

Suunnitteludokumentti

Oppimistavoitteiden hallintajärjestelmä harri

Helsinki 19.11.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9 op)

Projektiryhmä

Petri Kinnunen

Lasse Leino

Anne Pääkkö

Minna Ulmala

Asiakas

Harri Laine

Johtoryhmä

Kimmo Simola, vastuuhenkilö

Aleksi Yrttiaho, ohjaaja

Kotisivu

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

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
1.0	19.11.2007	Dokumentti oikoluettu
0.5	15.11.2007	Luokkasuunnittelu valmis
0.4	11.11.2007	Luokkasuunnittelun viimeistelyä
0.3	28.10.2007	Kuvat käyttöliittymästä dokumenttiin
0.2	24.10.2007	Käyttöliittymäsuunnittelu päivitetty
0.1	06.10.2007	Ensimmäinen L ^A T _E X-versio

Sisältö

1	Johdanto	1
1.1	Toteutuskieli ja -ympäristö	1
1.2	Sanasto	1
2	Arkkitehtuurisuunnittelu	2
2.1	Osajärjestelmät	2
2.1.1	Näkymä	2
2.1.2	Ohjain	3
2.1.3	Malli	3
3	Komponenttisuunnittelu	3
3.1	Käyttöliittymäsuunnittelu	3
3.1.1	Haku	4
3.1.2	Pikavalinta	4
3.1.3	Kurssilistaus	5
3.1.4	Kurssi	5
3.1.5	Oppimistavoitteet	6
3.1.6	Oppimistavoitteiden muokkaus	6
3.1.7	Esitietovaatimukset	7
3.1.8	Esitietovaatimusten muokkaus	8
3.1.9	Henkilöhallinta	8
3.2	Tietokantasuunnittelu	9
3.2.1	SQL-taulut	10
3.2.2	SQL-käskyjä kurssin lukitsemiseen	12
3.3	Integroinnin suunnittelu	12
3.3.1	Opettaja-näkymä	13
3.3.2	Opjakso-näkymä	13
3.3.3	Käyttäjän tunnistaminen	14
4	Luokkien suunnittelu	14
4.1	Yleistä	14
4.1.1	Enumit	14
4.2	Malli	15

4.2.1	Kurssihallinta	15
4.2.2	Kayttajahallinta	16
4.2.3	Tilahallinta	17
4.2.4	Selite	18
4.2.5	Kurssi	18
4.2.6	Teema	19
4.2.7	Esitietovaatimus	19
4.2.8	Oppimistavoite	20
4.3	Näkymän JSP-sivut	20
4.4	Ohjain	21

1 Johdanto

Oppimistavoitteiden hallintajärjestelmä eli harri tehdään Helsingin yliopiston Tietojenkäsittelytieteen laitokselle.

Järjestelmän avulla voidaan määritellä ja hallita kurssien eritasoisia oppimistavoitteita. Järjestelmän käyttäjät, opiskelijat ja opettajat, näkevät valitsemansa kurssin esitetietovaatimukset luettelona. Kurssin teemat ja oppimistavoitteet he näkevät hierarkkisesti. Kurssien suunnittelijat eli opettajat pääsevät muuttamaan niiden kurssien tietoja, joihin heillä on oikeudet. He voivat ylläpitää esitetietovaatimuksia, oppimistavoitteita ja teemoja. Lisäksi he voivat verrata esitetietovaatimusten ja oppimistavoitteiden oppimisen syvyys -arvoja.

Tässä dokumentissa kuvataan, miten järjestelmä tullaan toteuttamaan. Arkkitehtuurisuunnittelussa järjestelmä jaetaan osajärjestelmiin, jotka ovat järjestelmässä toimivia loogisia kokonaisuuksia. Ne voivat toimia riippumatta muista osajärjestelmistä. Jako seuraa MVC-arkkitehtuurityyliä (Model-View-Controller). Komponenttisuunnittelussa käydään yksityiskohtaisemmin läpi jokainen osajärjestelmä ja sen sisältö sekä suunnitellaan käyttöliittymä, tietokanta sekä integrointi Opetuksensuunnittelujärjestelmään. Suunnittelun tuloksena saadaan myös kuvaukset kaikista luokista.

Ryhmä toteuttaa Vaatimusdokumentissa 1. prioriteetilla sekä osan 2. prioriteetilla kuva-
tuista toiminnoista.

1.1 Toteutuskieli ja -ympäristö

Ohjelmisto toteutetaan Java-ohjelmointikielen versiolla 1.5 . Koska järjestelmä on tietokantapohjainen websovellus, käyttöliittymän toteutuksessa käytetään JSP (Java Server Pages) -tekniikkaa. Käyttöliittymän ulkonäköön käytetään CSS (Cascading Style Sheet) -tyylimäärittelyjä. Tietokanta tehdään Oracle:n versiolla 10g Enterprise Edition Release 10.2.0.3.0. Tietokantaa käsittelevät luokat toteutetaan Java Servletteinä, joissa ovat SQL-lauseet.

1.2 Sanasto

JSP eli Java Server Pages

JSP on Java-teknologia, jonka avulla voidaan dynaamisesti luoda HTML, XML tai muun tyyppisiä dokumentteja vastauksena Web-asiakkaan pyyntöön. Teknologia erottaa käyttöliittymän sisällöntuotannosta sallien muutokset käyttöliittymän ulkoasuun ilman että dynaamista sisältöä pitää muuttaa.

Java Servlet

Java Servletit ovat alustasta riippumattomia palvelimella olevia moduleita, joiden avulla saadaan lisättyä dynaamista sisältöä Web-palvelimelle. Luotu sisältö on yleensä HTML:ää, mutta se voi olla myös XML:ää. Servletit voivat hallita tiloja useiden palvelintransaktioiden läpi käyttäen HTTP-keksyjä, istuntomuuttujia tai URL-uudelleenkirjoitusta.

MVC eli Model-View-Controller

MVC on arkkitehtuurityyli, jossa ohjelma jaetaan kolmeen osaan: malliin, näkymään ja ohjaimen. Näiden tarkoitus on erottaa käyttöliittymä varsinaisesta toimintalogiikasta ja tietosisällöstä.

OSJ eli Opetuksensuunnittelujärjestelmä

OSJ on Helsingin yliopiston Tietojenkäsittelytieteen laitoksen tietojärjestelmä, jossa on laitoksella työskentelevien henkilöiden ja luennoitavien kurssien tiedot.

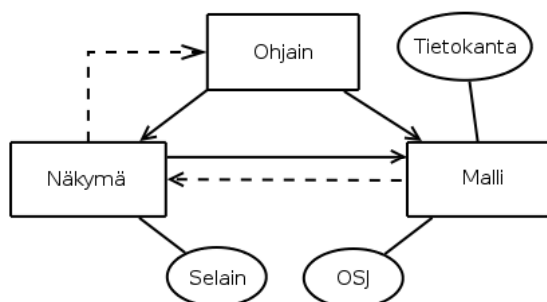
Versio

Versio on kurssiversion eri tilojen jaottelu kahteen ryhmään, julkiseen ja työversioon. Ryhmään julkinen kuuluu vain kurssiversion tila julkinen ja ryhmään työversio kuuluvat muut kurssiversion tilat: muokattava, suojattu, tarkastettava ja hylätty

2 Arkkitehtuurisuunnittelu

2.1 Osajärjestelmät

MVC-arkkitehtuurityylin mukaisesti järjestelmä koostuu kolmesta osajärjestelmästä: *näkymästä*, *ohjaimesta* ja *mallista*. Näiden lisäksi järjestelmässä on kaksi muuta osajärjestelmää: *tietokanta* ja ulkoinen järjestelmä *OSJ*.



Kuva 1: Järjestelmän arkkitehtuuri

2.1.1 Näkymä

Näkymässä toteutetaan selainpohjainen käyttöliittymä, joka vastaanottaa käyttäjäsyötteitä selaimen kautta. Käyttäjäsyytteet lähetetään ohjaimelle käsiteltäväksi.

2.1.2 Ohjain

Ohjain vastaanottaa näkymältä käyttäjäsyötteitä ja käsittelee ne. Kun käyttäjäpyyntö edellyttää tiedon hakua, tallennusta tai muokkausta, ohjain välittää pyynnön mallille.

2.1.3 Malli

Malli ottaa vastaan välitettyjä käyttäjäpyyntöjä ohjaimelta sekä sisäisiä järjestelmäpyyntöjä näkymältä. Se on yhteydessä kahteen osajärjestelmään: tietokantaan ja OSJ:ään.

Tietokanta toimii järjestelmän tietovarastona ja malli ylläpitää sitä SQL-lauseilla. Malli käsittelee vastaanotetut käyttäjä- ja järjestelmäpyynnöt sekä ottaa yhteyden tietokantaan, joko tiedon hakua tai tallennusta varten. Kun operaatio on tehty, malli ilmoittaa näkymälle muutoksen onnistumisesta tai palauttaa järjestelmäpyynnön haluaman tiedon.

OSJ:stä malli hakee mm. kurssi- ja henkilötietoja käyttäen näkymätauluja.

Malli käsittelee lisäksi asetustiedostoa (Configuration file), joka sisältää tietoja järjestelmän alkuarvoista. Tiedosto luetaan, kun järjestelmä käynnistetään. Asetustiedostossa määritellään järjestelmävastaava.

3 Komponenttisuunnittelu

Malli koostuu Java Servleteistä ja Java-luokista. Tämä komponentti on järjestelmän osa, joka on yhteydessä tietokantaan. Komponentin servletteihin on toteutettu tietokantaa muokkaavat tai tietoja kyselevät SQL-lauseet. Luokkien toteutus on kuvattu kappaleessa 4.2.

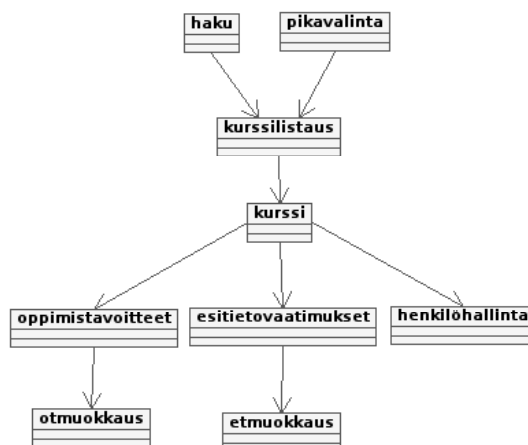
Näkymä koostuu JSP-tiedostoista. Tarkempi käyttöliittymäsuunnittelu on kuvattu kappaleessa 3.1 ja JSP-sivujen suunnittelu kappaleessa 4.3.

Ohjaimessa on vain yksi komponentti, joka vastaanottaa näkymältä tulevat muokkauspyynnöt ja lähettää nämä eteenpäin mallille. Tämän komponentin toteutus on kuvattu kappaleessa 4.4.

3.1 Käyttöliittymäsuunnittelu

Käyttöliittymä koostuu yhdeksästä käyttöliittymäkomponentista, joita ovat: *haku*, *pikavalinta*, *kurssilistaus*, *kurssi*, *oppimistavoitteet*, *oppimistavoitteiden muokkaus*, *esitietovaatimukset*, *esitietovaatimusten muokkaus* ja *henkilöhallinta* (kuva 2). Komponentit ovat käyttöliittymän osia, joita yhdistelemällä muodostetaan käyttäjälle näkyviä sivuja.

Sanalla *käyttäjä* viitataan yleisesti kaikkiin tietojärjestelmän käyttäjäryhmiin.



Kuva 2: Käyttöliittymän sivukartta

3.1.1 Haku

Haku on käyttöliittymäkomponentti, joka näkyy jokaisen sivun ylälaudassa ja sitä käytetään kurssien hakuun tietokannasta. Se koostuu *hakukentästä* ja *hakunapista* (kuva 3). Hakukenttä toimii kuten tavallinen lomakekenttä. Kurssia voidaan hakea kurssin nimellä (suomeksi tai englanniksi) tai kurssikoodilla. Käyttäjän suorittaessa haun, tulostetaan kurssilistaus kyseisellä hakuehdolla. Haku käynnistetään painamalla enteriä tai hakunappia.

Kuva 3: Haku

3.1.2 Pikavalinta

Pikavalinta sijaitsee hakupalkin alapuolella (kuva 4). Sitä käytetään kurssilistausten nopeaan hakuun tietokannasta. Kaikille muille käyttäjäryhmille paitsi opiskelijalle tarjotaan pikavalinnat: *omat kurssit* ja *kaikki kurssit*. Käyttäjän valitessa kaikki kurssit, näytetään kurssilistaus kaikista tietojärjestelmässä olevista kursseista. Valittaessa omat kurssit, näytetään käyttäjälle kurssilistaus kursseista, joihin hänellä on muokkaus- tai tarkastusoikeudet. Opiskelijalle näytetään kurssilistaus kaikista kursseista, kun kursseja haetaan ilman hakuehtoja.

Hakukentän ja hakunapin lisäksi pikavalintakomponentissa on myös kielen valinta painikkeet; toinen suomenkielille ja toinen englanninkielille. Suomenkieli on aloitusnäkyssä oletuksena.

Kuva 4: Pikavalinta

3.1.3 Kurssilistaus

Kurssilistaus on taulukkomuotoinen listaus kursseista (kuva 5) ja sen sisältö riippuu käyttäjistä. Opiskelijoille näytetään jokaisesta kurssista kurssin nimi ja kurssikoodi. Muille käyttäjille näytetään myös kurssin työversio. Opiskelijat näkevät vain julkaistut kurssit. Muut käyttäjät näkevät tietojärjestelmän kaikki kurssit. Kaikille muille käyttäjille, paitsi opiskelijoille ja järjestelmävastaavalle, näytetään oletus kurssilistaus, joka sisältää käyttäjän omat kurssit. Käyttäjän valitessa kurssin, siirrytään Kurssi-näkymään.

Hakusanalla nalle löytyi 5 kurssia.

Kurssikoodi	Kurssi	Työversio
58110	Johdatus nallekarhuihin	Muokattava
58120	Nallekarhujen anatomia	Muokattava
58130	Nallekarhujen ruokavalio	Suojattu
58140	Nallekarhujen kasvatusta	Suojattu
59110	Nallekarhujen sosiaalinen elämä	Tarkastettava

Kuva 5: Kurssilistaus

3.1.4 Kurssi

Kurssi-käyttöliittymäkomponentin otsikkona on kurssin nimi ja kurssikoodi. Sivun ylä-laidassa kerrotaan, missä tilassa kurssi on ja tarjotaan tilanmuutosnappi. Nämä eivät näy opiskelijoille. Kurssin tilaa voidaan muuttaa valitsemalla pudotuslistasta uusi tila ja painamalla tilanmuutosnappia. Muokkaaja ei pysty vaihtamaan kurssin tilaa. Vastuuhenkilö voi vaihtaa kurssin tilan joko suojatuksi tai muokattavaksi. Tarkastaja voi vaihtaa tilan joko julkaistuksi tai hylätyksi, jolloin kurssi siirtyy takaisin suojattuun tilaan.

Kurssin tiedot jaetaan kolmeen valinnaiseen osaan: oppimistavoitteisiin, esitetietovaatimukseen ja henkilöhallintaan. Kuhunkin osaan voidaan siirtyä kurssin nimen alla olevien linkkien avulla. Kun siirrytään kurssi-näkymään, oppimistavoitteet ovat oletuksena. Henkilöhallinta liittyy kurssin muokkaukseen ja esitellään muokkaus-osiossa kappaleessa 3.1.9. Henkilöhallinta-linkki näkyy vain kurssin vastuuhenkilölle ja järjestelmävastaavalle.

Kurssin nimi, kurssikoodi, tila ja navigointi linkit ovat näkyvillä jokaisella kurssisivulla, niin muokkaus- kuin selausnäkymissä.

Nallekarhujen sosiaalinen elämä (59110)

[Oppimistavoitteet](#) | [Esitetietovaatimukset](#) | [Henkilöhallinta](#)

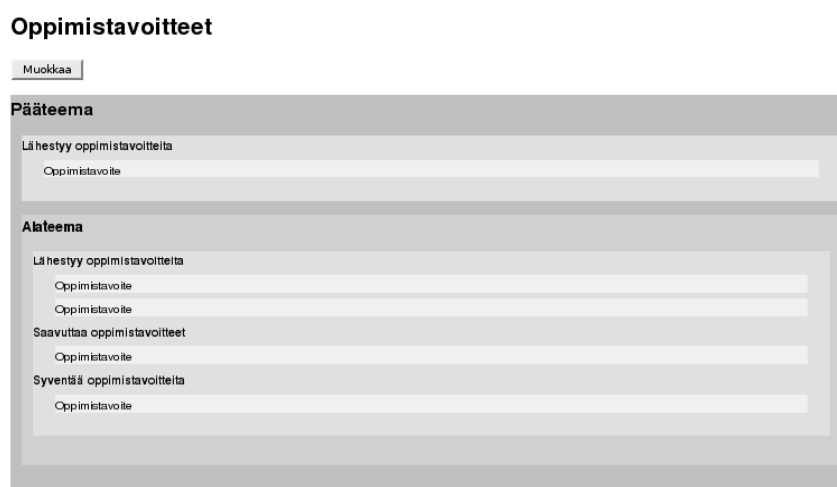
Kuva 6: Kurssi

3.1.5 Oppimistavoitteet

Oppimistavoitteet esitetään hierarkkisesti listattuna laatikoissa. Laatikko on ympäristöstään eri värillä korostettu alue, jolla tieto erotetaan muista. Pääteema muodostaa yhden laatikon. Sen sisälle tulee oppimistavoitteita ja alateemoja. Alateemat muodostavat toisen laatikon, joka on sisennetty pääteeman alle. Alateeman sisälle tulee oppimistavoitteita.

Oppimistavoitteet muodostavat oman laatikon, joka on sisennetty joko pää- tai alateeman alle. Jokainen oppimistavoitelatikko sisältää otsikot lähestyy, saavuttaa ja syventää oppimistavoitteita. Näiden alla on itse oppimistavoitteet listattuna (kuva 7).

Oppimistavoitteiden muokkausnäkyminen siirrytään painamalla listauksen yläpuolella olevaa muokkaa nappia. Nappi näkyy vain kurssin vastuuhenkilölle, muokkaajille ja järjestelmävastaavalle.



Kuva 7: Oppimistavoitteet

3.1.6 Oppimistavoitteiden muokkaus

Oppimistavoitteiden muokkausnäkyminen on identtinen selausnäkyminen kanssa, mutta pääteemojen, alateemojen ja oppimistavoitteiden kuvaukset ovat muokattavissa tekstikentissä (kuva 8). Oppimistavoitteita voi muokata muokkaaja, vastuuhenkilö ja järjestelmävastaava.

Pääteemoja voi lisätä painamalla listauksen yläpuolella olevaa lisää pääteema -painiketta. Tällöin ilmestyy uudelle pääteemalle tyhjä tekstikenttä suomeksi ja englanniksi sekä lisää alateema -painike. Tällä painikkeella alateemoja voi lisätä kuhunkin pääteemaan. Oppimistavoitteita voidaan lisätä painamalla lisää oppimistavoite painiketta, joita on jokaisen oppimistason kuvauksen vieressä.

Jokaisen pääteeman, alateeman ja oppimistavoitteen vieressä on valintapainike, jolla kyseinen elementti voidaan merkitä poistettavaksi. Elementit poistetaan painamalla poista

valitut -painiketta. Oppimistavoitteiden tekstikenttiä on poisto-valintapainikkeen vieressä kaksi, toinen suomenkielistä kuvausta ja toinen englanninkielistä kuvausta varten. Kuvausten jälkeen on oppimisen syvyyden valinta alasetelista.

Kun tarvittavat muutokset on tehty painetaan listauksen yläpuolella olevaa tallenna muutokset -nappia, joka tallentaa muutokset tietokantaan ja näyttää uuden oppimistavoite-näkymän.

Oppimistavoitteet

Lisää pääteema Poista valitut Tallenna muutokset

Pääteema In English Lisää alateema

Lähesty oppimistavoitteita Lisää oppimistavoite

Oppimistavoite In English Muistaa ulkoa

Saavuttaa oppimistavoitteet Lisää oppimistavoite

Syventää oppimistavoitteita Lisää oppimistavoite

Alateema In English

Lähesty oppimistavoitteita Lisää oppimistavoite

Oppimistavoite In English Muistaa ulkoa

Oppimistavoite In English Muistaa ulkoa

Saavuttaa oppimistavoitteet Lisää oppimistavoite

Oppimistavoite In English Muistaa ulkoa

Syventää oppimistavoitteita Lisää oppimistavoite

Oppimistavoite In English Muistaa ulkoa

Kuva 8: Oppimistavoitteiden muokkaus

3.1.7 Esitietovaatimukset

Esitietovaatimukset esitetään kolmen sarakkeen taulukkomuodossa (kuva 9). Sarakkeiden otsikot ovat: esitietovaatimus, kurssi ja pääteema. Esitietovaatimuksen alle tulee esitietovaatimusten kuvaukset, kurssin alle, mihin kurseihin esitietovaatimus liittyy ja pääteeman alle, mihin pääteemoihin kurssi liittyy. Esitietovaatimusten muokkausnäkyminen siirrytään painamalla listauksen yläpuolella olevaa muokkaa-nappia. Nappi näkyy vain kurssin vastuuhenkilölle, muokkaajille ja järjestelmävastaavalle.

Esitietovaatimukset

Muokkaa

Esitietovaatimus	Kurssi	Pääteemat
Perus nalletetus	Johdatus nallekarhuihin, Nallekarhujen anatomia	-
Nallekarhujen munosikä	Nallekarhujen kasvatus	Munosikä

Kuva 9: Esitietovaatimukset

3.1.8 Esitietovaatimusten muokkaus

Esitietovaatimusten muokkausnäky on identtinen selausnäkyänsä kanssa, mutta esitietovaatimukset ovat nyt muokattavissa tekstikentissä (kuva 10). Esitietovaatimuksia voi muokata muokkaaja, vastuuhenkilö ja järjestelmävastaava.

Ensimmäisenä sarakkeena on valintapainike, jolla voidaan valita esitietovaatimuksia poistettavaksi. Varsinainen poistaminen tehdään painamalla poista valitut -painiketta. Toinen sarake on muokattava esitietovaatimuksen tekstikenttä, joka on sekä suomeksi että englanniksi. Kolmas sarake on esitietovaatimuksen syvyys, joka valitaan alasvetolistasta.

Neljäs sarake on kurssilistaus, josta voidaan valita kurssit, mihin kyseinen esitietovaatimus on kytköksissä. Jos kurssia ei löydy listasta, voidaan sen nimi kirjoittaa listauksen vieressä olevaan tekstikenttään. Viimeisen sarakkeen listasta voidaan valita mihin pääteemoihin esitietovaatimus liittyy. Uusia esitietovaatimuksia saadaan painamalla uusi esitietovaatimus -painiketta. Kun muutokset on tehty painetaan tallenna muutokset -painiketta.

Esitietovaatimusten muokkaus

Uusi esitietovaatimus Poista valitut Tallenna muutokset

Esitietovaatimus	Syvyys	Kurssit	Muut kurssit	Pääteemat
<input type="checkbox"/> Perus nalletietous <input type="checkbox"/> Basic teddybear knowledge	Muistaa ulkoa	Johdatus nallekarhuihin Nallekarhujen anatomia Nallekarhujen ruokavalio		Murrosikä Aamuruoka Iltaruoka
<input type="checkbox"/> Nallekarhujen murrosikä <input type="checkbox"/> Puberty of Teddybears	Muistaa ulkoa	Johdatus nallekarhuihin Nallekarhujen anatomia Nallekarhujen ruokavalio		Murrosikä Aamuruoka Iltaruoka

Kuva 10: Esitietovaatimusten muokkaus

3.1.9 Henkilöhallinta

Henkilöhallinta näkyy käyttäjäryhmästä riippuen joko kahden tai neljän sarakkeen taulukkomuotoisena listauksena (kuva 11). Vastuuhenkilö saa käyttäjätaulukon, jossa ensimmäisen sarakkeen otsikkona on nimi ja toisen sarakkeen otsikkona muokkaaja. Nimi-sarakkeeseen listataan käyttäjien nimilyhennetunnukset ja muokkaaja-sarakkeeseen listataan valintapainike kunkin tunnuksen viereen.

Muokkaajan oikeuksia jaetaan valitsemalla valintapainikkeilla halutut henkilöt, jonka jälkeen painetaan tallenna -painiketta.

Järjestelmävastaavan henkilöhallinta näkymä toimii samoin, mutta järjestelmävastaavan sarakkeet ovat seuraavat: nimilyhennetunnus, muokkaaja, vastuuhenkilö ja tarkastaja. Järjestelmävastaava voi siis jakaa kaikkia käyttöoikeuksia järjestelmään.

Henkilöhallinta

Tallenna			
Nimi	Muokkaja	Vastuuhenkilö	Tarkastaja
Peter Steele	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Frank Horigan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Jack Bauer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kuva 11: Henkilöhallinta

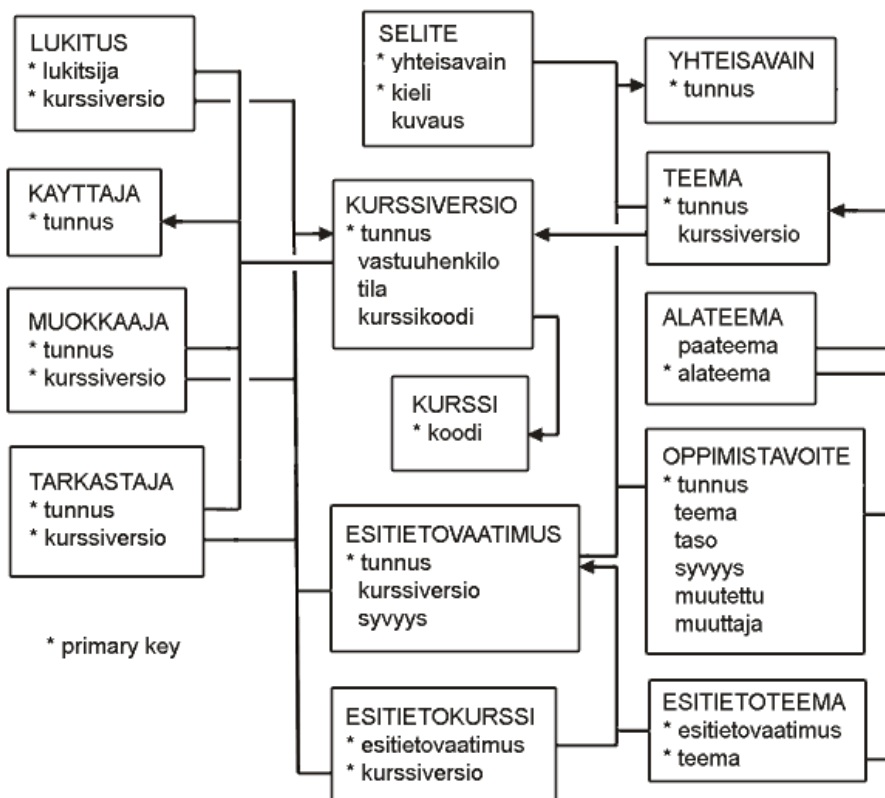
3.2 Tietokantasuunnittelu

Tietokanta on suunniteltu vaatimuskäsitteiden pohjalta. Tietokantakaaviosta näkyvät taulujen väliset yhteydet. Tietokannan taulut täyttävät seuraavat säännöt:

- taulussa ei ole toistuvia rivejä tai tyhjiä sarakkeita
- kaikki taulun ei-avain-sarakkeet ovat riippuvaisia koko avaimesta
- mikään taulun ei-avain-sarakkeista ei ole transitiivisesti riippuva avaimen sarakkeista
- mikään taulun ei-avain-sarakkeista ei ole funktionaalisesti riippuva muista kuin avaimen sarakkeista

Edellisestä seuraa, että tietokannan taulut ovat Boyce-Codd -normaalimuodossa.

Oppimistavoite-aulussa olevien kenttien taso ja syvyys selväkieliset tekstit ovat asetustiedostossa. Yhteisavain-aulussa luodaan yksilölliset avaimet esitetietovaatimuksille, teemoille ja oppimistavoitteille. Ts. näiden taulujen avaimet muodostavat yhtenäisen sarjan: 1, 2, 3 jne. Kurssin versiohallinta hoidetaan kurssiversio-aulun avulla.



Kuva 12: Järjestelmän tietokantakaavio

3.2.1 SQL-taulut

```

CREATE TABLE kayttaja (
    tunnus          VARCHAR(20) PRIMARY KEY
);
CREATE TABLE kurssi (
    koodi           VARCHAR(15) PRIMARY KEY
);
CREATE TABLE kurssiversio (
    tunnus          VARCHAR(20),
    vastuuhenkilo  VARCHAR(20) NOT NULL
        REFERENCES kayttaja(tunnus) ON DELETE CASCADE,
    tila            CHAR(1) DEFAULT 's',
    kurssikoodi    VARCHAR(15) NOT NULL
        REFERENCES kurssi(koodi) ON DELETE CASCADE,
    PRIMARY KEY (tunnus)
);
CREATE TABLE muokkaaja (
    tunnus          VARCHAR(20) NOT NULL
        REFERENCES kayttaja(tunnus) ON DELETE CASCADE,
    kurssiversio   VARCHAR(20) NOT NULL
        REFERENCES kurssiversio(tunnus) ON DELETE CASCADE,
    PRIMARY KEY (tunnus, kurssiversio)

```

```

);
CREATE TABLE tarkastaja (
    tunnus          VARCHAR(20) NOT NULL
                   REFERENCES kayttaja(tunnus) ON DELETE CASCADE,
    kurssiversio   VARCHAR(20) NOT NULL
                   REFERENCES kurssiversio(tunnus) ON DELETE CASCADE,
    PRIMARY KEY (tunnus, kurssiversio)
);
CREATE TABLE lukitus (
    lukitsija      VARCHAR(20) REFERENCES kayttaja(tunnus),
    kurssiversio   VARCHAR(20) NOT NULL
                   REFERENCES kurssiversio(tunnus) ON DELETE CASCADE,
    PRIMARY KEY (lukitsija, kurssiversio)
);
CREATE TABLE yhteisavain (
    tunnus          INTEGER PRIMARY KEY
);
CREATE TABLE teema (
    tunnus          INTEGER REFERENCES yhteisavain(tunnus)
                   ON DELETE CASCADE,
    kurssiversio   VARCHAR(20) NOT NULL
                   REFERENCES kurssiversio(tunnus) ON DELETE CASCADE,
    PRIMARY KEY (tunnus)
);
CREATE TABLE alateema (
    alateema       INTEGER
                   REFERENCES teema(tunnus) ON DELETE CASCADE,
    paateema       INTEGER
                   REFERENCES teema(tunnus),
    PRIMARY KEY (alateema)
);
CREATE TABLE oppimistavoite (
    tunnus          INTEGER REFERENCES yhteisavain(tunnus)
                   ON DELETE CASCADE,
    teema           INTEGER NOT NULL
                   REFERENCES teema(tunnus) ON DELETE CASCADE,
    taso            INTEGER NOT NULL,
    syvyys          INTEGER NOT NULL,
    muutettu       TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
    muuttaja       VARCHAR(20) NOT NULL,
    PRIMARY KEY (tunnus)
);
CREATE TABLE esitietovaatimus (
    tunnus          INTEGER REFERENCES yhteisavain(tunnus)
                   ON DELETE CASCADE,
    kurssiversio   VARCHAR(20) NOT NULL REFERENCES kurssiversio(tunnus)
                   ON DELETE CASCADE,
    syvyys         INTEGER,
    PRIMARY KEY (tunnus)
);
CREATE TABLE esitietokurssi (
    esitietovaatimus INTEGER REFERENCES esitietovaatimus(tunnus)
                   ON DELETE CASCADE,
    kurssiversio   VARCHAR(20) REFERENCES kurssiversio(tunnus)

```

```

        ON DELETE CASCADE,
    PRIMARY KEY (esitietovaatimus, kurssiversio)
);
CREATE TABLE esitietoteema (
    esitietovaatimus INTEGER REFERENCES esitietovaatimus(tunnus)
        ON DELETE CASCADE,
    teema              INTEGER REFERENCES teema(tunnus)
        ON DELETE CASCADE,
    PRIMARY KEY (esitietovaatimus, teema)
);
CREATE TABLE selite (
    yhteisavain      INTEGER NOT NULL REFERENCES yhteisavain(tunnus)
        ON DELETE CASCADE,
    kuvaus           VARCHAR(3000) NOT NULL,
    kieli             CHAR(1) NOT NULL,
    PRIMARY KEY (yhteisavain, kieli)
);

```

3.2.2 SQL-käskyjä kurssin lukitsemiseen

Lukituksen asettaminen kurssiin. Käyttäjällä voi olla kerrallaan useita kursseja lukittuna.

```

INSERT INTO lukitus VALUES (<käyttäjätunnus>, <kurssiversio>)
WHERE NOT EXISTS (SELECT * FROM lukitus WHERE kurssiversio =
<kurssiversio>)

```

Kurssin lukituksen poistaminen kurssin ja käyttäjän perusteella.

```

DELETE FROM lukitus WHERE lukitsija = <käyttäjätunnus>
AND kurssiversio = <kurssiversio>

```

Kurssien lukituksen poistaminen vain käyttäjän perusteella.

```

DELETE FROM lukitus WHERE lukitsija = <käyttäjätunnus>

```

Kurssin lukituksen poistaminen vain kurssin perusteella.

```

DELETE FROM lukitus WHERE kurssiversio = <kurssiversio>

```

Lukituksen poistaminen kaikista kursseista.

```

DELETE * FROM lukitus

```

3.3 Integroinnin suunnittelu

Rakennettavan järjestelmän tietokantaan luodaan näkymiä OSJ:ään, josta haetaan laitok-
sella työskentelevien henkilöiden ja pidettävien kurssien tietoja.

3.3.1 Opettaja-näkymä

Opettaja-näkymän avulla saadaan rakennettavan järjestelmän käyttöön laitoksella työskentelevien henkilöiden nimet, nimilyhennetunnukset, jotka ovat yksilöiviä, sekä käyttäjätunnus, jota käytetään käyttäjän tunnistamiseen rakennettavan järjestelmän sisäänkirjautumisessa.

Opettaja-näkymä on seuraavanlainen:

- **htunnus**, VARCHAR2(12), on henkilön nimilyhennetunnus, joka on yksilöivä,
- **ktunnus**, VARCHAR2(12), on henkilön käyttäjätunnus, joka on yksilöivä, mutta sitä käytetään vain henkilön tunnistamiseen sisäänkirjautumisessa,
- **sukunimi**, VARCHAR2(80), on henkilön sukunimi,
- **kutsumanimi**, VARCHAR2(15), on henkilön kutsumanimi,
- **email**, VARCHAR2(80), on henkilön sähköpostiosoite, jota ei tarvita rakennettavassa järjestelmässä.

Opettaja-näkymä luodaan tietokantaan SQL-lauseella:

```
CREATE VIEW opettaja (htunnus, ktunnus, sukunimi, kutsumanimi, email) AS
  select htunnus, ktunnus, sukunimi, kutsumanimi, sähkopostiosoite
  from tk_opha.henkilo
  where aktiivisuus='K';
```

3.3.2 Opjakso-näkymä

Opjakso-näkymän avulla saadaan rakennettavan järjestelmän käyttöön laitoksella järjestettävien opintojaksojen eli kurssien koodit sekä suomen- ja englanninkieliset nimet.

Opjakso-näkymä on seuraavanlainen:

- **kurssikoodi**, VARCHAR2(15), on kurssin koodi,
- **nimi_suomi**, VARCHAR2(80), on kurssin suomenkielinen nimi,
- **nimi_englanti**, VARCHAR2(80), on kurssin englanninkielinen nimi,

Näkymässä on lisäksi seuraavat tiedot, joita ei tarvita rakennettavassa järjestelmässä:

- **nimi_ruotsi**, VARCHAR2(80), on kurssin ruotsinkielinen nimi,
- **opintopisteet**, NUMBER(4,1), on kurssin laajuus opintopisteinä,
- **opintopisteet_ylaraja**, NUMBER(4,1), on kurssin maksimilaajuus opintopisteinä, joka voi olla eri kuin opintopisteet.

- **tyyppi**, VARCHAR2(1), on kurssin tyyppi: A = labra, K = luentokurssi tai S = seminaari,
- **taso**, VARCHAR2(1), kertoo kurssin tason: perusopinnot, aineopinnot, syventävät tai muuta.

Opjakso-näkymä luodaan tietokantaan SQL-lauseella:

```
CREATE VIEW opjakso (kurssikoodi, nimi_suomi, nimi_ruotsi, nimi_englanti,
  opintopisteet, opintopisteet_ylaraja, tyyppi, taso) AS
  select kurssikoodi, nimi_suomi, nimi_ruotsi, nimi_englanti,
  opintopisteet, opintopisteet_ylaraja, tyyppi, taso
  from tk_opha.opintojakso;
```

3.3.3 Käyttäjän tunnistaminen

Käyttäjätunnistustusta varten järjestelmä tehdään laitoksen Webbi-palvelimelle, sysdb-palvelimelle. Kun käyttäjä menee järjestelmän URL-osoitteeseen, sysdb-palvelin on määriteltä näyttämään laitoksen Intran sisäänkirjaus-sivu, jossa kysytään käyttäjän käyttäjätunnus ja salasana. Sysdb-palvelin tunnistaa käyttäjän vertaamalla käyttäjän antamia tietoja sysdb-palvelimella olevaan opettajalistaan. Jos tiedot löytyvät opettajalistasta, niin näytetään järjestelmän ensimmäinen sivu. Järjestelmä saa käyttäjätunnuksen ensimmäisellä sivulla luotavan HttpServletRequest-olion metodilla getRemoteUser. Sen jälkeen käyttäjän nimilyhennetunnus ja nimi haetaan tk_opha.opettaja-näkymästä, jollion tiedetään, kuka käyttää järjestelmää.

4 Luokkien suunnittelu

4.1 Yleistä

4.1.1 Enumit

public enum Versio { julkinen, työversio }

Versio on enum-tyyppi, joka voi saada arvokseen *julkinen* tai *työversio*. Työversioon kuuluvat kurssiversion tilat: *muokattava*, *suojattu*, *hylätty* ja *tarkastettava*.

public enum Tila { muokattava, suojattu, tarkastettava, hylätty, julkinen }

Tila on enum-tyyppi, joka voi saada arvokseen *muokattava*, *suojattu*, *tarkastettava*, *hylätty*, *julkinen*.

public enum Kieli { FI, EN }

Kieli on enum-tyyppi, joka voi saada arvokseen *FI* (suomi) tai *EN* (englanti).

public enum Taso { Lähestyy, Saavuttaa, Syventää }

Taso on enum-tyyppi, joka voi saada arvokseen *Lähestyy*, *Saavuttaa*, *Syventää*.

4.2 Malli

Mallissa on kolme luokkaa, jotka tarjoavat pääsyn tietokannan tietoihin ja viisi luokkaa tietojen välittämiseen. Useissa metodeissa pyydetään parametrina kutsuvan käyttäjän käyttäjätunnusta, jolla varmistetaan, että kutsuvalla taholla on oikeus kyseisiin operaatioihin. Rajapinta List löytyy paketista java.util ja muut projektin omasta paketista harri.

4.2.1 Kurssihallinta

public List<Kurssi> annaKurssit()

Palauttaa kaikki kurssit Kurssi-olioina List:n sisällä.

public List<Kurssi> annaKurssit(String hakujono)

Sama kuin edellä, mutta hakujono etsii vastaavuuksia kurssikoodista ja kurssin nimestä molemmilla kielillä. Hakujonossa ei ole erikoismerkkejä.

public List<Kurssi> annaKurssit(Versio versio)

Palauttaa kaikki kyseisen version kurssit Kurssi-olioina List:n sisällä.

public List<Kurssi> annaKurssit(String hakujono, Versio versio)

Sama kuin edellä, mutta hakujono etsii vastaavuuksia kurssikoodista ja kurssin nimestä molemmilla kielillä. Hakujonossa ei ole erikoismerkkejä.

public List<Kurssi> annaKurssit(String kayttaja)

Sama kuin edellä, mutta palauttaa kurssit tietyn käyttäjän tunnuksen perusteella. Kurssi lisätään listalle, jos käyttäjä on kurssin muokkaaja, vastuuhenkilö tai tarkastaja.

public Kurssi annaKurssi(String kurssiversio)

Palauttaa Kurssi-olion kurssiversion tunnuksen perusteella.

public List<Teema> annaTeemat(String kurssiversio)

Palauttaa kaikki kurssiversion teemat Teema-oliona List:n sisällä.

public Teema annaTeema(String tunnus)

Palauttaa Teema-olion teeman tunnuksen perusteella.

public List<Teema> lisaaTeemat(List<Teema> teemat)

Lisää tietokantaan teemoja listan sisältämien Teema-olioiden tietojen perusteella. Jos teema on tunnuksellaan jo tietokannassa, niin vanha korvataan. Jos Teema-oliossa teeman tunnus on null, niin lisätään uusi teema kantaan ja palautetaan olio takaisin uudella generoidulla tunnuksella varustettuna.

public void poistaTeemat(List<Teema> teemat)

Poistaa listan teemat tietokannasta.

public List<Esitietovaatimus> annaEsitietovaatimukset(String kurssiversio)

Palauttaa kaikki kurssiversion esitietovaatimukset Esitietovaatimus-olioina List:n sisällä.

public Esitietovaatimus annaEsitietovaatimus(String tunnus)

Palauttaa Esitietovaatimus-olion esitietovaatimuksen tunnuksen perusteella.

public List<Esitietovaatimus> lisaaEsitietovaatimus(List<Esitietovaatimus> esitie-

tovaatimukset)

Lisää tietokantaan esitietovaatimuksia listan sisältämien Esitietovaatimus-olioiden tietojen perusteella. Jos esitietovaatimus on tunnuksellaan jo tietokannassa, niin vanha korvataan. Jos Esitietovaatimus-oliossa esitietovaatimuksen tunnus on null, niin lisätään uusi esitietovaatimus tietokantaan ja palautetaan olio takaisin uudella generoidulla tunnuksella varustettuna.

public void poistaEsitietovaatimukset(List<Esitietovaatimus>)

Poistaa listan esitietovaatimukset tietokannasta.

public List<Oppimistavoite> annaOppimistavoitteet(String teema)

Palauttaa kaikki teeman oppimistavoitteet teeman tunnuksen perusteella Oppimistavoite-olioina List:n sisällä.

public Oppimistavoite annaOppimistavoite(String tunnus)

Palauttaa Oppimistavoite-olion oppimistavoitteen tunnuksen perusteella.

public List<Oppimistavoite> lisääOppimistavoitteet(List<Oppimistavoite> oppimistavoitteet)

Lisää tietokantaan oppimistavoitteita listan sisältämien Oppimistavoite-olioiden tietojen perusteella. Jos oppimistavoite on tunnuksellaan jo tietokannassa, niin vanha korvataan. Jos Oppimistavoite-oliossa oppimistavoitteen tunnus on null, niin lisätään uusi oppimistavoite tietokantaan ja palautetaan olio takaisin uudella generoidulla tunnuksella varustettuna.

public void poistaOppimistavoitteet(List<Oppimistavoite>)

Poistaa listan oppimistavoitteet tietokannasta.

public void lisääEsitietokurssi(String kurssiversio, String esitietovaatimus)

Lisää esitietokurssin tietokantaan kurssiversion tunnuksen ja esitietovaatimuksen tunnuksen perusteella.

public void poistaEsitietokurssi(String kurssiversio, String esitietovaatimus)

Poistaa esitietokurssin tietokannasta kurssiversion tunnuksen ja esitietovaatimuksen tunnuksen perusteella.

public void lisääEsitietoteema(String kurssiversio, String esitietovaatimus)

Lisää esitietoteeman tietokantaan kurssiversion tunnuksen ja esitietovaatimuksen tunnuksen perusteella.

public void poistaEsitietoteema(String kurssiversio, String esitietovaatimus)

Poistaa esitietoteeman tietokannasta kurssiversion tunnuksen ja esitietovaatimuksen tunnuksen perusteella.

4.2.2 Käyttajahallinta**public List<String> annaKayttajat(String kayttaja)**

Palauttaa kaikkien järjestelmän käyttäjien tunnuksat. Parametri kayttaja varmistaa, että kysyjä on rooliltaan vastuuhenkilö tai järjestelmävastaava.

public List<String> annaMuokkaajat(String kurssiversio)

Palauttaa kyseisen kurssin muokkaajien tunnukset listassa tai tyhjän listan, jos muokkajia ei ole.

public String annaVastuuhenkilö(String kurssiversio)

Palauttaa kurssin vastuuhenkilön tunnuksen.

public List<String> annaTarkastajat(String kurssiversio)

Palauttaa kyseisen kurssin tarkastajien tunnukset listassa tai tyhjän listan, jos tarkastajia ei ole.

public boolean onkoMuokkaaja(String käyttaja, String kurssiversio)

Palauttaa true, jos kyseinen käyttäjä kuuluu kurssin muokkaajiin.

public boolean onkoVastuuhenkilö(String käyttaja, String kurssiversio)

Palauttaa true, jos kyseinen käyttäjä on kurssin vastuuhenkilö.

public boolean onkoTarkastaja(String käyttaja, String kurssiversio)

Palauttaa true, jos kyseinen käyttäjä on kurssin tarkastaja.

public boolean onkoJarjestelmavastaava(String käyttaja)

Palauttaa true, jos kyseinen käyttäjä on järjestelmävastaava.

public void lisääMuokkaajat(String kurssiversio, List<String> muokkaajat)

Lisää tietokantaan listalla oleville käyttäjille muokkaan oikeudet haluttuun kurssiin.

public void muutaVastuuhenkilö(String kurssiversio, String vastuuhenkilö)

Muuttaa kurssin vastuuhenkilöä.

public void lisääTarkastajat(String kurssiversio, List<String> tarkastajat)

Lisää tietokantaan listalla oleville käyttäjille tarkastajan oikeudet haluttuun kurssiin.

public void poistaMuokkaajat(String kurssiversio, List<String> muokkaajat)

Poistaa kurssilta muokkajia.

public boolean poistaTarkastaja(String kurssiversio, List<String> tarkastaja)

Poistaa kurssilta tarkastajia. Parametri käyttäjä varmistaa metodin kutsujan oikeuden operaatioon.

4.2.3 Tilahallinta

public Tila annaKurssinTila(String kurssiversio)

Palauttaa kurssin tilan.

public void muutaKurssinTila(String kurssiversio, Tila tila)

Muuttaa kurssin tilaa. Tilat kuten edellä.

public boolean lukitseKurssi(String kurssiversio) Haluttu kurssi pyritään lukitsemaan käyttäjälle. Jos lukitseminen onnistuu, palautetaan true.

public void vapautaKurssi(String kurssiversio) Käyttäjä voi vapauttaa kurssin, jos hän on tietokannassa sen lukitsija. Vastuuhenkilö ja järjestelmävastaava voivat poistaa kurssin lukituksen, vaikka eivät olisi sitä itse lukinneet.

public void lahetäTarkastettavaksi(String kurssiversio)

Muuttaa kurssiversion tilan tarkastettavaksi.

public void hylkaaKurssi(String kurssiversio)

Muuttaa tarkastettavana olleen kurssiversion tilan hylätyksi ('h').

public void julkaiseKurssi(String kurssiversio)

Kahdentaa valitun kurssiversion ja sen kaikki teemat, esitietovaatimukset, oppimistavoitteet yms. tietokantaan ja luo kaikille omat uudet tunnukset. Todellinen mahtikäsky. Vain järjestelmävastaavan ja kurssin tarkastajan käytettävissä.

public void poistaKurssi(String kurssiversio)

Järjestelmävastaava voi poistaa tietokannasta kurssin, jota ei esim. enää opeteta. Tätä metodia tullaan käyttämään hyvin harvoin.

public void haeUudetKurssit()

Hakee OSJ:stä kurseja ja lisää uudet kurssit tietokantaan.

4.2.4 Selite

Selite on abstrakti luokka, jonka metodeilla hallitaan kuvauksia ja niiden eri kieliversioita. Perivät luokat ovat Kurssi, Teema, Oppimistavoite ja Esitietovaatimus.

public String annaKuvaus(Kieli kieli)

public void asetaKuvaus(String kuvaus, Kieli kieli)

4.2.5 Kurssi

Kurssi on luokka, jolla avulla välitetään kurssin (kurssiversion) tietoja. Luokka perii Selite-luokan. Tila on Enum-tyyppi, jonka arvona voi olla muokattava, suojattu, tarkastettava, hylätty tai julkinen.

public Kurssi(String tunnus, String tyoversionTunnus, String kurssikoodi, Tila tila, String vastuuhenkilo)

public String annaKurssikoodi()

public void asetaKurssikoodi(String kurssikoodi)

public Tila annaTila()

public void asetaTila(Tila tila)

public String annaTunnus(Versio versio)

public void asetaTunnus(String tunnus, Versio versio)

public String annaVastuuhenkilo()

public void asetaVastuuhenkilo(String vastuuhenkilo)

4.2.6 Teema

Teema on luokka, jolla välitetään teeman tietoja. Luokka perii Selite-luokan. Teema-olioon kapseloidaan pääteema-alateema-oppimistavoite -rakenne.

```
public Teema(String kurssiversio)  
public Teema(String tunnus, String kurssiversio)  
public String annaKurssiversio()  
public void asetaKurssiversio(String kurssiversio)  
public String annaTunnus()  
public void asetaTunnus(String tunnus)  
public List<Oppimistavoite> annaOppimistavoitteet()  
public void asetaOppimistavoite(Oppimistavoite oppimistavoite)  
public List<Teema> annaAlateemat()  
public void asetaAlateema(Teema alateema)
```

4.2.7 Esitietovaatimus

Esitietovaatimus on luokka, jolla Malli välittää esitietovaatimusten tietoja. Luokka perii Selite-luokan.

```
public Esitietovaatimus(String kurssiversio, int syvyys)  
public Esitietovaatimus(String tunnus, String kurssiversio, int syvyys) Jos tunnus on  
null, esitietovaatimusta pidetään uutena ja tietokantaan viettäessä sille generoidaan tunnus.  
public String annaKurssiversio()  
public void asetaKurssiversio(String kurssiversio)  
public int annaSyvyys()  
public void asetaSyvyys(int syvyys)  
public String annaTunnus()  
public void asetaTunnus(String tunnus)  
public List<String> annaEsitietokurssit()  
public void asetaEsitietokurssi(String esitietokurssi)  
public List<String> annaEsitietoteemat()  
public void asetaEsitietoteema(String esitietoteema)
```

4.2.8 Oppimistavoite

Oppimistavoite on luokka, jolla Malli välittää oppimistavoiteen tietoja. Luokka perii Selite-luokan.

public Oppimistavoite(String tunnus, String teema, Taso taso, int syvyys, Date muutettu, String muuttaja)

public Oppimistavoite(String teema, Taso taso, int syvyys, Date muutettu, String muuttaja) Jos tunnus on null, oppimistavoitetta pidetään uutena ja tietokantaan viettäessä sille generoidaan tunnus.

public String annaTunnus()

public void asetaTunnus(String tunnus)

public String annaTeema()

public void asetaTeema(String teema)

public Taso annaTaso()

public void asetaTaso(Taso taso)

public int annaSyvyys()

public void asetaSyvyys(int syvyys)

public Date annaMuutettu()

public void asetaMuutettu(Date muutettu)

public String annaMuuttaja()

public void asetaMuuttaja(String muuttaja)

4.3 Näkymän JSP-sivut

Käyttöliittymäsuunnittelussa esiteltiin yhdeksän eri käyttöliittymä komponenttia, joita yhdistelemällä saadaan rakennettua käyttäjälle www-sivuja. Jokaisesta komponentista tehdään oma JSP-sivunsa. Tallennukseen liittyvät operaatiot käsitellään omilla JSP-sivuillaan.

index.jsp on aloitussivu, joka kokoaa järjestelmän perusnäkökuvan eli seuraavat sivut: haku.jsp, pikavalinta.jsp.

haku.jsp sivulla käyttäjä voi syöttää hakusanan, joka välitetään sivulle kurssilistaus.jsp.

pikavalinta.jsp sivulla käyttäjä voi valita *omat kurssit* tai *kaikki kurssit*. Valinta välitetään sivulle kurssilistaus.jsp.

kurssilistaus.jsp sivulla käyttäjä voi valita listatuista kursseista, joko julkaistun tai työversion. Valinta välitetään sivulle kurssi.jsp.

kurssi.jsp sivulla käyttäjä voi muuttaa kurssin tilaa. Muutos käsitellään samalla sivulla. Käyttäjä voi myös siirtyä seuraaville JSP-sivuille: oppimistavoitteet.jsp, esitietovaatimukset.jsp tai henkilöhallinta.jsp.

oppimistavoitteet.jsp sivulla käyttäjä voi siirtyä kurssin muokkaustilaan, eli otmuokkaus.jsp sivulle.

otmuokkaus.jsp sivulla pääteemoja, alateemojen ja oppimistavoitteiden lisääminen ja poistaminen kutsuvat otmuokkaus.jsp sivua itseään. Tallentaminen käsitellään sivulla ottallennus.jsp.

ottallennus.jsp käsittelee oppimistavoitteiden tallennuksen ja palauttaa käyttäjän sivulle oppimistavoitteet.jsp.

esitietovaatimukset sivulla käyttäjä voi siirtyä esitietovaatimusten muokkaustilaan, eli etmuokkaus.jsp sivulle.

etmuokkaus.jsp sivulla esitietovaatimusten lisääminen ja poistaminen kutsuvat etmuokkaus.jsp sivua itseään. Tallentaminen käsitellään sivulla ettallennus.jsp.

ettallennus.jsp käsittelee esitietovaatimusten tallennuksen ja palauttaa käyttäjän sivulle esitietovaatimukset.jsp.

henkilohallinta.jsp sivun tallennus käsitellään hhtallennus.jsp:ssä.

hhtallennus.jsp käsittelee henkilöhallinnan tallennuksen ja palauttaa käyttäjän sivulle oppimistavoitteet.jsp.

4.4 Ohjain

Kaikki tiedon päivitys hoidetaan ohjaimen kautta. Kaikissa metodeissa parametria *String kayttaja* käytetään tarkistamaan, onko kutsujalla tarvittavat oikeudet kurssiin.

public void lisaaOppimistavoitteet(String kayttaja, String kurssiversio, List<Teema> teemat)

Kutsuu mallin metodeita siten, että parametreina annetut pääteemat, alateemat ja oppimistavoitteet lisätään tai muutetaan kyseisen kurssin tauluissa.

public void poistaOppimistavoitteet(String kayttaja, String kurssiversio, List<Teema> teemat)

Kutsuu mallin metodeita siten, että parametreina annetut pääteemat, alateemat ja oppimistavoitteet poistetaan kyseisen kurssin tauluista.

public void lisaaEsitietovaatimukset(String kayttaja, String kurssiversio, List<Esitietovaatimus> esitietovaatimukset)

Kutsuu mallin metodeita siten, että parametreina annetut esitietovaatimukset ja niiden liitokset lisätään tai muutetaan kyseisen kurssin tauluissa.

public void poistaEsitietovaatimukset(String kayttaja, String kurssiversio, List<Esitietovaatimus> esitietovaatimukset)

Kutsuu mallin metodeita siten, että parametreina annetut esitietovaatimukset ja niiden liitokset poistetaan kyseisen kurssin tauluista.

public void talletaKayttajat(String kayttaja, String kurssiversio, List<String> muokkaajat, List<String> tarkastajat, String vastuuhenkilö)

Kutsuu mallin metodeita siten, että parametreina annettujen käyttäjien oikeudet päivite-

tään kyseisen kurssin tauluihin.

public void muutaKurssinTila(String kayttaja, String kurssiversio, Tila tila)

Kutsuu mallin metodia, joka muuttaa kurssin tilan parametrina annetun tilan mukaiseksi.

public void poistaKurssi(String kayttaja, String kurssiversio)

Järjestelmävastaava voi poistaa tietokannasta kurssin, jota ei esim. enää opeteta. Tätä metodia tullaan käyttämään hyvin harvoin.