

KYYHKY-PROJEKTI

Timo Aarniovuori
Teija Alasalmi
Jaakko Hyvärinen
Maunu Tuomainen

Sovellussuunnitelma
17.5.2004 versio 1.0

Jyväskylän Yliopisto
Tietotekniikan Laitos

KYYHKY-PROJEKTIN TIETOJA

Tekijät:

Timo Aarniovuori (taarnio@cc.jyu.fi)

Teija Alasalmi (teikku@cc.jyu.fi)

Jaakko Hyvärinen (japahyva@cc.jyu.fi)

Maunu Tuomainen (mttuomai@cc.jyu.fi)

Yhteystiedot:

Kyyhky-projektin projektitila sijaitsee Agorassa, huoneessa C222.2. Huoneen puhelinnumero on 014-260 4963. Projektiorganisaation sähköpostilista on kyyhky@korppi.jyu.fi . Sen keskustelua voi seurata osoitteessa <https://korppi.it.jyu.fi/list-archive/kyyhky/>. Projektin kotisivu löytyy osoitteesta <http://sovellusprojektit.it.jyu.fi/kyyhky/>.

Työn nimi: Kyyhky-projekti, sovellussuunnitelma

Työ: Sovellussuunnitelma tietotekniikan Sovellusprojektiin

Tiivistelmä:

Kyyhky-projekti toteuttaa päiväkirjan rajapinnan Jyväskylän yliopiston tietotekniikan laitoksella toteutettuun Korppi-nimiseen järjestelmään. Rajapinnan avulla pystytään synkronoimaan tapahtumia Korppi-järjestelmän kalenteriosion ja erilaisten mikrojen, matkapuhelimien ja muiden kämmenlaitteiden kalenterien kesken. Sovellussuunnitelmassa kuvataan, miten järjestelmä toteutetaan.

Avainsanat: synkronointi, sähköinen kalenteri, kalenterisovellus, SyncML, Sync4j, WWW-sovellus, Korppi-järjestelmä.

VERSIOHISTORIA

Versio	Päiväys	Tehnyt	Muutokset
0.1	2.4.2004	Timo Aarniovuori	Ensimmäinen versio.
0.2	20.4.2004	Timo Aarniovuori	Toinen versio.
0.3	13.5.2004	Timo Aarniovuori	Kolmas versio. Muutettu kappalejakoja ja korjattu kirjoitusvirheitä.
1.0	17.5	Timo Aarniovuori	Neljäs versio. Korjattu viitteitä ja kirjoitusvirheitä, hyväksytty.

SISÄLLYSLUETTELO

1	JOHDANTO	1
2	TERMEJÄ	2
	2.1 Korppi-järjestelmään liittyvät termit	2
	2.2 Käytettäviin työkaluihin, ohjelmistoihin sekä ohjelmointikieliin liittyvät termit.....	2
	2.3 SyncML-tekniikkaan liittyvät termit.....	4
3	JÄRJESTELMÄN ARKKITEHTUURI, TOIMINTA JA TYÖKALUT	7
	3.1 Toteutettavan järjestelmän arkkitehtuuri.....	7
	3.2 Työkalut ja kehitysympäristöt	8
	3.3 JSP-tekniikka	9
4	SYNC4J-PALVELIN	10
	4.1 Kaksisuuntainen synkronointi	10
	4.2 Synkronointi Sync4j-synkronointipalvelimella.....	11
	4.2 Sync4j-synkronointipalvelimen Korppi-moduuli.....	12
5	KÄYTTÖLIITTYMÄ	15
	5.1 Käyttöliittymän JSP-sivujen rakenne.....	15
	5.2 Konflikti-ilmoitukset.....	17
	5.3 Konfliktien käsittely	18
	5.4 Ryhmätapahtuman ristiriidan käsittely	20
	5.5 Kalenterin tunnistetietojen muokkaus	22
6	TIETOKANTA	24
	6.1 Sync4j-palvelimen tarvitsemat taulut.....	24
	6.2 Sync4j-palvelimen ja Kotka-tietokannan yhteistoimintaan tarvittavat taulut.....	26
7	TESTAUSPERIAATTEITA	28
8	LÄHDEKOODIN KOMMENTOINTI JA NIMEÄMISKÄYTÄNNÖT	28
9	YHTEENVETO	34
	LÄHTEET	36
	LIITE. TIETOKANNAN ER-KAAVIO	37

1 JOHDANTO

Kyyhky-projekti on Jyväskylän yliopiston tietotekniikan laitoksella keväällä 2004 toteutettava sovellusprojekti. Projekti toteuttaa Korppi-opintotietojärjestelmään päiväkirjapinnan, jonka avulla erilaisten mikrojen, matkapuhelinten ja kämmenlaitteiden kalenterisovellukset voidaan synkronoida Korppi-järjestelmän kalenterin kanssa.

Kyyhky-projektiryhmään kuuluvat Timo Aarniovuori (TIE), Teija Alasalmi (ENG, TIE), Jaakko Hyvärinen (TJT) ja Maunu Tuomainen (MAT). Vastaavana ohjaajana toimii lehtori Jukka-Pekka Santanen ja teknisenä ohjaajana tietotekniikan opiskelija Tiina Pöyhönen. Tilaajan edustajana toimivat koordinaattori Antti Auer virtuaaliyliopistohankkeesta ja lehtori Vesa Lappalainen tietotekniikan laitokselta.

Dokumentissa kuvataan, miten päiväkirjapinta toteutetaan. Ryhmä toteuttaa myös muita dokumentteja. Vaatimusmäärittelyssä määritellään vaatimukset, jotka toteutettavan järjestelmän tulee täyttää. Projektisuunnitelmassa käsitellään projektin taustoja, tavoitteita, resursseja, organisaatiota, aikataulua ja läpivientiä, dokumentteja ja kokouskäytäntöä sekä projektin toteutumiseen liittyviä riskejä. Testaussuunnitelmassa esitellään projektissa toteutettavan järjestelmän testauksen toteutusstrategiat, testitapaukset ja testausympäristö. Projektiraportti laaditaan projektin lopussa ja siinä kuvataan projektin kulkua ja toteutumista. Sovellusraportti kuvaa toteutetun sovelluksen rakennetta ja toimintaa sekä toimivuutta ja tavoitteiden toteutumista.

Luvussa 2 kuvataan projektiin ja sovellukseen liittyviä termejä. Luvussa 3 esitellään järjestelmän arkkitehtuuria ja toimintaa sekä käytettyjä tekniikoita ja työkaluja. Luvussa 4 tarkastellaan Sync4j-palvelimen toimintaa ja sen Korppi-moduulia. Luvussa 5 esitetään käyttöliittymän JSP-sivujen rakenne ja konfliktienhallintaosan käyttöliittymänäkymä. Luvussa 6 on kuvattu alustava tietokantasuunnitelma. Luvussa 7 käydään läpi testausperiaatteita ja luvussa 8 tarkastellaan lähdekoodin kommentointi ja nimeämiskäytäntöjä.

2 TERMEJÄ

Luvussa kuvataan Kyyhky-projektin dokumentoinnissa käytettyä termistöä.

2.1 Korppi-järjestelmään liittyvät termit

Korppi-järjestelmään ja sitä kehittäneisiin projekteihin liittyviä termejä ovat seuraavat:

Kiuru	toteutti Korppi-järjestelmään salivaraussovelluksen.
Korppi	on Jyväskylän yliopistossa kehitetty opintotietojärjestelmä.
Kotka	on Korppi-järjestelmän tietokanta ja henkilötietojen hallintaosio.
Kolibri	toteutti Korppi-järjestelmään päivyri- ja ajanvarausosion.

2.2 Käytettäviin työkaluihin, ohjelmistoihin sekä ohjelmointikieliin liittyvät termit

Sovelluksen toteutuksessa käytettäviin ohjelmointi- ja kuvauskieliin liittyviä termejä ovat seuraavat:

CSS	eli Cascading Style Sheets on WWW-sivujen ulkoasun kuvaamiseen käytetty kieli.
HTML	(HyperText Markup Language) on WWW-sivujen sisältöä ja rakennetta kuvaava kieli.
Java	on Sun Microsystemsin kehittämä laitteistoriippumaton olio-ohjelmointikieli.
Java-pavut	(engl. <i>JavaBeans</i>) ovat Java-ohjelmointikielellä toteutettuja uudelleenkäytettäviä komponentteja, joita voidaan kutsua JSP-sivuilla.
JSP	eli Java Server Pages on skriptaustyylinen ohjelmointitekniikka, jossa HTML-koodin sekaan on mahdollista lisätä Java-kielellä kirjoitettua koodia.
Relaatiotietokanta	on tietokanta, jossa tiedot esitetään tauluina (engl. <i>table</i>) ja niitä yhdistävinä suhteina.
Servletti	on palvelimella sijaitseva sovelma (engl. <i>applet</i>), joka toteuttaa HTTP-palvelimen pyynnöstä tietyn toiminnon.

SQL	eli Structured Query Language on tietokantojen hallintaan kehitetty standardoitu kieli.
Tietokanta	on kokoelma loogisesti yhteenliittyvää tietoa.
XML	on rakenteinen dokumenttien määrittely- ja kuvauskieli.

Sovelluksen kehitysympäristöön ja työkaluihin liittyviä termejä ovat seuraavat:

Apache	on vapaan lähdekoodin HTTP-palvelinohjelmisto.
CVS	(Concurrent Version System) on ohjelmistokehityksessä käytettävä versionhallintajärjestelmä.
HTTP	(HyperText Transfer Protocol) on WWW-tekniikassa käytettävä tiedonsiirtoprotokolla, jolla selain pyytää haluttuja sivuja WWW-palvelimelta.
JAAS	(Java Authentication and Authorization Service) on sovellusten käyttäjien autentikointiin ja käyttöoikeuksien hallintaan käytetty tekniikka.
JBoss	on vapaan lähdekoodin sovelluspalvelin. Se tarjoaa mm. autentikointi-, säikeistys- ja tietokantapalveluja.
JDBC	eli Java Database Connectivity on Java-tekniikan käyttämä tietosilta erilaisiin tietokantoihin.
J2EE	(Java 2 Enterprise Edition) on standardi, joka mahdollistaa palvelinsovellusten tuottamisen niin, että sovelluksia voi ajaa millä tahansa J2EE-palvelimella.
Palvelin	on WWW-sovellusten tapauksessa ohjelmisto, joka palvelee asiakkaana toimivien selainten hakupyynnöitä.
PostgreSQL	on vapaa relaatiotietokannan hallintajärjestelmä.
Protokolla	eli yhteiskäytäntö on sovittu liikennöintitapa, jota noudattaen laitteet tai ohjelmat voivat vaihtaa tietoa keskenään. Protokolla määrittelee siirrettävän tiedon esitystavan sekä mekanismin, jolla tieto siirretään.
Skriptaus	tarkoittaa WWW-sivujen tapauksessa ohjelmakoodin kirjoittamista HTML-dokumenttien sisään. Palvelinpuolen

skriptauksessa koodi ajetaan palvelinkoneessa ja asiakaspään skriptauksessa selaimessa.

Selain on ohjelma, joka käyttäjän koneella tulkkaa HTML-kieliset sivut kuvaruudulla esitettävään muotoon.

SSL (Secure Sockets Layer) on protokolla, jonka yli voidaan muodostaa yhteyksiä muilla protokollilla, kuten esimerkiksi HTTP:llä. Yhteys on salattu.

Tomcat on vapaan lähdekoodin servletti- ja JSP-moottori.

Versionhallintajärjestelmä on järjestelmä, johon voidaan mm. tallentaa tiedostoja siten, että tiedostojen kaikki versiot ovat palautettavissa.

2.3 SyncML-tekniikkaan liittyvät termit

Toteutustekniikkaan liittyviä termejä ovat seuraavat:

Avaintenkartoitus (engl. *ID mapping*) kertoo, mitkä asiakkaan tunnisteet (LUID) ja palvelimen tunnisteet (GUID) vastaavat toisiaan. Tehtävän tekee Sync4j-palvelin.

GUID (*Global Unique Identifier*) on tietueen yksilöivä numero SyncML-palvelimella.

Henkilökohtainen kalenteri on pöytäkoneessa, matkapuhelimessa, taskumikrossa tai verkossa toimiva sähköinen kalenteri.

Hidas synkronointi (engl. *slow synchronization*) on kaksisuuntaisen synkronoinnin erikoismuoto, jossa kaikki asiakkaan tietueet lähetetään palvelimelle. Palvelin vertaa jokaista tietoalkiota omiinsa ja tekee näille analyysin.

IMEI (*International Mobile Equipment Identifier*) on mobiililaitteen yksilöivä 15-osainen numero.

Kaksisuuntainen synkronointi (engl. *two-way synchronization*) on yleisin synkronointimenetelmä, jossa asiakas ja palvelin lähettävät toisilleen muuttuneet tietoalkionsa.

Konflikti on synkronoinnin virhetilanne, jossa samaa tapahtumaa on muokattu synkronoitavissa sovelluksissa.

Ryhmätapahtumaristiriita	on ristiriitatilanne, joka voi syntyä synkronoitaessa henkilökohtaista kalenteria ja Korpin kalenteria ryhmätapahtumien osalta. Esimerkiksi luennoitsija muuttaa luentoaikaa henkilökohtaisessa kalenterissaan ja synkronoi kalenterin Korpin kalenterin kanssa. Tällöin on varmistettava, muutetaanko luentoajat myös kaikkien muiden luennolle osallistuvien henkilöiden kalentereihin.
LUID	(<i>Locally Unique Identifier</i>) on tietueen yksilöivä numero asiakaslaitteessa.
MD5	(Message digest 5) on funktio, jota käytetään tiedon salaukseen. Se laskee annetusta syötteestä vasteen. Alkuperäisen syötteen palauttaminen vasteesta on mahdotonta.
SyncML	(Synchronization Markup Language) on synkronointiprotokolla henkilökohtaisen tiedon synkronointiin.
Synkronointi	(engl. <i>synchronization</i>) tarkoittaa saman tiedon pitämistä ajantasaisena kahden tai useamman tietokannan välillä.
Synkronointiankkuri	(engl. <i>synchronization anchor</i>) on aikaleima tai merkkijono, joka kuvaa synkronointitapahtumaa. Synkronointiosapuolet tallentavat toistensa synkronointiankkurit, ja siten voivat todeta toisissaan mahdollisesti tapahtuneet häiriötilat. SyncML määrittelee kaksi synkronointiankkuria: <code>Next</code> ja <code>Last</code> .
Sync4j	on Java-ohjelmointikielellä toteutettu vapaan lähdekoodin SyncML-palvelinohjelmisto.
Työpöytäsynkronointi	(engl. <i>desktop synchronization</i>) on työpöytäsovelluksen ja mobiililaitteen välillä tapahtuva synkronointi.
URI	(<i>Uniform Resource Identifier</i>) on universaali tapa yksilöidä resursseja.

- vCalendar** on laitteisto- ja tiedonsiirtoriippumaton tiedostoformaatti kalenteritiedon tallentamiseen ja välittämiseen.
- Virkistyssynkronointi** (engl. *refresh synchronization*) on synkronointimenetelmä, jossa lähettäjä lähettää kaikki tietoalkionsa vastaanottajalle, joka korvaa omat tietonsa lähettäjän tiedoilla. Menetelmä toimii molempiin suuntiin, eli joko asiakkaalta palvelimelle, tai palvelimelta asiakkaalle.
- Yksisuuntainen synkronointi** (engl. *one-way synchronization*) on synkronointimenetelmä, jossa muuttuneet tiedot lähetetään yhteen suuntaan, joko asiakkaalta palvelimelle tai palvelimelta asiakkaalle. Vastaanottaja ei lähetä omia muuttuneita tietojaan.

3 JÄRJESTELMÄN ARKKITEHTUURI, TOIMINTA JA TYÖKALUT

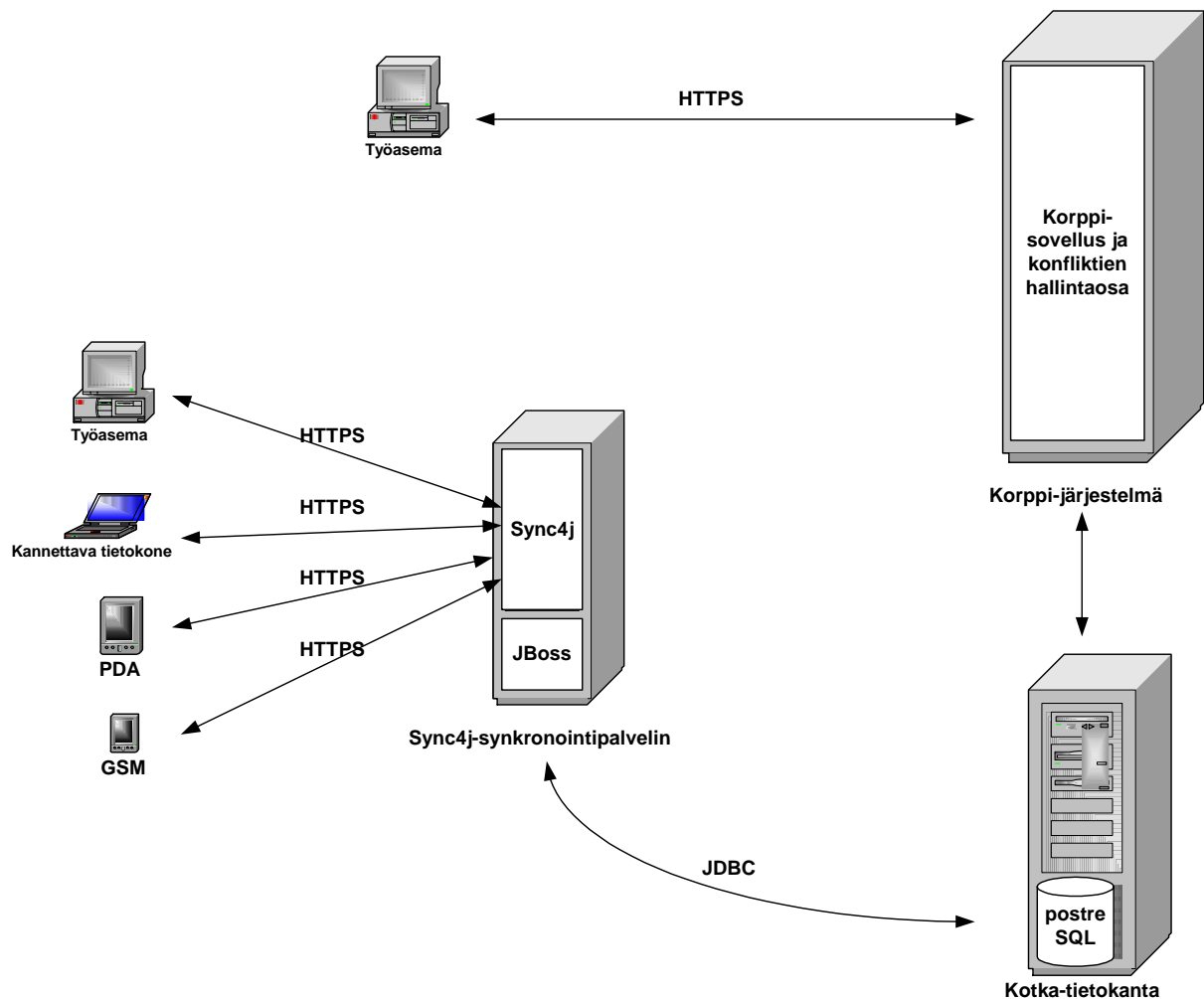
Kyyhky-projektin toteuttaa päivyrirajapinta Jyväskylän yliopiston tietotekniikan laitoksella toteutettuun Korppi-nimiseen järjestelmään. Rajapinnan avulla pystytään synkronoimaan tapahtumia Korppi-järjestelmän kalenteriosion ja erilaisten mikrojen, matkapuhelimien ja muiden kämmenlaitteiden kalenterien kesken. Tarkemmin toiminnot on esitetty Vaatimusmäärittely-dokumentissa.

Luvussa 3.1 esitellään järjestelmän arkkitehtuuria. Luvussa 3.2 kuvataan järjestelmässä käytettyjä työkaluja. Luvussa 3.3 tarkastellaan lyhyesti JSP-tekniikkaa.

3.1 Toteutettavan järjestelmän arkkitehtuuri

Toteutettavan järjestelmän arkkitehtuuri on kuvan 1 kaltainen. Synkronointipalvelu toteutetaan Sync4j-synkronointipalvelimella. Synkronointipalvelin toimii JBoss-sovelluspalvelimen version 4.0.1 päällä ja se hyödyntää tämän palveluita, kuten JAAS-autentikointia, säikeistystä ja tietokantayhteyksiä. Sync4j-palvelimesta on tosin tullut uusi versio 2.0, joka toimii ilman J2EE-sovelluspalvelinta. Kyyhky-projektissa käytetään kuitenkin JBoss-palvelimen kanssa toimivaa Sync4j-palvelinta.

Sovelluspalvelimella on suora yhteys Kotka-tietokantaan. Konfliktienhallintaosa toteutetaan Korppi-järjestelmän käyttöliittymäosaan.



Kuva 1. Synkronointijärjestelmän rakenne.

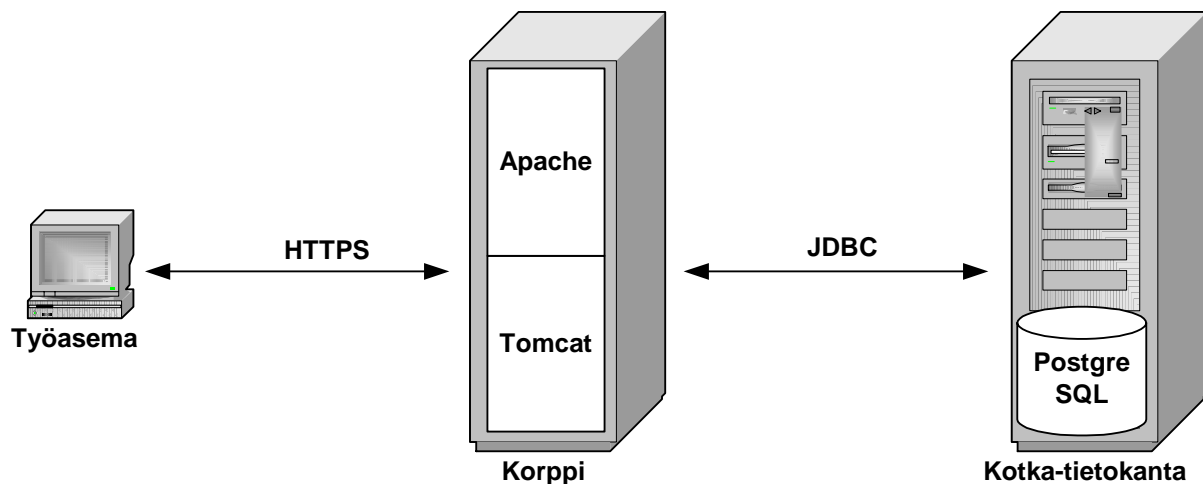
3.2 Työkalut ja kehitysympäristöt

Sovelluksen kehitysympäristönä projektissa käytetään Linux Fedora Core 1-levityspakettia ja Microsoft Windows XP Professional:ia. Lisäksi käytössä ovat Sync4j-synkronointipalvelin, JBoss-sovelluspalvelin versio 4.0.1 ja Kotka-tietokanta.

Sovelluskehittimenä on JBuilder 9 Enterprise ja versionhallintajärjestelmänä CVS. Linux-koneisiin on asennettu myös HTTP-palvelimeksi Apache ja tietokantannan hallintajärjestelmäksi PostgreSQL. Käytössä on lisäksi tietokannanhallintaohjelmisto phpPgAdmin. Tarkemmin asiakaslaitteita ja ohjelmistoja kuvataan luvussa 7.

3.3 JSP-tekniikka

Konfliktienhallintaosa toteutetaan Korppi-järjestelmään käyttämällä JSP-tekniikkaa. JSP-tekniikka perustuu Javan Servlet-rajapintaan. Tekniikka mahdollistaa dynaamisen sisällön luonnin WWW-sivulle. Dynaamisuus tarkoittaa, että HTML-dokumentit luodaan palvelimella dynaamisesti käyttäjän toiminnon ja tietokannasta haetun datan perusteella, ennen dokumentin lähettämistä asiakasohjelmistolle.



Kuva 2. Korppi-järjestelmän rakenne.

Korppi-järjestelmässä Apache-palvelin ottaa vastaan käyttäjän selaimelta tulevan sivupyynnön. Se päättää, onko kyseessä staattinen HTML-sivu vai JSP-sivu. Staattiset sivut Apache palauttaa suoraan takaisin selaimen. JSP-pyyntö välittyy Tomcat-servlettimoottorille. Saatuaan pyynnön Tomcat tutkii, onko vastaavaa Korppi-järjestelmän servlettiä olemassa. Olemassaoleva servletti suoritetaan ja palautetaan selaimelle.

Java-pavut ovat Javalla toteutettuja luokkia. Java-koodi keskitetään papuihin siksi, että JSP-sivujen ylläpidettävyys pysyy hyvänä. Kyyhky-ryhmän on tarkoitus toteuttaa yleiskäyttöisiä Java-papuja, joita olisi helppo muiden kehittäjien käyttää ja ylläpitää. Ryhmä toteuttaa konfliktienhallintaosaan useita papuja, jotka hallitsevat mm. konfliktien havaitsemista ja virheilmoitusten tulostamista.

4 SYNC4J-PALVELIN

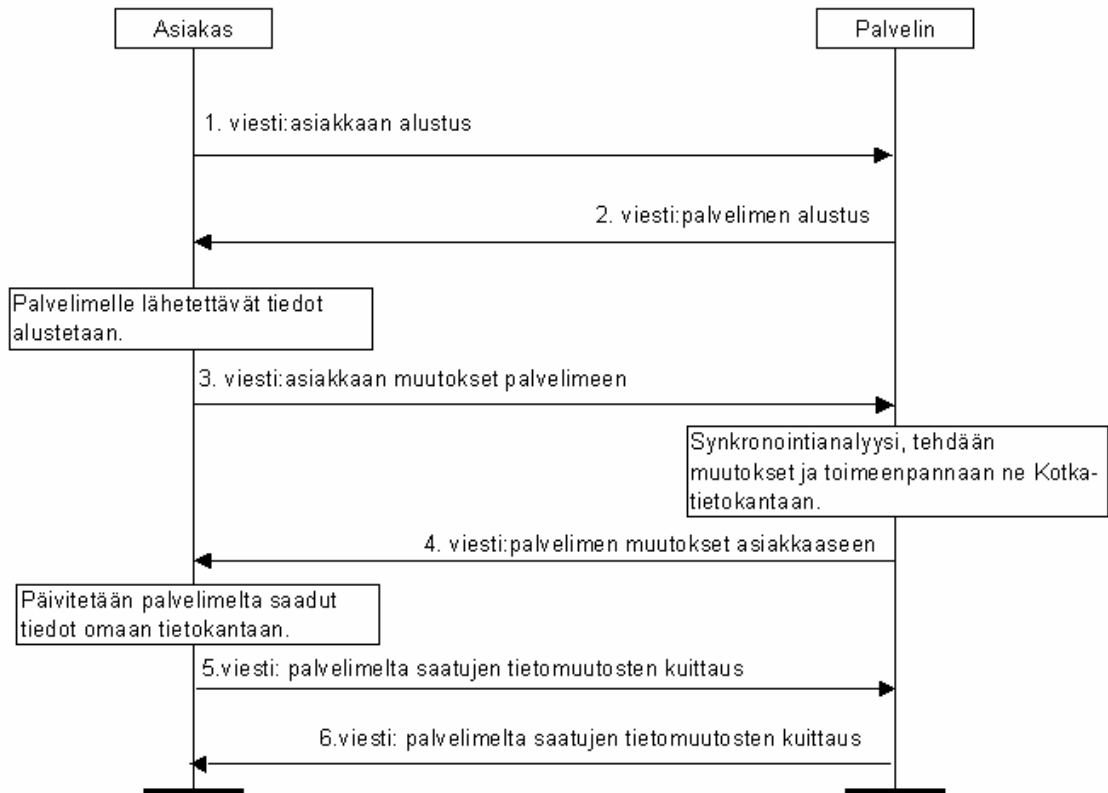
Sync4j-ohjelmiston Sync4j-synkronointipalvelin on Javalla toteutettu J2EE-sovelluspalvelimen päällä toimiva tuote. Koska Sync4j-synkronointipalvelin on tehty noudattamaan J2EE:n periaatteita, voidaan palvelin sijoittaa toimimaan mille tahansa J2EE-sovelluspalvelimelle.

Projektissa käytetään JBoss-sovelluspalvelinta. Sync4j-synkronointipalvelin pyörii siis samalla koneella kuin JBoss ja käyttää sen palveluita (esimerkiksi todennus ja tietokantayhteydet) toimiakseen.

4.1 Kaksisuuntainen synkronointi

Esitellään lyhyesti yleisimmin synkronoinnissa käytetty synkronointitapa, kaksisuuntainen synkronointi. Sync4j-synkronointipalvelimen tukemia muita synkronointitapoja ovat hidas synkronointi ja yksisuuntainen synkronointi molempiin suuntiin. Erilaisia synkronointitapauksia kuvataan tarkemmin Vaatimusmäärittelyssä.

Käyttäjä ottaa HTTPS-yhteyden henkilökohtaisen kalenterinsa SyncML:ää tukevalla asiakasohjelmistolla Sync4j-synkronointipalvelimelle. Synkronointi-istunto etenee kuvan 3 mukaisesti.



Kuva 3. Kaksisuuntainen synkronointi.

4.2 Synkronointi Sync4j-synkronointipalvelimella

Sync4j-synkronointipalvelin keskusteleee asiakasohjelmiston tai -laitteen kanssa HTTP- tai HTTPS-protokollan yli käyttäen SyncML-viestejä protokollan [8] mukaisesti.

Käyttäjää todennetaan JBossin JAAS-autentikointipalveluita hyödyntäen. Mikäli käyttäjän todennus onnistuu, vaihdetaan sanomia protokollan mukaisesti (katso kuva 3 sivulla 11).

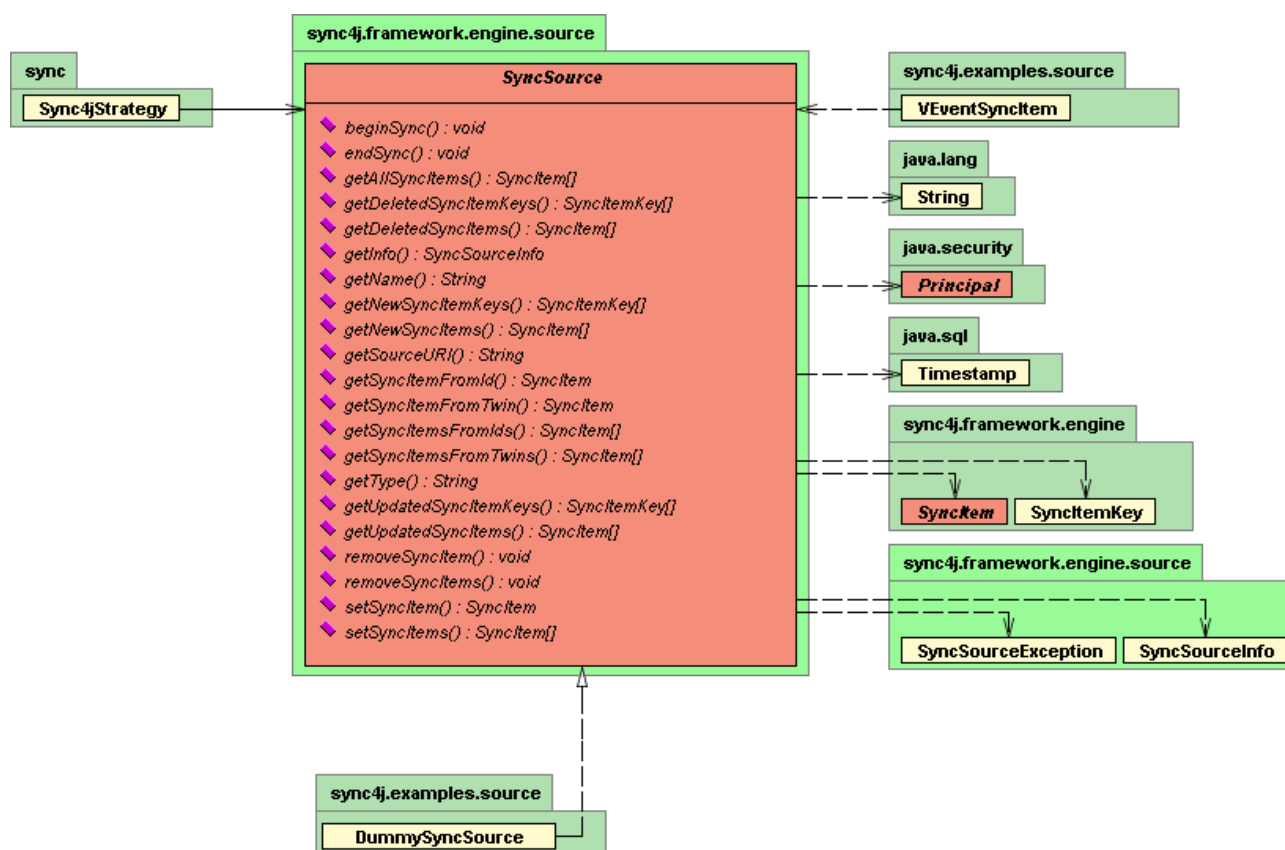
Palvelin luo asiakkaan viestissä päivitettävien tapahtumien data-elementtien sisällöistä SyncItem-olioita. Koska kyseessä on vCalendar-muotoista tietoa, parsitaan sisältö edelleen VEvent-olioon. Palvelin vertaa asiakkaan uusia, muokattuja ja poistettuja tapahtumia Korpin tapahtumiin ja päättää SyncStrategy-rajapinnan toteuttavan säännösten (vrt. synkronointitaulukko Vaatimusmäärittelyssä) mukaan, miten kuhunkin päivitykseen suhtaudutaan. Strategia toteutetaan käyttäen lähteessä [2] esiteltyä strategia-

suunnittelumallia. Mahdolliset muutokset Kotka tietokantaan suoritetaan sekä protokollan mukaisesti lähetetään palvelinpään muutokset asiakkaalle.

4.3 Sync4j-synkronointipalvelimen Korppi-moduuli

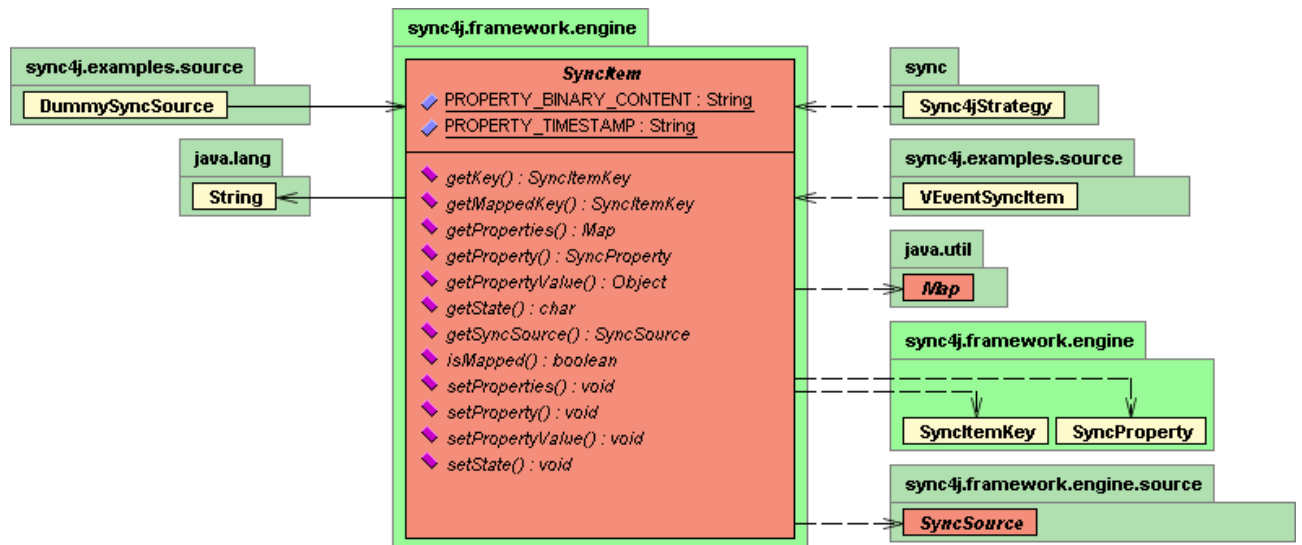
Sync4j-synkronointipalvelimessa on mukana yksinkertainen tiedostoista lukeva ja kirjoittava synkronointilähde (`FileSystemSyncSource`). Korpin kalenterin synkronointiin sitä kuitenkin ei voi soveltaa, joten projekti kehittää soveltuvan, Kotka-tietokantaa ymmärtävän, `SyncSource`-rajapinnan toteuttavan synkronointilähteen. Myös asiakkaan lähettämän vCalendar-tiedon jäsentämiseen projekti kehittää tarvittavat luokat.

Kuvassa 4 esitetään synkronointilähteen rajapinta ja siihen liittyvät luokat ja rajapinnat.



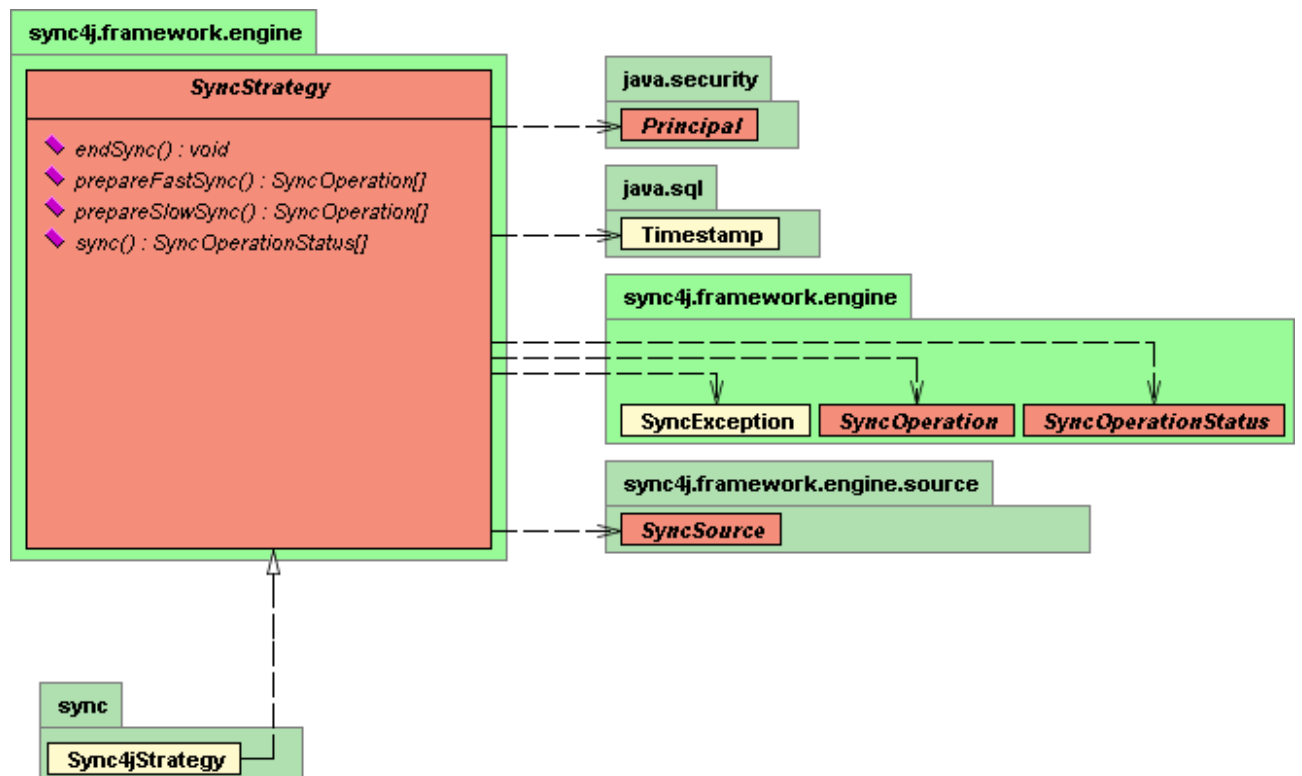
Kuva 4. SyncSource-rajapintamäärittely synkronointilähteelle.

Kuvassa 5 esitetään rajapintamäärittys synkronoitavalle tietue oliolle.



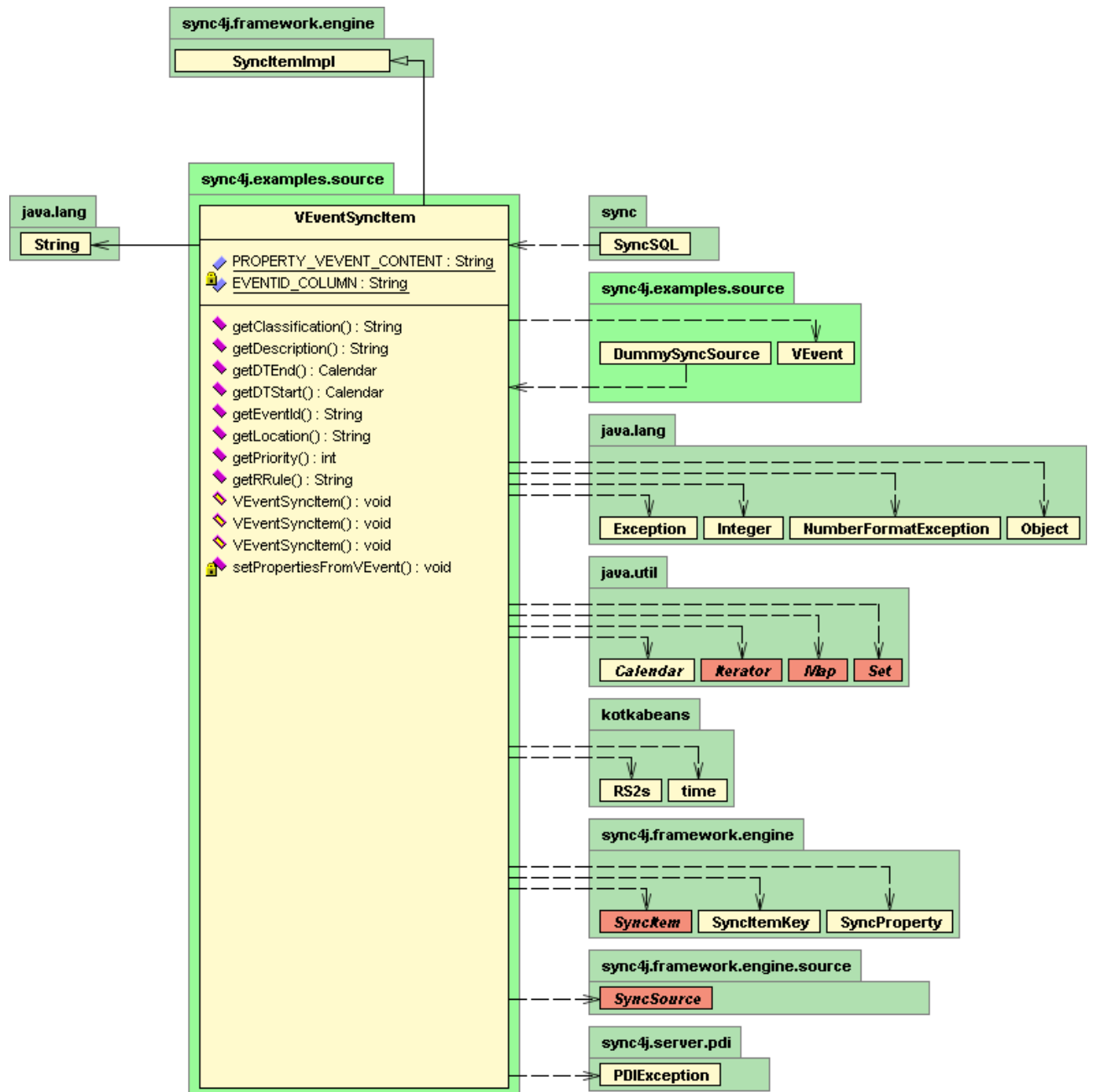
Kuva 5. SyncItem-rajapintamäärittys.

Kuvassa 6 esitetään synkronointistrategiarajapinta.



Kuva 6. SyncStrategy-rajapinta.

Kuvassa 7 esitetään suunnitelma vCalendar-formaatin toteuttavasta vEventSyncItem-luokasta.



Kuva 7. vEventSyncItem-luokka.

5 KÄYTTÖLIITTYMÄ

Järjestelmän pääasiallisen käyttöliittymän muodostaa henkilökohtaisen kalenterin oma käyttöliittymä, josta synkronointi käynnistetään ja pääosin hallitaan. Käyttäjään synkronointia käyttäjän laite tulee pystyä tunnistamaan.

Järjestelmässä voi tulla kuitenkin konflikteja ja ryhmätapahtumaristiriitoja Korpin kalenterin ja henkilökohtaisen kalenterin välillä. Näitä tilanteita on pystyttävä jotenkin hallitsemaan, joten Korpin käyttöliittymäosaan projektissa toteutetaan konfliktienhallintaosa. Sillä käyttäjä pystyy ratkaisemaan konfliktitapaukset ja ryhmätapahtumaristiriidat sekä muokkaamaan Korpin kalenterin tunnistetietoja, kuten kalenterin nimeä ja kuvausta. Konfliktienhallintaosa toteutetaan JSP-tekniikalla.

Korppi-järjestelmään toteutettavat uudet sivut noudattavat Korppi-järjestelmän nykyisiä ratkaisuja. Synkronointisovelluksen tarvitsemat linkit tulevat näkyviin Korppi-järjestelmän navigointipuussa.

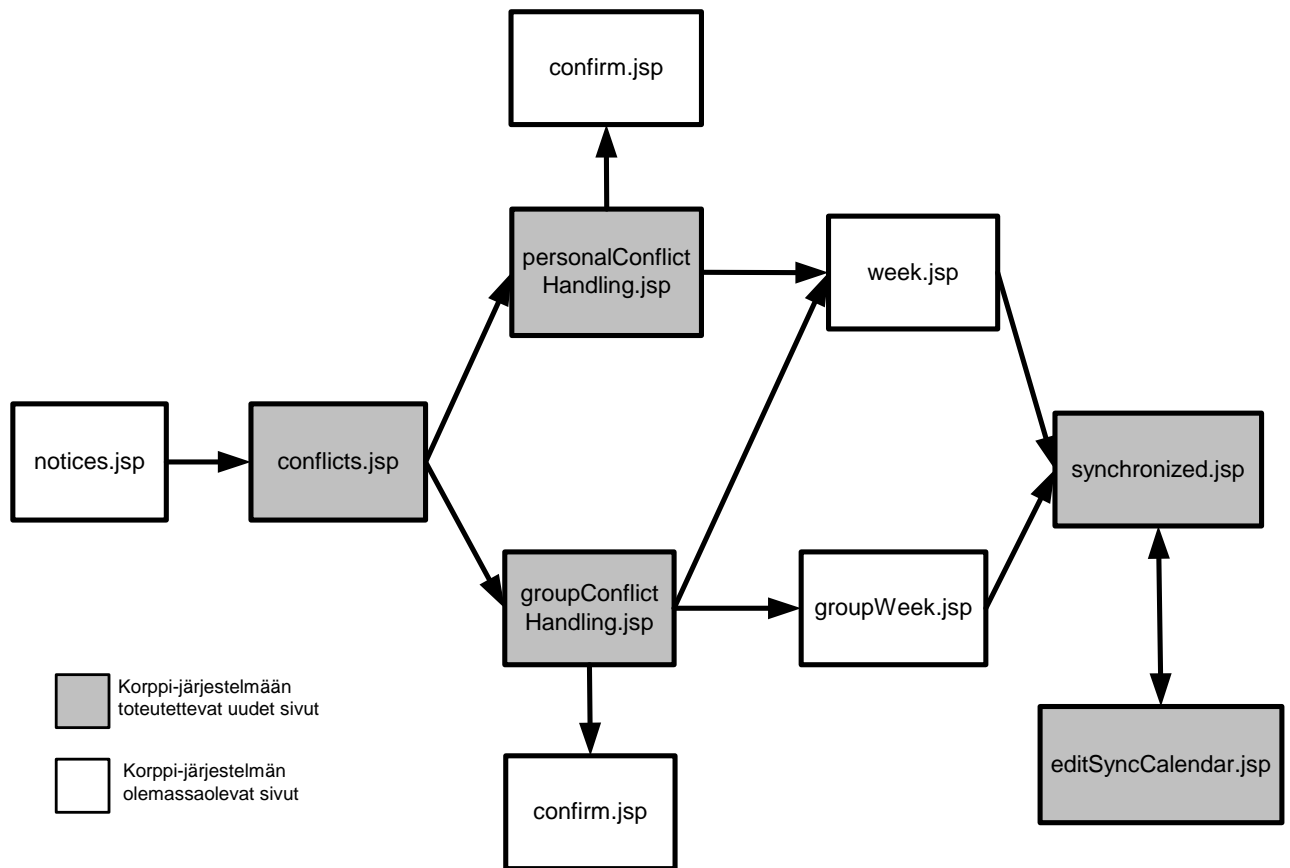
5.1 Käyttöliittymän JSP-sivujen rakenne

Käyttöliittymän JSP-sivujen rakenne on kuvan 8 mukainen. Käyttäjän konfliktit näytetään osana Korppi-järjestelmän nykyistä ilmoitustaulua sivulla `notices.jsp`. Sivulta käyttäjä voi siirtyä sivulle `conflict.jsp`, jossa esitetään kaikki konfliktit Korpin kalenterin ja henkilökohtaisen kalenterin välillä sekä ryhmätapahtumaristiriidat Korpin ryhmäkalenterin ja henkilökohtaisen kalenterin välillä. Sivulta käyttäjä voi siirtyä käsittelemään kutakin konfliktia tai ryhmätapahtumaristiriitaa erikseen. Henkilökohtaisen kalenterin konflikti ratkaistaan sivulla `personalConflictHandling.jsp` ja ryhmätapahtuman ristiriita sivulla `groupConflictHandling.jsp`.

Voimaan jäävän tapahtuman valinta varmistetaan kummassakin tapauksessa sivulla `confirm.jsp`. Sivulla `personalConflictHandling.jsp` on myös linkki kyseisen henkilön Korpin viikkokalenteriin konfliktitapahtuman kohdalle ja sivulla `groupConflictHandling.jsp` on linkki Korpin ryhmän viikkokalenteriin

ryhmätapahtumaristiriidan kohdalle. Kalenterisivuilta `week.jsp` ja `groupweek.jsp` päästään sivulle `synhronized.jsp`, jossa voidaan valita muokattava henkilökohtainen kalenteri. Sivulla `editSyncCalendar.jsp` voidaan muokata henkilökohtaisen kalenterin tietoja.

Projektissa toteutetaan uusia JSP-sivuja ja tehdään muutoksia jo olemassa oleviin. Uusia sivuja ovat `conflict.jsp`, `personalConflictHandling.jsp`, `groupConflictHandling.jsp`, `synhronized.jsp` ja `editSyncCalendar.jsp`. Olemassa olevia sivuja ovat `notices.jsp`, `confirm.jsp`, `week.jsp` ja `groupweek.jsp`.



Kuva 8. Konfliktienhallintaosan rakenne.

5.2 Konflikti-ilmoitukset

Kun käyttäjällä on konflikteja tai ryhmätapahtumien ristiriitoja henkilökohtaisen ja Korppi-kalenterin kanssa, ilmoitetaan tästä sekä kuvan 9 ilmoitustaulussa että kuvan 10 kalenterisivulla.

The screenshot shows the Korppi system interface. At the top right, it identifies the user as Timo Aarniovuori, a student at the Department of Information Technology, with the date 17.3.2004. The main header contains navigation icons for 'EXIT', '@', a hand, and a question mark, labeled 'Poistu', 'Palaute', 'Asetukset', and 'Opastus' respectively.

On the left, there is a calendar for March 2004, with the 17th highlighted. Below the calendar are links for 'Opiskelusi', 'Kalenteri', 'Opintojen suunnittelu', 'Henkilötiedot', 'Ryhmät', 'Lue viestit', 'Jätä viesti', 'Kyselyt', and 'Salivaraukset'. There is also a note about switching between monthly and weekly calendars and a 'Testikone!' section with a menu time of 20 ms and the date 2004.03.17.13.18.

The main content area is titled 'Ilmoitustaulu' (Notification Board). A prominent red-bordered box contains the warning: 'KALENTERISSASI ON KONFLIKTEJA! [[Konfliktien käsittely](#)]'. Below this, there are two sections of messages:

Korppi-järjestelmän yleiset viestit:

Poista	Lähetetty	Lähetittäjä	Viesti	Url
<input type="checkbox"/>	28.11.2003 14:19	Sakari Saukkonen	Sinua on pyydetty liittymään ryhmään vastaajat.	Tarkemmat tiedot ryhmästä.
<input type="checkbox"/>	9.6.2003 21:56	Minna Elina Pesonen	Sinut on poistettu kurssilta XRU101 Ruotsin peruskurssi - akademisk svenska.	Kurssisi Korpissa
<input type="checkbox"/>	30.4.2003 12:29	Sami Petteri Kollanus	Sinut on lisätty opiskelijaksi kurssille TJTC25 Ohjelmistotekniikka.	Kurssi Korpissa

Henkilökohtaiset viestisi:

Poista	Lähetetty	Lähetittäjä	Viesti	Url
<input type="checkbox"/>	16.3.2004 11:03	Timo Tapani Aarniovuori	Koe!	-- [Vastaa]
<input type="checkbox"/>	4.2.2004 13:41	Jukka-Pekka Santanen	Sinut on lisätty kurssin TIE280 Sovellusprojekti opetusryhmään.	Kurssi Korpissa [Vastaa]
<input type="checkbox"/>	4.2.2004 12:55	Jaakko Paavo Huvärinen	Sinua on pyydetty liittymään ryhmään Korppi svnm1	Tarkemmat tiedot ryhmästä [Vastaa]

Kuva 9. Ilmoitustaulu sivu `notices.jsp` konfliktitilanteessa.

<Maaliskuu 2004>

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

KALENTERISSASI ON KONFLIKTEJA! [[Konfliktien käsittely](#)]

Viikonäkymä, Timo Aarniovuori
Viikko 12

Hyppää päivämäärään:

[viikko 10](#) [viikko 11](#) **viikko 12** [viikko 13](#) [viikko 14](#)

Opiskelusi

Kalenteri

[Päivä](#)

[Viikko](#)

[Kuukausi](#)

[Vuosi](#)

[Lisää tapahtuma](#)

[Ryhmän tapahtumat](#)

[Ohjaukset](#)

[Synkronointi](#)

[Henkilötiedot](#)

[Ryhmät](#)

[Ilmoitustaulu](#)

[Kyselyt](#)

[Asetuksista voit vaihtaa kuukausikalenterin paikkaa.](#)

[Korppi in English](#)

v2004.02.27

Kello	Ma 15.3.	Ti 16.3.	Ke 17.3.	To 18.3.	Pe 19.3.	La 20.3.	Su 21.3.
8:00	[+]	[+]	[+]	[+] 08:00 - 10:00 Tietoliikenneteoria ---	[+]	[+]	[+]
10:00	[+] 10:00 - 12:00 Tietoliikenneteoria ---	[+]	[+] 10:15 - 12:00 Kyyhky-palaveri ---	[+]	[+]	[+]	[+]

Kuva 10. Kalenterisivu week .jsp konfliktitilanteessa.

5.3 Konfliktien käsittely

Sivulla *Ilmoitustaulu* (Katso kuva 9) linkistä *Konfliktien käsittely* päästään käsittelemään konflikteja ja ryhmätapahtumien ristiriitoja. Tapaukset listataan yksitellen kuvan 11 sivulla.

Korppi EXIT @ Asetukset Opastus Poistu Palaute

Timo Aarniovuori
Tietotekniikan laitos
[opiskelija]
ke 17.3.2004

<Maaliskuu 2004>
Ma Ti Ke To Pe La Su
10: 1 2 3 4 5 6 7
11: 8 9 10 11 12 13 14
12: 15 16 17 18 19 20 21
13: 22 23 24 25 26 27 28
14: 29 30 31

Opiskelusi
Kalenteri
Opintojen suunnittelu
Henkilötiedot
Ryhmät
Lue viestit
Jätä viesti
Kyselyt
Salivaraukset

Asetuksista voit vaihtaa
kuukausikalenterin paikkaa.

Korppi in English

Testikone!
menu time: 20 ms
v2004.03.17.13.18

Konfliktien käsittely

Kalenterin henkilökohtaisten tapahtumien konfliktit

Valitse tapahtuma, jota haluat käsitellä.

Kalenteri	Päivämäärä	Aika	Nimi	Käsittely
Nokia 6800	28.11.2003	14:00 - 16:00	Uimahalli	[Konfliktin käsittely]
Korppi	9.6.2003	18:00 - 20:00	Uimahalli	

Kalenteri	Päivämäärä	Aika	Nimi	Käsittely
Nokia 6800	28.11.2003	10:00 - 12:00	Luento	[Konfliktin käsittely]
Korppi	9.6.2003	14:00 - 16:00	Luento	

Kalenterin ryhmätapahtumien ristiriidat

Valitse tapahtuma, jota haluat käsitellä.

Kalenteri	Päivämäärä	Aika	Nimi	Käsittely
Nokia 6800	28.11.2003	14:00 - 16:00	Palaveri	[Ristiriidan käsittely]
Korppi	9.6.2003	18:00 - 20:00	Palaveri	

Sivun tekeminen kesti 361 ms

DB.load: diff avg: 30, 25 max diff: 30, 40 ms, nr: 0 0, DB.getLoadStatus(): 0, DB.hasFreeTime(): true

Kuva 11. Sivun conflicts.jsp konfliktien listaamiseksi.

Sivulta päästään ratkaisemaan konfliktit ja ryhmätapahtumine ristiriidat yksitellen. *Konfliktin käsittely*-linkistä avautuu kuvan 12 sivu.

Korppi EXIT @ Asetukset Opastus Poistu Palaute

Timo Aarniovuori
Tietotekniikan laitos
[opiskelija]
ke 17.3.2004

<Maaliskuu 2004>
Ma Ti Ke To Pe La Su
10: 1 2 3 4 5 6 7
11: 8 9 10 11 12 13 14
12: 15 16 17 18 19 20 21
13: 22 23 24 25 26 27 28
14: 29 30 31

Opiskelusi
Kalenteri
Opintojen suunnittelu
Henkilötiedot
Ryhmät
Lue viestit
Jätä viesti
Kyselyt
Salivaraukset

Asetuksista voit vaihtaa
kuukausikalenterin paikkaa.

Korppi in English

Testikone!
menu time: 20 ms
v2004.03.17.13.18

Konfliktin käsittely

Kalenterin henkilökohtaisten tapahtumien konfliktit

Valitse kalenteritapahtuma, jonka haluat jäädän voimaan.

Valinta	Kalenteri	Päivämäärä	Aika	Nimi	Linkki kalenterinäkömään
<input type="radio"/>	Nokia 6800	28.11.2003	14:00 - 16:00	Uimahalli	[Henkilökohtainen kalenteri]
<input type="radio"/>	Korppi	9.6.2003	18:00 - 20:00	Uimahalli	[Henkilökohtainen kalenteri]

Hyväksy Peruuta

Sivun tekeminen kesti 361 ms

DB.load: diff avg: 30, 25 max diff: 30, 40 ms, nr: 0 0, DB.getLoadStatus(): 0, DB.hasFreeTime(): true

Kuva 12. Sivun personalConflictHandling.jsp konfliktin ratkaisemiseksi.

Sivulla voidaan valita kalenteritapahtuma, jonka halutaan jäävän voimaan. Valinta tehdään radiopainikkeella ja hyväksytään *Hyväksy*-painikkeesta. Toiminto voidaan myös perua *Peruuta*-painikkeesta, jolloin palataan kuvan 11 konfliktien listaussivulle. Sivulta pääsee myös kuvan 10 kalenterinäkömään, jossa konfliktitapahtuma näytetään. Tapahtuman valinta varmistetaan käyttäjältä kuvan 13 varmistussivulla.

Korppi Poistu Palaute Asetukset Opastus

Timo Aarniovuori
Tietotekniikan laito:
[opiskelija]
ke 17.3.2004

<Maaliskuu 2004>
Ma Ti Ke To Pe La Su
10: 1 2 3 4 5 6 7
11: 8 9 10 11 12 13 14
12: 15 16 17 18 19 20 21
13: 22 23 24 25 26 27 28
14: 29 30 31

Varmistus
Hyväksytkö valintasi: Korppi 28.11.2003 14:00 - 16:00 Palaveri?
HUOM! Tapahtumat joita et valinnut poistetaan kalenteristasi pysyvästi!
Hyväksy Peruuta
Sivun tekeminen kesti 361 ms
DB.load: diff avg: 30, 25 max diff: 30, 40 ms, nr: 0 0, DB.getLoadStatus(): 0, DB.hasFreeTime(): true

Opiskelusi
Kalenteri
Opintojen suunnittelu
Henkilötiedot
Ryhmät
Lue viestit
Jätä viesti
Kyselyt
Salivaraukset
Asetuksista voit vaihtaa
kuukausikalenterin paikkaa.
Korppi in English
Testikone!
menu time: 20 ms
v2004.03.17.13.18

Kuva 13. Sivulla `confirm.jsp` varmistetaan tapahtuman hyväksyntä.

5.4 Ryhmätapahtuman ristiriidan käsittely

Konfliktien listaussivulta (katso kuva 11) valittu ryhmätapahtuman ristiriita johtaa kuvan 14 näkömään. Kummastakin ristiriitaisesta tapahtumasta on linkki sekä ryhmäkalenteriin että Korpin henkilökohtaiseen kalenteriin.




Poistu


Palaute


Asetukset


Opastus

Timo Aarniovuori
Tietotekniikan laitos
[opiskelij
ke 17.3.2004

<Maaliskuu 2004>

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

Ristiriidan käsittely

Kalenterin ryhmätapahtumien ristiriidat

Valitse kalenteritapahtuma, jonka haluat jäävän voimaan.

Valinta	Kalenteri	Päivämäärä	Aika	Nimi	Linkki kalenterinäkymiin
<input type="radio"/>	Korppi	28.11.2003	14:00 - 16:00	Palaveri	Henkilökohtainen kalenteri Ryhmäkalenteri
<input type="radio"/>	Nokia 6800	9.6.2003	18:00 - 20:00	Palaveri	Henkilökohtainen kalenteri Ryhmäkalenteri

Sivun tekeminen kesti 361 ms

DB.load: diff avg: 30, 25 max diff: 30, 40 ms, nr: 0 0, DB.getLoadStatus(): 0, DB.hasFreeTime(): true

[Opiskelusi](#)
[Kalenteri](#)
[Opintojen suunnittelu](#)
[Henkilötiedot](#)
[Ryhmät](#)
[Lue viestit](#)
[Jätä viesti](#)
[Kyselyt](#)
[Salivaraukset](#)

[Asetuksista voit vaihtaa kuukausikalenterin paikkaa.](#)


[Korppi in English](#)


Testikone!
menu time: 20 ms


v2004.03.17.13.18

Kuva 14. Sivun `groupConflictHandling.jsp` ryhmätapahtuman ristiriitatilanteen ratkaisemiseksi.


Jos valitaan ryhmätapahtuma, joka muuttuu myös muille ryhmän jäsenille, varmistetaan se käyttäjältä kuvan 15 sivulla. Painettaessa *Kyllä*-painiketta, tapahtuma välittyy koko ryhmälle.




Poistu


Palaute


Asetukset


Opastus

Timo Aarniovuori
Tietotekniikan laitos
[opiskelija]
ke 17.3.2004

<Maaliskuu 2004>

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

Varmistus ryhmätapahtumalle

Hyväksytkö valintasi: Nokia 6800 9.6.2003 18:00 - 20:00 Palaveri

HUOM! Tapahtuman muutokset välitetään koko ryhmälle?

Ryhmä: Kyyhky

Sivun tekeminen kesti 361 ms

DB.load: diff avg: 30, 25 max diff: 30, 40 ms, nr: 0 0, DB.getLoadStatus(): 0, DB.hasFreeTime(): true

[Opiskelusi](#)
[Kalenteri](#)
[Opintojen suunnittelu](#)
[Henkilötiedot](#)
[Ryhmät](#)
[Lue viestit](#)
[Jätä viesti](#)
[Kyselyt](#)
[Salivaraukset](#)

[Asetuksista voit vaihtaa kuukausikalenterin paikkaa.](#)

[Korppi in English](#)

Testikone!
menu time: 20 ms

v2004.03.17.13.18

Kuva 15. Sivulla `confirm.jsp` varmistetaan ryhmätapahtuma ristiriidan käsittely.

5.5 Kalenterin tunnistetietojen muokkaus

Konfliktienhallintaosaan toteutetaan myös kuvan 16 synkronointisivu, jossa voidaan antaa synkronoitavalle kalenterille nimi ja lisätietoja sekä muokata näitä tietoja.

Korppi

Poistu Palautte Asetukset Opastus

Timo Aarniovuori
Tietotekniikan laitos
[opiskelija]
to 18.3.2004

<Maaliskuu 2004 >
Ma Ti Ke To Pe La Su
10: 1 2 3 4 5 6 7
11: 8 9 10 11 12 13 14
12: 15 16 17 18 19 20 21
13: 22 23 24 25 26 27 28
14: 29 30 31

Synkronointi

Olemassaolevat kalenterit

Kalenterisi jotka voidaan synkronoida Korpin kalenterin kanssa.

Valinta	Tunnustekoodi	Nimi	Lisätietoja
<input checked="" type="radio"/>	123456789012345	Nokia	Oman puhelimen kalenteri
<input type="radio"/>	abcdefg1234	Outlook	Oma outlook

Muokkaa Peruuta

Opiskelusi
Kalenteri
Päivä
Viikko
Kuukausi
Vuosi
[Lisää tapahtuma](#)
[Ryhmän tapahtumat](#)
[Ohiausaajat](#)
Synkronointi
[Henkilötiedot](#)
[Ryhmät](#)
[Ilmoitustaulu](#)
[Kyselyt](#)
Asetuksista voit valita kuukausikalenterin
pakkeen.
[Korppi in English](#)
v2004.02.27

Kuva 16. Sivun synhronized.jsp kalenterin valintaan.

Valittaessa kalenteri ja painettaessa *Muokkaa*-painiketta avautuu kuvan 17 sivu.

Korppi

EXIT Poistu @ Palaute Asetukset Opastus

Timo Aarniovuori
Tietotekniikan laitos
[opiskelija]
to 18.3.2004

<Maaliskuu 2004>

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

Muokkaa kalenterin tietoja

Muokkaa kalenteriasi: Nokia

Tunnistekoodi:

Kalenterin/laitteen nimi:

Lisätietoja:

Tallenna Peruuta

Opiskelusi
Kalenteri
Päivä
Viikko
Kuukausi
Vuosi
Lisää tapahtuma
Ryhmän tapahtumat
Ohjaukset
Synkronointi
Henkilötiedot
Ryhmät
Ilmoitustaulu
Kyselyt
Asetuksista voit vaihtaa
kuukausikalenterin paikkaa.
Korppi in English
v2004.02.27

Kuva 17. Sivun editCalendar.jsp näkymä kalenterin tunnistetietojen muokkaukseen.

Sivulta voi muokata kyseisen kalenterin tietoja ja tallentaa muutokset. *Peruuta*-painikkeesta päästään takaisin kuvan 16 synkronointisivulle.

6 TIETOKANTA

Kotka-tietokanta ei sellaisenaan sovellu Kyyhky-projektissa toteutettavaa sovellusta varten, joten tietokantaan on tehtävä uusia tauluja. Sync4j-palvelin vaatii omat tietokantataulut, jotka on lisättävä Kotka-tietokantaan. On myös luotava taulut palvelimen tietokannan ja Kotka-tietokannan yhteistoimintaa varten. Luvussa on kuvattu tarvittavat taulut sovellukselle.

6.1 Sync4j-palvelimen tarvitsemat taulut

Taulukoissa 1-5 kuvataan Sync4j-palvelimen taulut, jotka palvelin tarvitsee toimiakseen.

Nimi	sync4j_principal
Tarkoitus	Käyttäjällä voi olla useampi synkronoitava henkilökohtainen kalenteri. Kukin taulun rivi kuvaa käyttäjän ja laitteen yhdistelmää. Taulu on siis many-to-many -relaatio sync4j_device- ja Person-taulujen välillä.
Avainkenttä	principalid
Kentät	
device	on viittaus laitteen tunnisteeseen sync4j_device-aulussa.
principalid	on avainkenttä.
personid	on viittaus käyttäjän tunnisteeseen Person-aulussa.
account	on synkronointiin käytetty käyttäjätunnus (tarvitaan vain, jos eri kuin Korpin tunnus).
passwd	on synkronointiin käytetty salasana (tarvitaan vain, jos eri kuin Korpin tunnus).

Taulukko 1. sync4j_principal-taulu.

Nimi	sync4j_device
Tarkoitus	Jokainen taulun rivi kuvaa synkronointiin käytettyä laitetta tai kalenteriohjelmaa pöytäkoneella.
Avainkenttä	deviceid
Kentät	
deviceid	on laitteen tunniste (esimerkiksi IMEI-tunniste).
description	on kuvaus laitteesta.
type	on kalenterin nimi tai tyyppi (esimerkiksi Nokia 6600).

Taulukko 2. sync4j_device-taulu.

Nimi	sync4j_sync_source
Tarkoitus	Jokainen taulun rivi kuvaa synkronoitavan tiedon lähdettä (esimerkiksi Korppi).
Avainkenttä	uri
Kentät	
uri	on synkronointilähteen URI.
config	on synkronointilähteen konfigurointitiedoston nimi polkuineen.

Taulukko 3. sync4j_sync_source-taulu.

Nimi	sync4j_last_sync
Tarkoitus	Taulu kuvaa viimeisimpään synkronointiin liittyvät tiedot jokaiselle laite- ja käyttäjäparille jokaisen synkronointilähteen suhteen.
Avainkentät	principal, sync_source
Kentät	
principal	on viittaus laite- ja käyttäjäparin tunnisteeseen sync4j_principal-taulussa.
sync_source	on viittaus synkronointilähteen tunnisteeseen sync4j_source-taulussa.
last_anchor	on SyncML-protokollan mukainen last-synkronointiankkuri.
start_sync	on viimeisimmän synkronoinnin aloitusajankohta.
end_sync	on viimeisimmän synkronoinnin lopetusajankohta.

Taulukko 4. sync4j_last_sync-taulu.

Nimi	sync4j_client_mapping
Tarkoitus	Taulu kuvaa palvelimen ja asiakkaan tapahtumien avainten (luid ja guid) vastaavuudet synkronointilähdekohtaisesti.
Avainkenttä	principal, sync_source, luid, guid
Kentät	
principal	on viittaus laite- ja käyttäjäparin tunnisteeseen sync4j_principal-taulussa.
sync_source	on viittaus synkronointilähteen tunnisteeseen sync4j_source-taulussa.
luid	on tapahtuman avain asiakkaan tietokannassa.
guid	on tapahtuman avain Korpin tietokannassa, eli viittaus tapahtuman tunnisteeseen Event-taulussa.

Taulukko 5. sync4j_client_mapping-taulu.

6.2 Sync4j-palvelimen ja Kotka-tietokannan yhteistoimintaan tarvittavat taulut

Taulukot 6-8 mahdollistavat Kotka-tietokannan ja Sync4j-palvelimen yhteistoiminnan synkronointijärjestelmässä.

Nimi	EventPrincipal
Tarkoitus	Taulu kuvaa kutakin tapahtumaa seuraavat laite- ja käyttäjäparit.
Avainkenttä	guid_eventid, principal
Kentät	
eventid	on viittaus tapahtuman tunnisteeseen Event-taulussa.
principalid	on viittaus laite- ja käyttäjäparin tunnisteeseen sync4j_principal-taulussa.

Taulukko 6. eventprincipal-taulu.

Nimi	eventConflict
Tarkoitus	Ilmaisee kutakin tapahtumaa koskevat konflikti- ja ryhmäristiriitatilanteet.
Avainkenttä	eventid, conflicteventid
Kentät	
eventid	on viittaus tapahtuman tunnisteeseen Event-taulussa.
conflicteventid	on viittaus sen tapahtuman tunnisteeseen, jonka kanssa eventid:llä viitattu tapahtuma on konfliktissa.
deleted	kertoo, onko rivi poistettu.
typeid	kertoo konfliktityypin, jotka ovat: 1 protokollan mukainen konflikti ja 2 ryhmätapahtumaristiiriita.

Taulukko 7. eventConflict-taulu.

Nimi	eventRecurrence
Tarkoitus	Taulu sisältää toistuviin tapahtumiin liittyvää lisätietoa toistumissäännöstä ja juuritapahtumasta (=tapahtuma, josta toisteet kopioidaan).
Avainkenttä	Eventide
Kentät	
eventid	on viittaus tapahtuman tunnisteeseen Event-taulussa.
recurrencefrom	on viittaus juuritapahtuman tunnisteeseen Event-taulussa. Pitäisi voida olla NULL, lue juuritapahtuma.
deleted	kertoo, onko rivi poistettu.
rule	on tapahtuman toistumissääntö.

Taulukko 8. eventRecurrence-taulu.

7 TESTAUSPERIAATTEITA

Ryhmä suorittaa testausta aina, kun on saatu valmiiksi jokin toimiva ohjelman osa. Synkronointipalvelimen testausta suoritetaan Nokian eri puhelinmalleilla, Palm-taskumikrolla, NovellGroupWise-ohjelmalla ja MultiSync-synkronointiohjelmalla, joka tarjoaa synkronointirajapinnan monille eri kalenterityypeille. Lisäksi synkronointia on tarkoitus testata jollakin Outlookin asiakas-ohjelmalla.

Konfliktienhallintaosaa testataan Windows- ja Linux-ympäristöissä käytössä olevilla selaimilla. Windows XP-käyttöjärjestelmässä testauksessa käytettävät selaimet ovat MozillaFirebird 0.6.1, Opera 7.2 ja Internet Explorer 6.0. Linuxissa selaimena on Mozilla 1.4.1.

8 LÄHDEKOODIN KOMMENTOINTI JA NIMEÄMISKÄYTÄNNÖT

Ohjelmakoodin kommentoinnissa ja muuttujien nimeämisessä tullaan noudattamaan kahta eri käytäntöä. Sync4j-palvelimeen liittyvät koodit kommentoidaan käyttäen valmista Sync4j-koodausstandardia [1].

Korppi-järjestelmään liittyvät koodit kommentoidaan käyttäen yleisiä Korpin kommentointikäytänteitä. Kommentointi ja muuttujien nimeäminen tapahtuvat seuraavien periaatteiden mukaisesti:

- Metodien ja attribuuttien nimet aloitetaan pienellä kirjaimella.
- Luokkien nimet aloitetaan isolla kirjaimella.
- Kaikkien luokkamäärittysten, attribuuttien ja metodien alkuun lisätään JavaDoc - kommentit.
- Tiedostojen alkuun liitetään GPL-lisenssin suositusten mukaan kuvaus lisenssistä.
- Koodissa käytetään kahden merkin sisennystä.
- Lähdekoodi ja kommentit kirjoitetaan englanniksi.
- JSP-sivujen HTML-elementit kirjoitetaan XHTML 1.0-standardin mukaisesti.
- JSP-sivuilla ei käytetä tag-ominaisuutta.

Seuraavassa on esimerkki kommenttilohkosta konfliktienhallintaosassa. Kommenttilohko on samanlainen sekä JSP- että java-tiedostossa.

```

/*****
*****
*
*                               PAGE DESCRIPTION
*
*****
*****
*
* NAME: test.jsp
* LANGUAGE: jsp
* DATE: 22.3.2004
* AUTHOR: Timo Aarniovuori, Jyväskylän yliopisto
*
*****
*****
*                               COPYRIGHT (C) KYHKY-PROJEKTIRYHMÄ
*
*   This file is part of Korppi.
*
*   Korppi is free software; you can redistribute it and/or modify
*   it under the terms of the GNU General Public License as
published by
*   the Free Software Foundation; either version 2 of the License,
or
*   (at your option) any later version.
*
*   Korppi is distributed in the hope that it will be useful,
*   but WITHOUT ANY WARRANTY; without even the implied warranty of
*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
*   GNU General Public License for more details.
*
*   You should have received a copy of the GNU General Public License
*   along with Korppi; if not, write to the Free Software
*   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307  USA
*****
*****/

```

```

/*****
*****
*
*                               UPDATES
*
*****
*****
*

```

```
* 24.3.2004
*   - Page completely rewritten (Timo Aarniovuori)
*
*****
*****/
```

```
<%
```

```
Some kind of code
```

```
%>
```

Seuraavassa esimerkki Sync4j:n java-luokasta.

```
/*
```

```
Copyright (c) 2001 sync4j project
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "sync4j" must not be used to endorse or promote products derived from this software without prior written permission.
4. Products derived from this software may not be called "sync4j", nor may "sync4j" appear in their name, without prior written permission.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the

software itself an acknowledgement equivalent to the following:

"This product includes software developed by the sync4j project."

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE SYNC4J AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

```
package sync4j;
```

```
import java.io.IOException;
```

```
/**
```

```
* This is the template for Sync4j Java source files. This section  
* is in HTML, so separate any paragraphs with HTML paragraph tags.
```

```
*
```

```
* <P>
```

```
*
```

```
* For example, this is a new paragraph.
```

```
*
```

```
* <P>
```

```
*
```

```
* @see AnotherClass
```

```
* @author Your name
```

```
* @version $Id: Sync4j\040Coding\040Standards.doc,v 1.1.1.1
```

```
2002/11/02 13:22:13 stefano_fornari Exp $
```

```
*
```

```
**/
```

```
public class Template {
```

```
// -----
```

```
Constants
```

```

/** A constant */
public static final String A_CONSTANT = "a constant value";

// -----
Private data

/** A member variable */
private String member;
/** A static member variable /
private String staticMemberVariable;

// -----
Properties

/**
 * a property
 */
String SomeProperty;

/**
 * Sets property SomeProperty
 * @beaninfo
 * description: (description of the purpose of this property)
 * displayName: (user-readable English name of this property)
 */
public void setSomeProperty(String someProperty) {
    this.someProperty = someProperty;
}

/**
 * Returns property SomeProperty
 */
public String getSomeProperty() {
    return someProperty;
}

/**
 * a boolean property
 */
boolean booleanProperty;

```

```

/**
 * Sets property BooleanProperty
 *
 * @beaninfo
 * description: (description of the purpose of this property)
 * displayName: (user-readable English name of this property)
 **/
public void setBooleanProperty(boolean booleanProperty) {
    this.booleanProperty = bBooleanProperty;
}

/**
 * Returns property BooleanProperty
 **/
public boolean isBooleanProperty() {
    return booleanProperty;
}

// -----
Constructors

/**
 * Constructs a null Template.
 */
public Template (String someParameter) {
}

// ----- Public
methods

/**
 *
 * A sample function
 *
 * @param pArg and argument
 * @return a string value
 * @exception IOException if something went wrong
 **/
public String myFunction (int pArg)
throws IOException
{
    return "something";
}

```

```

}

//-----
// Overrides java.lang.Object method
//-----
/**
 * @return a String representation of this object
 */
public String toString ()
{
    StringBuffer buf = new StringBuffer ();
    buf.append (getClass ().getName ());
    buf.append ('[');
    buf.append ("this object's value, as a String");
    buf.append (']');
    return buf.toString ();
}

// ----- Private
methods
/**
 *
 * A sample private function
 *
 */
private void aPrivateMethod(){
}
}

```

9 YHTEENVETO

Sovellussuunnitelmassa on kuvattu toteutettavan synkronointijärjestelmän arkkitehtuuria, toimintaa ja suunnitelmia. Koska järjestelmä vaatii toimiakseen sekä Sync4j-palvelimen ohjelmointia että konfliktienhallintaosan Korppi-järjestelmään, on näitä osia pyritty kuvaamaan kattavasti. Lisäksi on käyty läpi lähdekoodin kommentointia ja nimeämiskäytäntöjä sekä testausperiaatteita ja tietokantarakennetta.

Konfliktienhallintaosan käyttöliittymän ja tietokantaan tulee useita muutoksia. Lopulliset versiot kuvataan Sovellusraportissa.

LÄHTEET

- [1] Fornari Stefano, "Sync4j Coding Standards", saatavilla HTML-muodossa <URL: <http://sync4j.funambol.com/project/development/sync4j-coding-standards.html>>, Sync4j, 2002.
- [2] Gamma E., Helm R., Johnson R., Vlissides J., "Design Patterns - Elements of Reusable Object-Oriented Software", Addison-Wesley, 1995.
- [3] Hilpinen Toni, Koivuniemi Marko, Mäkinen Jussi ja Nurminen Miika, "Kiuru-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [4] Lintunen Sampsa, Pöyhönen Tiina, Vähä-Ruka Teemu ja Ylönen Timo, "Kottarainen-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2003.
- [5] Paadar Juha Pekka, "SyncML:n käyttö henkilökohtaisen tiedon synkronointiin", saatavilla pdf-muodossa <URL: <http://www.cc.jyu.fi/~ppaadar/gradu.pdf>>, pro gradu -tutkielma, Jyväskylän yliopisto, tietotekniikan laitos, 2003.
- [6] Santanen Jukka-Pekka, "Opinnäytteiden kirjoittaminen, lyhyt oppimäärä", Jyväskylän yliopisto, tietotekniikan laitos, 2000.
- [7] Santanen Jukka-Pekka, "Tietotekniikan Sovellusprojektien ohje", Jyväskylän yliopisto, tietotekniikan laitos, 2004.
- [8] Open Mobile Alliance, "SyncML Representation Protocol Specification V1.0", saatavilla pdf-muodossa <URL: http://www.openmobilealliance.org/tech/affiliates/syncml/syncml_represent_v10_20001207.pdf>, Open Mobile Alliance, 2000.

LIITE. TIETOKANNAN ER-KAAVIO

