatkokehityksessä, osittain päällekkäin Kiuru-projektin tuotosten kanssa. Jatkokehittäjien toteuttamia muutoksia ei ole dokumentoitu projektin dokumentteihin.

Olemassaolevien papujen lisäksi Kiuru-projekti suunnitteli ja toteutti runsaasti uusia papuja. Osaksi tämä johtuu JSP-sivujen uudesta käsittelylogiikasta (katso luku 4.2) ja osaksi siitä, että projekti asetti yhdeksi osatavoitteeksi koko järjestelmää helpottavien uusien yleiskäyttöisten papujen luomisen. Esimerkiksi `HtmlBean`- ja `SimpleDB`-pavut (luvut 7.3.1 ja 7.3.4) yksinkertaistavat JSP-sivuja huomattavasti.

Luvussa kuvataan Kiuru-projektin toteuttamat lisäykset ja muutokset olemassaoleviin papuihin. Kaikkien uusien ja muokattujen papujen lähdekoodi on tulostettuna muiden projektiin liittyvien lähdekoodien yhteydessä. Tarkemmat kuvaukset metodeista löytyvät papujen JavaDoc-dokumentaatiosta.

7.1 kotkabeans-pakettiin tehdyt muutokset

`kotkabeans`-paketti pyrittiin pitämään mahdollisimman muuttumattomana, koska paketin pavut ovat koko järjestelmän käytössä. Olemassaolevia metodeja ei muutettu lukuunottamatta `HtmlMenu`-luokan `createTree`-metodia. Muihin luokkiin tehtiin pieniä lisäyksiä.

7.1.1 cSQL.java

cSQL-pavussa on järjestelmän eri moduulien käyttämiä SQL-lauseita. Papu otettiin käyttöön Korppi-järjestelmän jatkokehityksessä.

Kiuru-projekti lisäsi papuun seuraavat metodit:

<code>getOrganisationById</code>	palauttaa tunnisteella määritellyn organisaation nimen käyttäjän kielellä.
<code>getAccessRightLevels</code>	palauttaa järjestelmän käyttöoikeustasot käyttäjän kielellä.

7.1.2 DB.java

DB-papu on ollut järjestelmän rajapinta tietokannan käyttöön Kotka-projektista alkaen.

Papuun lisättiin uusi metodi `prepareStatement`. Metodia käytetään valmisteltujen SQL-lauseiden luontiin. Valmisteltu SQL-lause parantaa suorituskykyä erityisesti tilanteessa, jos joudutaan ajamaan monta kertaa lause, joka on rakenteeltaan sama, mutta lauseen parametrit muuttuvat välillä. Esimerkiksi kyselysarja

```
select * from person
  where personid = 1 and homeorganisationid = 10;

select * from person
  where personid = 2 and homeorganisationid = 10;
```

voidaan suorittaa tehokkaammin määrittelemällä valmisteltu SQL-lause seuraavasti:

```
select * from person
  where personid = ? and homeorganisationid = ?;
```

Ennen kyselyn ajamista määritellään ainoastaan kysymysmerkillä määritellyt parametrit.

7.1.3 Encoder.java

Papua käytetään koodaamaan käyttäjältä kysyttäviä merkkijonoja. Esimerkiksi SQL-lauseisiin menevissä tiedoissa ei saa olla lauseen rakennetta sekoittavia tai muuten ”vaarallisia” merkkejä. Papu otettiin käyttöön Korppi-järjestelmän jatkokehityksessä.

Papuun lisättiin metodi `LatexEncode`, joka tunnistaa syötemerkkijonosta \LaTeX -ladontaohjelman käyttämät ohjausmerkit ja muuttaa ne tavallisina merkkeinä tulostuvaan muotoon.

7.1.4 HTMLMenu.java

HTMLMenu-papu vastaa Korppi-järjestelmän käyttöliittymän vasemmassa reunassa näkyvästä navigointipuusta. Papu otettiin käyttöön Korppi-järjestelmän jatkokehityksessä. Selvitys papun toiminnasta löytyy HTML-dokumenttina osoitteesta <http://jamshedpur.it.jyu.fi/doku.html#MENU>.

Papuun lisättiin vakioita Kiuru-sovelluksen fyysisen polun määräämiseksi. Lisäksi metodiin `createTree` lisättiin `Salivaraukset`-osio navigointipalkkiin ja linkit uusille ylläpitäjän sivuille.

Muokatun papun toiminta ei ole täysin toiveiden mukainen. Johtuen JSP-sivujen uudesta käsittelylogiikasta käsiteltävän sivun URL voi muuttua toimintojen aikana. Esimerkiksi `ModifySpace.jsp`:stä siirrytään salin tietoja tallennettaessa sivulle `SpaceHandler.jsp` (katso luku 6.4.3), jolloin navigointipalkki ”romahtaa kasaan”. Tällöin valittuna olevaa sivua tai sovelluksen osaa ei voi havaita navigointipalkista. Toivottavampaa olisi, että navigointipalkki säilyttäisi edellisen tilansa. Kiuru-projektin puitteissa tätä ominaisuutta ei kuitenkaan korjattu.

7.1.5 User.java

User-papu vastaa käyttäjän tietojen ylläpidosta. Papu on ollut käytössä Kotka-projektista lähtien.

Kiuru-projekti lisäsi papuun metodin `hasAdminRight`. Metodi palauttaa tiedon siitä, onko käyttäjällä ylläpitäjän oikeudet.

7.1.6 Tools.java

Tools-papu sisältää yleiskäyttöisiä metodeja. Se on ollut käytössä Kotka-projektista lähtien.

Papuun lisättiin seuraavat metodit:

<code>getRequest-</code>	palauttaa merkkijonon, jossa on ilmaistu
<code>ParametersAs-</code>	HTTP-kyselyn parametrit HTML-kielen
<code>Hidden</code>	hidden-elementteinä.
<code>synchronizeStreams</code>	synkronoi Javan syöte- ja tulostusvirran niin, että syötevirta tulostetaan rivi kerrallaan tulostusvirtaan, kunnes syötevirrassa ei ole mitään.

7.2 kolibribeans-pakettiin tehdyt muutokset

kolibribeans-paketin pavut liittyvät lähinnä kalenterimoduulin käyttöön. Uusia metodeja jouduttiin lisäämään salin viikkonäkymää varten. Lisäksi raporttien luonnissa käytettiin pohjana kalenterimoduulin koodia.

7.2.1 EventBean.java

EventBean on Kolibri-projektin luoma yksittäisen tapahtuman kapseloiva papu.

Kiuru-projekti lisäsi papuun seuraavat varausjärjestelmää tukevat metodit:

<code>getContactPerson</code>	palauttaa varaukseen liittyvän varaajan yhteyshenkilön. Jos tapahtuma ei ole varaus, metodi palauttaa tapahtuman omistavan tapahtumaryhmän nimen.
<code>getOrganisation</code>	palauttaa varaukseen liittyvän varaajaorganisaation. Jos tapahtuma ei ole varaus, metodi yrittää etsiä organisaation kurssista, johon tapahtuma liittyy. Jos kyseessä ei ole opetustapahtuma, metodi palauttaa tyhjän merkkijonon.

7.2.2 EventBeanContainer.java

EventBeanContainer-papu on Kolibri-projektin luoma tietorakenne tapahtumajoukkoa varten. Kiuru-projekti korjasi pavun muodostimessa olevan virheen, joka liittyi henkilön tunnistenumeron määrittämiseen.

7.2.3 QueryContainer.java

QueryContainer-pavussa ovat Kolibri-projektin tietokantakyselyt. Papu vastaa kotkabeans-paketissa olevaa cSQL-papua (katso luku 7.1.1).

Papuun lisättiin metodi `getSpacesEvents`, joka palauttaa kaikki tiettyyn saliin liittyvät tapahtumat. Metodia käytetään raporttien luonnissa (katso luku 6.3.6).

7.2.4 Week.java

Papu kapseloi kalenterin viikkonäytön. Se otettiin käyttöön Kolibri-projektissa.

Papuun lisättiin metodi `printLatexWeekCalendar`. Se palauttaa tulosvirtaan viikkonäkymän \LaTeX -formaatissa. Metodin toteutus vastaa samassa pavussa olevaa `printWeekCalendarBookTime`-metodia.

7.3 Kiurubeans-paketti

Kiuru-projektin toteuttamat uudet pavut jakautuvat karkeasti seuraaviin luokkiin:

- Yleiskäyttöiset pavut, kuten `HtmlBean` ja `KiuruString`.
- Kotka-tietokannan kohteiden käsittelijät, kuten `KiuruString` ja `KiuruReservation`.
- Hakutoimintojen käsittelijät, kuten `SpaceGroupSearch`.
- Kotka-tietokannan ja Timmi-järjestelmän synkronointiin liittyvät pavut, kuten `TimmiParser2`.
- Timmi-järjestelmän tietokohteiden käsittelijät, kuten `TimmiCustomer` ja `TimmiSpace`.

Papujen suunnittelussa ja toteutuksessa pyrittiin hyödyntämään Javan olio-ohjelmointia tukevia ominaisuuksia. Luokat `KiuruHandler` (luku 7.3.9), `SearchHandler` (luku 7.3.13), `StatementItem` (luku 7.3.6) ja `TimmiObject` (luku 7.3.20) ovat yliluokkia muille järjestelmän pavuille.

7.3.1 HtmlBean.java

Papu sisältää HTML-sivujen generointia helpottavia yleiskäyttöisiä metodeja. Se sisältää attribuuttien `get-` ja `set-` metodien lisäksi seuraavat merkijonoja palauttavat metodit:

<code>bottomNavigator</code>	palauttaa varausjärjestelmän sivujen alaosassa olevan linkkilistan.
<code>createSearchForm</code>	luo HTML-lomakkeen, johon käyttäjä voi syöttää hakuehdon. Käytetään esimerkiksi henkilön hakemisessa nimen mukaan.
<code>getHidden</code>	palauttaa piilotetun HTML-lomakkeen elementin.
<code>getSubmit</code>	palauttaa HTML-lomakkeen painikkeen.
<code>makeCheckBox</code>	luo HTML-lomakkeen valintaruutulistian annettujen tietueiden perusteella.
<code>makeRadioGroup</code>	luo HTML-lomakkeen poissulkevan radiopainikelistian annettujen tietueiden perusteella.
<code>makeSelectList</code>	luo HTML-lomakkeen valintalistian annettujen tietueiden perusteella.
<code>translate</code>	kääntää annetun merkkijonon <code>User</code> -pavun <code>T-</code> metodilla ja palauttaa tuloksen erityisfontilla. Käytetään mm. ilmoitusten välittämisessä käyttäjälle.

7.3.2 StringPair.java

`StringPair`-papu kapseloi merkkijonomuotoisen avaimesta ja arvosta koostuvan parin. Se ei sisällä attribuuttien `get-` ja `set-` metodien lisäksi muita metodeja.

7.3.3 KiuruString.java

KiuruString-papu sisältää merkkijonojen käyttöä helpottavia yleiskäyttöisiä metodeja. Se sisältää attribuuttien get- ja set-metodien lisäksi seuraavat metodit:

<code>arrayToCSV</code>	muuntaa taulukon erotinmerkillä erotelluksi merkkijonoksi.
<code>getCurrentPage</code>	palauttaa käyttäjän käsittelemän sivun URL:n.
<code>quote</code>	lisää merkkijonon ympärille tavalliset lainausmerkit (").
<code>sqlQuote</code>	lisää merkkijonon ympärille yksinkertaiset lainausmerkit (').
<code>sqlTrim</code>	muuntaa Encode-luokan SQLEncode-metodilla merkkijonon SQL:n hyväksymään muotoon ja poistaa siitä ylimääräiset tyhjät merkit.

7.3.4 SimpleDb.java

SimpleDb-papu sisältää tietokannan käyttöä helpottavia yleiskäyttöisiä metodeja. Se sisältää attribuuttien get- ja set-metodien lisäksi seuraavat metodit:

<code>addSqlDeleted</code>	lisää SQL-lauseeseen <code>deleted</code> -kenttien tarkistuksen annetuille tauluille.
<code>addSqlFrom</code>	lisää SQL-lauseeseen <code>FROM</code> -osan annetuille tauluille.
<code>checkIfEditable</code>	tutkii, voiko annettua tietokantakyselystä saattaa <code>RS2</code> -joukkoa käydä läpi. Jos voi, asettaa kursorin joukon ensimmäiseen tietueeseen.
<code>executeTransaction</code>	käynnistää <code>StatementContainer</code> -luokassa määritellyn SQL-lausesarjan. Lauseet suoritetaan transaktiona.
<code>formatOrSet</code>	palauttaa muotoiltuna SQL-lauseen osan, jossa on listattuna <code>OR</code> -ehdolla annettuja numeroita määrätyssä taulussa.
<code>formatSearchString</code>	käsittelee <code>HtmlBean</code> -pavussa annetun <code>createSearchForm</code> -tyyliseen hakulomakkeeseen annetun merkkijonon ja antaa sen edelleen <code>Tools</code> -luokan <code>parseSearch</code> -metodille. Palauttaa SQL-lauseen osan.
<code>getString</code>	muuntaa tietokannan kyselystä saadun <code>RS2</code> -joukon merkkijonokentän niin, että mahdollisen <code>NULL</code> -viitteen sijaan palautetaan tyhjä merkkijono.
<code>recordExists</code>	tutkii, onko määrätyillä ehdoilla annettu tietue olemassa tietokannassa. Tulkitsee myös <code>deleted=true</code> -määritellyn tietueen olemassaolevaksi. Käytetään tietokohteiden käsittelijäluokkien metodissa, joka tallentaa olion tietokantaan.
<code>RS2toArray</code>	muuntaa <code>RS2</code> -hakujoukon taulukoksi.
<code>simpleQuery</code>	on yksinkertainen SQL-kysely, joka luo tietokantayhteyden ja varmistaa, että se suljetaan ennen kuin metodista poistutaan.
<code>simpleUpdate</code>	on yksinkertainen SQL-komento, joka luo tietokantayhteyden ja varmistaa, että se suljetaan ennen kuin metodista poistutaan.

7.3.5 StatementContainer.java

StatementContainer-papu on säiliöluokka StatementItem-luokasta perityille olioille. Luokka automatisoi parametrisoitujen SQL-lauseiden ajoa. Se sisältää attribuuttien get- ja set-metodien lisäksi seuraavat metodit:

addStatement	lisää säiliöön uuden parametrin.
prepare	valmistelee SQL-lauseen.
prepareItems	valmistelee SQL-lauseen parametrin. Metodia kutsutaan, jos parametreja halutaan muuttaa sen jälkeen, kun lause on alunperin valmisteltu.

7.3.6 StatementItem.java

StatementItem-papu on yliluokka valmistellun SQL-lauseen parametrille. Se määrittelee attribuuttien get- ja set-metodien lisäksi metodin assignItem, joka vastaa yksittäisen parametrin valmistelusta oikealla tietotyypillä.

7.3.7 IntItem.java

IntItem on peritty StatementItem-luokasta (luku 7.3.6). Se kapseloi kokonaislukumuotoisen SQL-lauseen parametrin. Papu ylikirjoittaa metodin assignItem.

7.3.8 StringItem.java

StringItem on peritty StatementItem-luokasta (luku 7.3.6). Se kapseloi merkkijonomuotoisen SQL-lauseen parametrin. Papu ylikirjoittaa metodin assignItem.

7.3.9 KiuruHandler.java

KiuruHandler on abstrakti yliluokka tietokannan tietokohteiden ja hakuluokkien käsittelijöille. Luokkien yhteisenä tekijänä on ”toiminnon” käsite. Toimintoja voivat olla esim. tietojen tallennus tietokantaan tai haku jollakin kriteerillä. Toiminnon määrää olion tila.

JSP-sivuja käytettäessä olion tila määräytyy HTTP-parametrien avulla ja toiminto ajetaan aina sivulle tultaessa. Toiminto yksinkertaistaa merkittävästi JSP-sivujen koodausta. JSP-käyttöliittymäsivulla ei tarvitse erikseen käsitellä, mitä sivun pitäisi tehdä, vaan riittää kutsua `performAction`-metodia. Lisätietoja JSP-sivujen toimintalogiikasta on luvussa [4.2](#).

Papu sisältää attribuuttien `get`- ja `set`-metodien lisäksi seuraavat metodit:

<code>performAction</code>	suorittaa toiminnon. Toiminnot määrätään yleensä JSP-sivujen parametreina. Eri JSP-sivujen parametrit on kuvattu luvussa 6 .
<code>resetErrorState</code>	palauttaa mahdollisen virheilmoituksen (tai tilannetietoja käyttäjälle), tyhjentää <code>Error</code> -olion ja asettaa olion normaaliin tilaan (jos on oltu virhetilassa). Metodia kutsutaan yleensä JSP-sivun alussa.

7.3.10 KiuruSpace.java

KiuruSpace abstrahoi tietokannan Space-aulun tietueen. Se on peritty KiuruHandler-luokasta (luku [7.3.9](#)). Papu ylikirjoittaa metodin `PerformAction` sekä sisältää attribuuttien `get`- ja `set`-metodien lisäksi seuraavat metodit:

<code>addSpaceGroups</code>	liittää salin yhteen tai useampaan saliryhmään.
<code>assignSpaceFromDataBase</code>	asettaa salin attribuutit tietokannasta.
<code>checkReservationRequestRight</code>	tutkii, onko käyttäjällä oikeus tehdä varauspyyntöjä annettuun saliin.
<code>checkUserRights</code>	tutkii, onko käyttäjällä oikeus vahvistaa varauksia annettuun saliin.
<code>deleteRecord</code>	poistaa salin tietokannasta.
<code>putSpaceToDataBase</code>	tallentaa salin attribuutit tietokantaan.
<code>resetActionState</code>	tyhjentää pavun toiminnon. Metodia kutsutaan yleensä JSP-käyttöliittymäsivun alussa.

7.3.11 SpaceGroup.java

SpaceGroup abstrahoi tietokannan SpaceGroup-taulun tietueen. Se on peritty KiuruHandler-luokasta (luku 7.3.9). Papu ylikirjoittaa metodin PerformAction sekä sisältää attribuuttien get- ja set-metodien lisäksi seuraavat metodit:

addPersons	lisää henkilöitä saliryhmän vahvistajiksi tai muokkaajiksi.
addSpaces	lisää saleja saliryhmään.
addToFavorites	lisää annetut salit käyttäjän mielisaliryhmään.
assignSpaceGroupFromDataBase	asettaa saliryhmän attribuutit tietokannasta.
deletePersons	poistaa henkilöitä saliryhmästä.
deleteRecord	poistaa saliryhmän tietokannasta.
deleteSpaces	poistaa saleja saliryhmästä.
putSpaceGroupToDataBase	tallentaa saliryhmän attribuutit tietokantaan.
resetActionState	tyhjentää pavun toiminnon. Metodia kutsutaan yleensä JSP-käyttöliittymäsivun alussa.
updateModifyRight	tarkistaa käyttäjän muokkausoikeudet saliryhmän tietoihin.

7.3.12 KiuruReservation.java

KiuruReservation abstrahoi tietokannan Reservation-taulun tietueen. Papu sisältää attribuuttien get- ja set-metodien lisäksi seuraavat metodit:

addPersonToReservation	liittää henkilön varaukseen määrättyssä roolissa.
isReservationOk	tarkistaa onnistuiko varaus.
isReserved	tarkistaa, onko annettu sali varattu annettuna aikana.
makeReservation	varaa salin annetuilla tiedoilla.

7.3.13 SearchHandler.java

SearchHandler on abstrakti ylliluokka hakuluokkien käsittelijöille. Se on peritty KiuruHandler-luokasta (luku 7.3.9). Luokkien yhteisenä tekijänä on ”toiminnon” käsitteen lisäksi kenttien järjestys.

Papu sisältää attribuuttien get- ja set-metodien lisäksi seuraavat metodit:

assignDefaultOrder	järjestää haun kentät oletusjärjestykseen.
getOrderClause	palauttaa järjestysosan SQL-lauseesta.

7.3.14 SpaceSearch.java

SpaceSearch on hakuluokka saleille. Se on peritty SearchHandler-luokasta (luku 7.3.13). Papu ylikirjoittaa metodin PerformAction sekä sisältää attribuuttien get- ja set-metodien lisäksi seuraavat metodit:

getFavoriteSpaces	palauttaa käyttäjän mielisalit.
getSpaceGroupSpaces	palauttaa annetun saliryhmän salit.
getSpaces	palauttaa kaikki salit, jotka sopivat annettuihin hakukriteereihin.
getSpacesCode	palauttaa koodin salille, jonka tunniste on annettu.
resetActionState	tyhjentää pavun toiminnon. Metodia kutsutaan yleensä JSP-käyttöliittymäsivun alussa.
updateCleared-SearchField	varmistaa hakuehdon tyhjentämisen, kun käyttäjä hakee tyhjällä merkkijonolla tietoa. Metodia kutsutaan yleensä JSP-käyttöliittymäsivun alussa.

7.3.15 PersonSearch.java

PersonSearch on hakuluokka henkilöille. Se on peritty SearchHandler-luokasta (luku 7.3.13). Papu ylikirjoittaa metodin PerformAction sekä sisältää attribuuttien get- ja set-metodien lisäksi seuraavat metodit:

<code>getPersonIdsFromSpaceGroup</code>	palauttaa annettuun saliryhmään liittyvien henkilöiden tunnistet.
<code>getPersons</code>	palauttaa kaikki henkilöt, jotka sopivat annettuun hakuhehtoon.
<code>getPersonsFromSpaceGroup</code>	palauttaa annettuun saliryhmään liittyvät henkilöt.

7.3.16 SpaceGroupSearch.java

`SpaceGroupSearch` on hakuluokka saliryhmille. Se on peritty luokasta `SearchHandler` (luku 7.3.13). Papu ylikirjoittaa metodin `performAction` sekä sisältää attribuuttien `get-` ja `set-`metodien lisäksi seuraavat metodit:

<code>getFavoriteSpaceGroups</code>	palauttaa tunnisteen käyttäjän mielisaliryhmälle, jos sellainen on olemassa.
<code>getSpaceGroups</code>	palauttaa kaikki saliryhmät.
<code>getSpaceGroupsMatched</code>	palauttaa saliryhmät, jotka sopivat annettuun hakuhehtoon.
<code>getSpaceGroupWithAccessLevel</code>	palauttaa saliryhmät, joita käyttäjällä on oikeus muokata.
<code>updateClearedSearchField</code>	varmistaa, hakuhehdon tyhjentämisen, kun käyttäjä hakee tyhjällä merkkijonolla tietoa. Metodia kutsutaan yleensä JSP-käyttöliittymäsivun alussa.

7.3.17 TimmiParser2.java

`TimmiParser2`-papu jäsentää XML-muotoista dataa muodostaen näistä ylläluokan `TimmiObject` perilliset sisältävän listan. Konstruktorille voidaan parametrina antaa jäsennettävän tiedoston nimi tai HTTP-vasteesta muodostettava `InputSource`-tyyppinen parametri. Varsinaisesta parsinnasta huolehtivia metodeita kutsutaan konstruktorissa, ja ne muodostavat löytämistään olioista erilliset listat. Kaikki oliot lisätään kuitenkin yleiseen `TimmiObject`-muotoiseen listaan.

Jäsentävien metodien apumetodi on seuraava:

`getParameter` on apumetodi, jota käytetään muissa jäsentävissä metodeissa. Se hakee annetusta `Nodelist`-listasta annetun parametrin nimisen tagin arvon.

Oliolistoja muodostavat metodit saavat parametrinaan juurielementin, josta jäsenitys aloitetaan. Näitä metodeja ovat seuraavat:

<code>setError</code>	palauttaa Timmi-rajapinnan tuottamasta vasteesta jäsenneen virhekoodin.
<code>setClients</code>	muodostaa listan jäsenneestä datasta löydettyistä <code>TimmiCustomer</code> -olioista.
<code>setContactPersons</code>	muodostaa listan jäsenneestä datasta löydettyistä <code>TimmiContactPerson</code> -olioista.
<code>setNotReservedReservations</code>	muodostaa listan ajoista, joille varausta ei voitu vahvistaa.
<code>setReservations</code>	muodostaa listan jäsenneestä datasta löydettyistä <code>TimmiReservation</code> -olioista.
<code>setSpaces</code>	muodostaa listan jäsenneestä datasta löydettyistä <code>TimmiSpace</code> -olioista. Samalla muodostuu lista <code>TimmiUseType</code> -olioista, sillä <code>TimmiSpace</code> voi sisältää useita <code>TimmiUseType</code> -olioita.
<code>setBuildings</code>	muodostaa listan jäsenneestä datasta löydettyistä <code>TimmiBuilding</code> -olioista.

7.3.18 `Timmi2KiuruSynchronizer.java`

`Timmi2KiuruSynchronizer` on Timmi- ja Korppi-järjestelmien välisestä synkronoinnista huolehtiva papu. Se kysyy Timmi-rajapinnalta parametreillä määritellyn ajanjakson tiedot. Se käyttää `TimmiParser2`:lta (katso luku 7.3.17) saamaansa oliolistaa hyväkseen pyytäen jokaista oliota tallentamaan itsensä Korppi-järjestelmän tietokantaan.

Seuraavat metodit vastaavat varsinaisista kyselyistä:

<code>doRequest101</code>	tekee Timmi-rajapintadokumentin luvun 3.1 mukaisen kyselyn ja tallentaa tulokset Korppi-järjestelmän tietokantaan.
<code>doRequest102</code>	tekee Timmi-rajapintadokumentin luvun 3.2 mukaisen kyselyn ja tallentaa tulokset Korppi-järjestelmän tietokantaan.
<code>doRequest104</code>	tekee Timmi-rajapintadokumentin luvun 3.4 mukaisen kyselyn ja tallentaa tulokset Korppi-järjestelmän tietokantaan. Tämän metodin toimivuuden edellytyksenä on <code>doRequest101</code> -metodin muodostama lista.

7.3.19 TimmiLog2Kiuru.java

`TimmiLog2Kiuru` on Timmin muutoslokin tietojen siirtämisestä Korppi-järjestelmän Kotka-tietokantaan huolehtiva papu. Se tarkkailee tiettyä IP-osoitetta ja porttia, josta annettu heräte on merkki lokitiedoissa tapahtuneesta muutoksesta. Komponentti jäsentää XML-muotoiset lokitiedostot ja tallentaa ne Korppi-järjestelmän tietokantaan.

Komponentilla on seuraavat kaksi metodia:

<code>accept</code>	määrittelee kelvollisten lokitiedostojen tiedostonimet käytettäväksi lokitiedoista muodostettavaa listaa generoitaessa.
<code>run</code>	suoritetaan, kun ohjelma on saanut herätteen ennalta määritellystä IP-osoitteesta ja portista. Metodi hakee lokitiedostolistan ja jäsentää tiedostot ja tallentaa tulokset. Tämän jälkeen käsitelty lokitiedosto poistetaan.

7.3.20 TimmiObject.java

`TimmiObject`-papu on ylikuokka pavuille `TimmiBuilding`, `TimmiContactPerson`, `TimmiCustomer`, `TimmiReservation`, `TimmiSpace` ja `TimmiUserType`. Pavun tarkoitus on tarjota rajapinta em. muille pavuille. Ainoa vaatimus, jonka `TimmiObject`-papu asettaa, on metodin `saveToDB` toteutus.

7.3.21 TimmiContactPerson.java

TimmiContactPerson-papu sisältää yhden Timmissä olevan yhteyshenkilön tiedot. Pavun metodit ovat sen attribuuttien get- ja set-metodien lisäksi saveToDB. Metodia ei ole toteutettu ja sen toteutus on sovittu jatkokehitykseen.

7.3.22 TimmiCustomer.java

TimmiCustomer-papu sisältää yhden Timmissä olevan asiakkaan tiedot. Pavun metodit ovat sen attribuuttien get- ja set-metodien lisäksi saveToDB. Metodia ei ole toteutettu ja sen toteutus on sovittu jatkokehitykseen.

7.3.23 TimmiReservation.java

TimmiReservation-papu sisältää yhden Timmissä olevan varauksen tiedot. Pavun metodit ovat sen attribuuttien get- ja set-metodien lisäksi seuraavat:

saveToDB	tallentaa tietonsa Kotka-tietokantaan.
checkExistance	tarkistaa, onko varaus Kotka-tietokannassa. Jos on, niin papu myös tarkistaa, mitä muutoksia varaukseen on tullut tai onko muutoksia ollenkaan. Tätä metodia SaveToDB-metodi käyttää joissain tapauksissa apunaan.

7.3.24 TimmiBuilding.java

TimmiBuilding-papu sisältää yhden Timmissä olevan rakennuksen tiedot. Attribuutti tilat sisältää listan rakennuksessa olevista tiloista eli TimmiSpace-olioista. Pavun metodit ovat sen attribuuttien get- ja set-metodien lisäksi seuraavat:

saveToDB	tallentaa tietonsa Kotka-tietokantaan. Metodia ei ole toteutettu ja sen toteutus on sovittu jatkokehitykseen.
AddTila	lisää tilat-listaan yhden tilan. Tilat-attribuutilla ei siis ole varsinaista set-metodia.

7.3.25 TimmiSpace.java

TimmiSpace-papu sisältää yhden Timmissä olevan tilan tiedot. Attribuutti `kayttoTyypit` sisältää listan tilassa olevista käyttötyypeistä eli `TimmiUseType`-olioista. Pavun metodit ovat sen attribuuttien `get`- ja `set`-metodien lisäksi seuraavat:

<code>saveToDB</code>	tallentaa tietonsa Kotka-tietokantaan. Metodia ei ole toteutettu ja sen toteutus on sovittu jatkokehitykseen.
<code>AddKayttoTyyppi</code>	lisää <code>kayttoTyypit</code> -listaan yhden käyttötyypin. <code>KayttoTyypit</code> -attribuutilla ei siis ole varsinaista <code>set</code> -metodia.

7.3.26 TimmiUseType.java

`TimmiUseType`-papu sisältää yhden Timmissä olevan käyttötyypin tiedot. Pavun metodit ovat sen attribuuttien `get`- ja `set`-metodien lisäksi `saveToDB`. Metodia ei ole toteutettu ja sen toteutus on sovittu jatkokehitykseen.

8 Sovelluksen testaaminen

Luvussa tarkastellaan sovelluksen testaamisen yleisiä periaatteita ja esitellään käytettyjä testausympäristöjä. Projektiryhmä ei ehtinyt toteuttaa kaiken kattavaa testausta projektin puitteissa. Kuitenkin jokaisen komponentin valmistuttua komponentin kirjoittaja teki muutamia testejä, jotta karkeimmat ohjelmointivirheet saatiin karsittua.

8.1 Testausympäristöt

Sovelluksen testaamiseen käytettiin muutamaa eri selainversiota. Tällä pyrittiin varmistamaan edes jonkinlainen käytettävyys kaikilla yleisimmillä selaimilla. Taulukossa 1 on esitelty sovelluksen testauksessa käytetyt käyttöjärjestelmät, selaimet ja selainversiot.

KÄYTTÖJÄRJESTELMÄ	SELAIN	VERSIOT
Windows 2000	Internet Explorer	5.0 ja 6.0
	Mozilla	1.1
	Opera	6.3
Windows 98	Netscape Navigator	7.0
Red Hat Linux	Opera	6.3
	Mozilla	1.1

Taulukko 1: Testauksessa käytetyt käyttöjärjestelmät ja selaimet.

Huomattavia poikkeuksia ei ilmennyt eri selainten kesken. Joitakin ongelmia oli kirjasimiin liittyen, mutta nämä ongelmat poistuvat, kun Kiuru-sovellus yhdistetään uusimpien Korppi-järjestelmän kehitysversioiden kanssa. Kirjasinongelmat liittyvät CSS-tyylitiedostoihin. Kyseisiin asioihin on paneuduttu Korppi-järjestelmän kehitystyössä sen jälkeen, kun Kiuru-projektin versionhallinnan kehityshaara eriytettiin päähaarasta.

8.2 Testausperiaatteita

Sovellusta pyrittiin testaamaan koko kehityksen ajan. Loppuvaiheessa ilmeni pieniä vikoja, jotka johtuivat eri komponenttien yhteentoimimattomuudesta. Nämä viat saatiin kuitenkin korjattua helposti grep-työkalun avustuksella.

Sovellukselle ei kuitenkaan ajettu erillisiä, automaattisesti toistettavia testitauksia, mitä voidaan pitää suurena puutteena. Jos projektille olisi asetettu enemmän aikaa, sovellukselle olisi voinut suorittaa perusteellisen integraatiotestauksen. Ohjaajista Jukka-Pekka ja tilaajan edustajista Vesa testasivat kiitettävästi järjestelmää ja heiltä saimmekin paljon parannusehdotuksia ja tietoa järjestelmän virheistä. Suurimman osan virheistä saimme korjattua, mutta virheitä löytyy varmasti vielä lisää vähäisestä testauksesta johtuen. Suurin jatkokehitystarve onkin tehdyn järjestelmän testaus.

8.3 Testaustulokset

Koska sovellusta ei ole testattu tarpeeksi, ei ryhmällä ole esittää järjestelmällisen ja kattavan testauksen tuloksia. Vikoja on korjattu sitä mukaa, kun niitä on ilmennyt. Näihin liittyviä asioita käsitellään luvuissa [9](#) ja [10](#)

9 Ohjelman virheet ja toiminnan puutteet

Luvussa kuvataan Kiuru-projektin toteuttamaan sovellukseen jääneitä virheellisiä ja puutteellisia 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.

9.1 Navigointi

Navigointipuun selkeyttä on usein kyseenalaistettu, eikä projektin puitteissa ehditty panostamaan tämän osa-alueen kehittämiseen. Navigointipuun täytyisi olla selkeä ja looginen sekä ominaisuuksien löytäminen tulisi olla helpompaa. Kiuru-projekti ei tuonut paljon uutta navigointirakenteeseen, joten siltä osin navigointipuun kehittäminen ei projektin piiriin kuulunutkaan.

9.2 Varausmoduuli

Varausmoduulissa ei toimi varausprosessin virhetarkistukset eikä moduulissa ole mahdollisuutta selailta varauksia käyttäjien mukaan. Varausprosessin virhetarkistukset ovat varausmoduulin suurin ongelma. Sovellus toimii, jos käyttäjä ei tee mitään väärin, vaan etenee lineaarisesti varausprosessin alusta loppuun saakka. Varausmoduulin saa kaatumaan antamalla väärän muotoisia syötteitä HTML-sivujen lomakkeisiin. Myös radiopainikeryhmän jättäminen kokonaan tyhjäksi kaataa sovelluksen. Sovellus kuitenkin toipuu virheestä painamalla selaimen takaisinpainiketta ja korjaamalla virheellisen syötteen.

Käyttäjakohtainen varausten seuranta ei toimi. Tämä ominaisuus olisi hyödyllinen lähinnä tilastointi- ja laskutusoperaatioissa. Vastaavia ominaisuuksia ei tällä hetkellä kuitenkaan ole suunniteltu Korppi-järjestelmään, joten tämä ei liene suuri puute.

9.3 Synkronointi ja lokitiedostojen käsittely

Timmi- ja Korppi-järjestelmien tietokantojen synkronointi toteutettiin siten, että erillinen synkronointikomponentti hoitaa koko kannan lukemisen Timmistä ja tietojen tallentamisen tietokantaan. Tämä operaatio on tarkoitettu toteutettavaksi kerran päivässä tai harvemmin.

Reaaliaikaisempi Korppi-järjestelmän salivaraustietojen synkronointi hoidetaan lokitiedostojen avulla. Lokitiedostot ovat Timmi-järjestelmän generoimia XML-muotoisia tiedostoja Timmi-järjestelmän tietokantaan tehdyistä muutoksista.

Synkronointiin liittyvät komponentit on toteutettu, mutta niitä ei ole testattu lähinnä Timmi-rajapinnan määrittelyn ja toteutuksen viivästymisen, sekä muiden rajapinnassa esiintulleiden ongelmien vuoksi. Lisäksi testaamatta jäi Vesa Lappalaisen toteuttaman lokihakemistoon tapahtuneista muutoksista ilmoittavan ohjelman toiminta.

Timmi-järjestelmän lokien generointia ei ole pystytty testaamaan, koska projektiryhmän käytössä ei ole ollut WinTimmi 2.0 -versiota. Tämän vuoksi lokien käsittelyn testausta ei ole voitu suorittaa, vaan testausta on suoritettu lähinnä kehitysvaiheessa mallisyötteillä.

9.4 Virheet käyttäjän syötteissä

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

9.5 Selaimiin liittyvät virheet

Joillakin selaimilla (esimerkiksi Windows-käyttöjärjestelmässä Operan kaikilla versioilla) käyttäjän jo tekemät valinnat ja syöttämät arvot häviävät, jos sivu ladataan uudelleen, vaikka parametreinä valinnat ja arvot tuodaankin. Tämä on selainriippuvainen ongelma, eikä siihen ole löytynyt ratkaisua.

10 Jatkokehitykseen siirretyt ominaisuudet

Luvussa kuvataan Kiuru-sovellukseen liittyviä parannusehdotuksia, joita projektin aikataulussa ei ehditty toteuttamaan. Jatkokehitykseen sovittiin myös monet käyttöliittymään liittyvät korjaukset. Nämä tulivat esille vasta projektin loppuvaiheessa, kun oli saatu aikaan jotain testattavaa. Osa tarvittavista korjauksista on kuitenkin toteutettu ja lähinnä jatkokehitykseen on jätetty ne ominaisuudet, joiden toteutus ei ole itsestäänselvää ja helppoa. Lukuun 10.4 on kirjattu jo projektin alussa esitetyt toiveet, joita projektin aikataulussa ei ehditty toteuttamaan.

10.1 Tietokannan rakenteeseen liittyvät kehitysideoit

Kotka-tietokantaan ja opettajan käyttöliittymään tulee lisätä mahdollisuus tilatarpeiden eli resurssipyynnöjen merkitsemiseen. Tilatarpeiden merkitseminen hoidetaan tällä hetkellä siten, että kurssin opettaja toimittaa tilatarpeet eteenpäin esim. amanuenssille. Amanuenssi suunnittelee kootusti laitoksen kurssien opetustapah- tumien sijoittamisen käytössä oleviin saleihin.

Tilatarpeet sisältävät vaatimuksia salin paikkamäärään, tyyppiin (esim. luentosali tai mikroluokka) sekä tarvittaviin resursseihin (esim. videoprojektori ja liitutaulu). Resurssit voivat liittyä saliin kiinteästi tai ne voivat olla erikseen varattavia. Näitä lisäyksiä voidaan käyttää hyväksi Kiuru-sovelluksen jatkokehityksessä.

Käyttäjaoikeuksien tarkastusta ei tietokannan päivityksiä tehtäessä systemaattisesti suoriteta. Jatkokehityksessä tulisikin käyttöoikeuksien tarkastus suorittaa ennen jokaista tietokannan päivitysoperaatiota.

10.2 Kehitysideoita sovelluksen toiminnasta

Varausmoduulin toimivuus sovelluksen tärkeimpänä komponenttina tulisi varmistaa. Lähinnä syytä olisi keskittyä virhetarkistuksiin, sillä ne estävät moduulin laajamittaisen käytön. Nykyinen versio sovelluksesta toimii, jos käyttäjä ei tee mitään väärin, vaan etenee lineaarisesti varausprosessin alusta loppuun saakka. Varausmoduulin saa kaatumaan antamalla väärän muotoisia syötteitä HTML-sivujen lomakkeisiin. Myös radiopainikeryhmän jättäminen kokonaan tyhjäksi kaataa sovelluksen. Sovellus kuitenkin toipuu virheestä painamalla selaimen takaisin-painiketta ja korjaamalla virheellisen syötteen.

Lisäksi varausmoduulista puuttuu ominaisuus **varauksien selailuun käyttäjien mukaan**. Käyttäjakohtainen varausten seuranta olisi hyödyllinen lähinnä tilastointi- ja laskutusoperaatioissa.

Järjestelmään voisi kehittää **Osittain vapaiden salien hakemisen** vakiovarausta tehtäessä. Tällä tarkoitetaan sitä, että hakuehdoissa voitaisiin määritellä, että jos sali on vapaana tiettyyn annettuun määrään tapahtumia, niin kyseinen sali näytetään hakutuloksissa. Tällä hetkellä haku palauttaa vain salit, jotka ovat vapaana kaikkina vakiovarauksessa ilmoitettuina päivinä. Tämän ominaisuuden toteutus vaatii erilaisen ajattelutavan, mitä toteutetussa versiossa on käytetty. Kyseinen muutostoive tuli esiin sen verran myöhäisessä vaiheessa projektia, että se päätettiin siirtää jatkokehityksen varaan. Ominaisuus kuitenkin helpottaa järjestelmän käyttöä, joten sen prioriteettia voidaan pitää suhteellisen korkeana.

Koodin ja metodien kieliasun yhtenäistäminen olisi tärkeää järjestelmän ylläpidettävyyden kannalta. Vaikka Kiuru-projekti pyrki englanninkielisen koodin tuottamiseen, jäi kehitysversioihin vielä suomenkielisiäkin metodeja ja parametrejä. Esimerkkinä tästä on sivun `makeReservation.jsp` käsittelijän `teeVarauspyynto`-parametri. Näiden korjaaminen parantaisi järjestelmän ylläpidettävyyttä ja yhtenäistäisi rakennetta, vaikkei mitään uutta toiminnallisuuteen toisi-kaan.

WWW-sivujen uudelleenohjaus hoidetaan käsittelijäpavuuissa JSP:n `forward`-metodilla, joka jättää selaimen osoitekenttään maininnan `HandlerBean`istä. Tämä aiheuttaa ongelmia navigointipuun suhteen. Tähän apuna voisi miettiä joko esimerkiksi `forward`-metodin korvaamista Javan `response.sendRedirect`-metodilla tai navigointipuun toteutuksen muuttamisella. Järjestelmään tulisi lisätä **sivu, joka hakee tietoja vapaista ajoista ja paikoista** käyttäjän antamalla kriteereillä. Tällä hetkellä järjestelmässä pystyy selailemaan vain omia varauksiaan ja varausta tehdessä etsiä vapaita aikoja.

10.3 Automaattisten sähköpostiviestien puutteista

Tällä hetkellä Kiuru-järjestelmä ilmoittaa varauspyynnöstä aina myös sähköpostitse vahvistusoikeuksien haltijalle. Järjestelmässä olis hyvä olla mahdollisuus valita, miten ilmoitukset toimitetaan. Näin käyttäjä voisi valita, että toimitetaanko ilmoitus Korppi-järjestelmän ilmoitustaululle, sähköpostitse tai molemmilla tavoilla.

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

10.4 Muita jatkokehitysideoita

Jo projektin alkuvaiheessa tuli esille toiveita tilatarpeiden suunnitteluun ja muokkaukseen käytettävästä ohjelmasta. Kurssien sijoittumisen suunnittelusta vastaavat henkilöt tarvitsisivat ohjelmaa, jonka avulla voisi sijoittaa kurssien ajankohdat siten, että opiskelijoilla ja luennoitsijoilla olisi mahdollisimman vähän päällekkäisyyksiä. Ohjelman tulisi ainakin ilmoittaa, jos esimerkiksi saman vuositason tai aineen kurssija on menossa samaan aikaan. Ihanteellinen tilanne olisi, jos ohjelma voisi sijoittaa kurssit valmiiksi lukujärjestykseen. Nämä vaatimukset on huomioitu tietokannassa, mutta varsinaista suunnitteluohjelmaa projektin resurssien puitteissa ei ehditty toteuttamaan.

Sali-salinvarausjärjestelmän korvaava Timmi-järjestelmä tuo uusia toiminnallisuuksia vain laskutuksen ja tilastoinnin osalta. Nämä ominaisuudet lienevät myös toteutettavissa tarvittaessa Korppi-järjestelmän jatkoksi.

Kaiken kaikkiaan jatkokehityksen suuntalinjoja on vaikea hahmottaa. Ihanne-tilannehan olisi, jos samaa tietoa ei tarvitsisi monessa paikassa säilyttää. Toisaalta tarpeita on niin paljon, että kaikkien ominaisuuksia yhdistäminen samaan sovellusympäristöön ei välttämättä olekaan paras tapa menetellä. Samaan aikaan Kiuru-sovelluksen kanssa kehitetty Tuure-järjestelmän seuraaja Tutka-järjestelmäkin voitaisiin ainakin periaatetasolla toteuttaa myös Korppi-järjestelmän jatkoksi. Jatkokehitysmahdollisuuksia on siis useita. Epäselvää on kuitenkin se, että miten nykyinen järjestelmä skaalautuu tulevaisuuden vaatimuksiin ja miten tarkoituksenmukaista järjestelmän laajentaminen on.

Kiuru-projektin 1. projektikansion kohdasta "Muuta materiaalia" löytyy yhteenveto Vesa Lappalaisen jatkokehitysideoista.

11 Asennusohjeet

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

Koska sovellus on olennaisesti Korppi-järjestelmään liitettävä moduuli, oletetaan, että muut tarvittavat osiot ovat jo asennettu. Korppi-sovelluksen asennusohje on liitetty Korppi-projektin projektikansioon.

11.1 Tietokannan päivitys

Tietokannan luomiseen ja vanhan tietokannan muokkaamiseen tarvittavat SQL-lauseet on koottu yhteen hakemistoon. Ensin tulee ajaa `kiuru_create.sql`, joka luo Kiuru-sovelluksen tarvitsemat tietokantataulut. `kiuru_alter.sql` muuttaa olemassaolevia tauluja. Seuraavaksi kannattaa ajaa `kiuru_index.sql`, joka luo tarvittavat indeksit. `kiuru_insert.sql` lisää Kiuru-sovelluksen tarvitsemia tietoja tietokantatauluihin. Lopuksi ajetaan vielä `kiuru_rules.sql`, joka luo tietokantaan Kiuru-sovelluksen tarvitseman säännön.

Luontilauseet voidaan suorittaa periaatteessa millä tahansa tietokannan käsittelyyn tarkoitettulla hallintajärjestelmällä. `kiuru_rules.sql` ei välttämättä toimi sellaisenaan muilla tietokannanhallintajärjestelmillä kuin PostgreSQL:llä.

Kiuru-sovelluksen käyttämät tietokannan taulut voidaan poistaa kannasta ajamalla tiedoston `kiuru_drop.sql` sisältämät lauseet.

11.2 Papujen asentaminen

Kiuru-sovelluksen käyttämät pavut sijaitsevat CD-ROMin hakemistossa `/kotka/WEB-INF/classes/kiurubeans/`. Pavut kannattanee kääntää uudelleen asennusympäristössä `java`-komennolla.

Muokatun `HTMLMenu`-pavun lähdekoodi on tiedostossa `HTMLMenu.java`. Myös tämä papu kannattaa kääntää uudelleen asennusympäristössä. Käännöksen tuloksena saatu `HTMLMenu.class`-tiedosto korvaa aiemman version.

11.3 JSP-sivujen asentaminen

Sovelluksen käyttämät JSP-sivut tulee kopioida `/kotka/`-hakemistoon. Tiedostojen asentaminen ei vaadi erityisiä toimenpiteitä.

Kiuru-sovelluksen suoritettavat JSP-sivut (katso luku 6) asennetaan hakemistoon `/kotka/reservation/`. Sivut ovat suorituskelpoisia sellaisenaan. TomCat-sovelluspalvelin kääntää JSP-sivut tarpeen mukaan.

11.4 Muut toimenpiteet

Sovelluksen käyttämiseksi täytyy muuttaa olemassaolevaa tietokannan dataa. `space`-taulun tietueille täytyy määritellä tilatyypit, jotta haut toimivat oikein.

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

12 Ylläpitäjän ohjeet

Luvussa kuvataan Kiuru-sovelluksen ylläpitäjän huomioitavia asioita. Erityisesti sovelluksen asennukseen (katso luku 11) liittyy joitain ylläpitäjän vastuulle asettuja tehtäviä.

12.1 Oikeuksien asettaminen

Sovelluksen asentamisen jälkeen ylläpitäjän täytyy määritellä hallinnolliset saliryhmät ja määritellä niihin varauspyyntöjen oikeustasot. Lisäksi varausten vahvistajille on annettava vahvistusoikeudet. Ilman oikeuksien asettamista sovelluksen käyttö rajoittuu mielisaliryhmien ja henkilökohtaisten hakuryhmien käyttöön.

Oikeuksia voi muokata ylläpidon käyttöliittymän sivuilla, joiden tekninen toteutus on kuvattu luvussa 6.4.

12.2 Ristiriitaiset varaukset

Varausjärjestelmä on rakennettu toisaalta kalenterin, toisaalta opettajan kurssitietojen määrittämisen päälle. Tietokannassa on ollut valmiina tapa salien yhdistämiseen tapahtumiin. Kiuru-projekti lisäsi varausjärjestelmää varten uudet taulut, jotka ovat osittain päällekkäin olemassaolevien taulujen (erityisesti `EventSpace`) kanssa. On mahdollista, että `EventSpace`-taulussa olevat tiedot ovat ristiriitaisissa `Reservation`-taulussa olevien tietojen kanssa.

Ristiriitaisuuksien havaitsemiseksi projekti lisäsi tietokantaan säännön, joka asettaa ristiriitatilanteissa `Reservation`-taulun `Inconsistent`-kentän arvoksi 1. Ristiriitojen hallintaan ei ehditty tehdä omaa käyttöliittymää, joten ylläpitäjän vastuulla on selvittää ongelmatilanteet, jos niitä ilmenee. Lisätietoa säännöstä on Kotka-tietokantaraportin luvussa 11.

12.3 Monikielisyys

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

Kiuru-projekti ei ole lisännyt uusien tietokantataulujen yhteyteen erillisiä käännöstauluja. Kaikki käännökset on hoidettu `User-pavun T`-metodilla.

13 Yhteenveto

Kiuru-projektiryhmä toteutti Korppi-järjestelmään tilanvarausmoduulin. Projekti sai toteutettua kaikki vaatimusmäärittelyssä mainitut kohdat lukuunottamatta varusten selailemista henkilöiden mukaan.

Sovellusraportissa kuvataan sovellukseen toteutetut ominaisuudet. Sovelluksen rakenne kuvataan raportissa yksityiskohtaisesti. Sovellusraporttiin on lisäksi kirjattu havaitut virheet ja puutteet sovelluksen toiminnassa. Toteuttamatta jääneet tai osittain toimivat osat on kuvattu raportissa jatkokehitysideoina. Jatkokehitykseen jäivät mm. resurssipyyntöjen käsittely ja oheisvarattavien käsittely.

Projektiryhmä toteutti myös rajapintakomponentit Timmi-järjestelmän kanssa kommunikoimista varten. Rajapintayhteyttä ei kuitenkaan testattu projektin puitteissa, vaan se jätettiin jatkokehitystehtäviksi. Syy tähän oli rajapinnan valmistuksen viivästyminen. Rajapinnan määrittely- ja toteutusvaiheet venyivät ja Yomi Oyj toimitti ensimmäisen testiversion projektiryhmän käyttöön vasta tammikuun 2003 alussa.

Lähteet

- [1] Hilpinen Toni, Koivuniemi Marko, Mäkinen Jussi ja Nurminen Miika ”Kiuru-projektin projektisuunnitelma”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [2] Hilpinen Toni, Koivuniemi Marko, Mäkinen Jussi ja Nurminen Miika ”Kiuru-projektin projektiraportti”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [3] Hilpinen Toni, Koivuniemi Marko, Mäkinen Jussi ja Nurminen Miika ”Kiuru-projektin sovellussuunnitelma”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [4] Hillebrand Minna, Silván Markus, Vanhanen Antti ja Ylitalo Marko, ”Koppelo-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [5] Jaakkola Mia, Juutinen Sanna, Lupari Matti ja Nieminen Mikko, ”Kolibri-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2001.
- [6] Lesonen Minna, Pekkanen Hannu, Tawast Tuukka ja Uuksulainen Heikki, ”Korppi-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2001.
- [7] Horppu Ismo, Mielityinen Markku ja Vire Markku, ”Kotka-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2000.
- [8] Horppu Ismo et al., ”Kotka-järjestelmän tietokantaraportti”, Jyväskylän yliopisto, tietotekniikan laitos, 2003.
- [9] Santanen Jukka-Pekka, ”Tietotekniikan Sovellusprojektien ohje”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [10] Santanen Jukka-Pekka, ”Opinnäytteiden kirjoittaminen, lyhyt oppimäärä”, Jyväskylän yliopisto, tietotekniikan laitos, 2000.