

KYMENLAAKSON AMMATTIKORKEAKOULU

Liiketalous/ Tietojenkäsittely

Mika Hatakka

MODULAARISEN JÄRJESTELMÄN LAAJENTAMINEN -
MOODLEN KURSSIEN ARKISTOINTILOHKO

Opinnäytetyö 2010

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Tietojenkäsittely

HATAKKA, MIKA

Modulaarisen järjestelmän laajentaminen -

Moodlen kurssien arkistointilohko

Opinnäytetyö

38 sivua + 42 liitesivua

Työn ohjaaja

Lehtori Päivi Hurri

Toimeksiantaja

Oppimisteknologiakeskus Kymiedu

Huhtikuu 2010

Avainsanat

oppimisympäristö, arkistointi, ohjelmointi, PHP

Laajennettavuus on yksi merkityksellisimmistä ominaisuuksista varsinkin avoimen lähdekoodin sovelluksissa ja järjestelmissä. Modulaarisuus antaa laajennettavuuteen selkeän toteutusmallin. Digitaalisen materiaalin arkistointi on toinen tärkeä tämän hetken kysymys. Digitaalista materiaalia tuotetaan aina vain enemmän mutta arkistoinnin tarpeeseen on vasta viime aikoina alettu kiinnittää huomiota. Tässä työssä yhdistetään laajennettavuus ja arkistointi, Moodle-oppimisympäristöön toteutettavan kurssien arkistointijärjestelmän kehittämiseksi.

Työn tavoitteena oli toteuttaa kurssien arkistointijärjestelmä Moodle-oppimisympäristöön. Oppimisteknologiakeskus Kymiedun ylläpitää Moodle-oppimisympäristöä Kymenlaakson ammattikorkeakoululle. Kurssien määrän jatkuva kasvu aiheuttaa tilanteen, jossa kurssien suuri määrä tekee paikoin oikean kurssin löytämisen haasteelliseksi. Arkistointijärjestelmä mahdollisti opetusalan selkiyttämisen siirtämällä vanhentuneet kurssit arkistoon, tietoja kuitenkin hävittämättä.

Työn toiminnallisena osana toteutettiin Moodleen kurssien arkistointijärjestelmä laajennusmoduulina. Raportissa tutustutaan järjestelmän suunnitteluun vaikuttaneisiin perusteisiin ja käydään läpi työn etenemisen vaiheet.

Modulaarisen rakenteen omaavat sovellukset ja järjestelmät antavat mahdollisuuden laajennusmoduulien toteuttamiseen. Työn aikana tuli hyvin ilmi, että mitä parempi on laajennettavan järjestelmän tuntemus, sitä helpompi on saada aikaan laadukas lopputulos. Toteutettavasta laajennuksesta on oltava tarkat määritykset ja on myös ymmärrettävä, mitä toimeksiantaja odottaa laajennukselta.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Business Information Technology

HATAKKA, MIKA

Expanding modular system -

Moodle course archive block

Bachelor's Thesis

38 pages + 42 pages of appendices

Supervisor

Päivi Hurri, Senior Lecturer

Commissioned by

The learning technology centre Kymiedu

April 2010

Keywords

learning environment, archive, programming, PHP

Expandability is one of the most important characteristics particularly in open source applications and systems. Modularity provides a clear model for expandability. Archiving digital material is another important issue at the moment. Digital materials are produced more and more, but have only recently begun to draw attention with the need for archiving. In this work these two are combined with Moodle, in the development of a course archive system.

The goal was to make a course archiving system to the Moodle learning environment. The Learning technology centre Kymiedu maintain a Moodle learning environment for the Kymenlaakso University of Applied Sciences. The increasing number of courses creates a situation in which a large number of courses will sometimes make it difficult to find the right course. A course archive system makes it possible to decrease the number of courses by moving the obsolete courses to the archive, without destroying any data.

The empirical research was producing the Moodle course archive system. The report introduces the system design criteria and goes through the production steps.

The applications or systems with modular structure allows for easy expansion. During the work it became very clear that the better the knowledge of the system is, the easier it is to achieve high quality results. Accurate specifications and understanding of what the client expects for the extension are also crucial.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	6
2	MODULAARISET JÄRJESTELMÄT	7
2.1	Yleistä	7
2.2	Modulaarisuuden kehitys	8
2.3	Moduulien tunnistaminen ja liittäminen	10
2.4	Rajapintojen tarjoamat mahdollisuudet	11
2.5	Uudelleenkäyttö	12
3	MOODLE-OPPIMISYMPÄRISTÖ	12
3.1	Yleistä	12
3.2	Moodlen rakenne	13
3.3	Aktiviteetit	14
3.4	Lohkot	15
3.5	Moodlen kurssijärjestelmä ja perusjärjestelmän hallinta	16
4	ARKISTOINTI	17
4.1	Yleistä	17
4.2	Arkistointi Moodlen näkökulmasta	18
5	ARKISTOINTIJÄRJESTELMÄ	19
5.1	Lähtökohta	19
5.2	Arkistointijärjestelmän toimintojen kartoitus	20
5.3	Suunnittelun lähtökohtia	20
5.4	Tietokanta	21
5.5	Lohkonäkymät	23
5.6	Arkistointijärjestelmän ikkunanäkymät	25
5.7	Kurssien arkistointi	26
5.7.1	Arkistointipyyntö	26
5.7.2	Varsinainen arkistointi	27

5.8	Kurssin palautus arkistosta	29
5.8.1	Kurssin haku arkistosta	29
5.8.2	Arkistosta palautus	31
6	SAAVUTETUT EDUT JA HAITAT	33
7	KEHITYSMAHDOLLISUUDET	34
7.1	Yleistä	34
7.2	Kehitysmahdollisuudet arkistointijärjestelmään	34
7.3	Kehitysmahdollisuudet Moodleen	35
8	JOHTOPÄÄTÖKSET	36
	LÄHTEET	38
	LIITTEET	
	Liite 1. Arkistointilohkon käyttöohje	
	Liite 2. Arkistointilohkon tekninen kuvaus	
	Liite 3. Arkistointilohkon lähdekoodi, CD-ROM	

1 JOHDANTO

Tämän päivän jatkuvasti kehittyvässä maailmassa erilaisten tietotekniikkaan perustuvien järjestelmien laajentaminen on tullut yhä merkittävämmäksi. Järjestelmien rakenteen tulee mukautua helposti uusien ominaisuuksien lisäämiseen. Modulaarinen rakenne antaa järjestelmän jatkokehitykselle ja lisäominaisuuksien toteuttamiselle hyvät puitteet. Järjestelmän laajentaminen ei ole mahdollista ilman sen toteuttajan antamaa mahdollisuutta laajentumiselle. Näiltä osin avoimeen lähdekoodiin perustavat järjestelmät ovat kulkeneet kehityksen eturintamassa ja vaikuttaneet perinteisen suljetun järjestelmänkehityksen suhtautumisen muuttumiseen avoimempaan suuntaan. Onhan perinteisesti ajateltu suljetun lähdekoodin olevan ainoa mahdollisuus taloudellisesti tuottavaan ohjelmistojen kehitykseen. Tämän osalta avoimen laajennettavat lähdekoodin projektit ovat tuoneet kelpo vaihtoehdon suljetun koodin ohjelmistokehitykselle.

Modulaariset järjestelmät ovat minulle jo entuudestaan tuttuja sulautettujen järjestelmien yhteydestä. Olen ollut mukana kehittämässä teollisuuden elektronista laskuri-perhettä, jossa modulaarisella rakenteella pyrittiin ohjelmiston mahdollisimman suureen uudelleen käyttöön. Oli haastavaa päästä siirtämään aiempia kokemuksia täysin uuteen ympäristöön. Aiemmasta kokemuksesta oli paljon hyötyä varsinkin suunnitteluvaiheessa, mitä tarkemmin suunnittelun pystyy toteuttamaan, sitä vaivattomammin varsinainen toteutusvaihe sujuu. Oli helpompaa paneutua suunnitteluvaiheeseen tietäen, että hyvä suunnittelu palkitsee varsin nopeasti itsensä.

Oppimisteknologiakeskus Kymiedu ylläpitää Kymenlaakson ammattikorkeakoulun Moodle-oppimisympäristöä. Oppimisympäristöä käytetään aktiivisesti opiskeluun normaalin lähiopetuksen tukena ja erityisesti verkko-opiskelun työalustana. Oppimisympäristöön on käytön aikana kertynyt suuri määrä kurseja ja Kymiedulla oli tarve kurssien arkistointijärjestelmälle. Minulle tarjoutui mahdollisuus paneutua modulaarisen järjestelmän laajentamiseen ja toteuttaa arkistointijärjestelmä Moodleen.

Moodle oli tuttu opiskelijan näkökulmasta ja työharjoittelun aikana olin jo tutustunut ylläpitopuoleen. Ohjelman kehityksen kannalta Moodle oli minulle kuitenkin täysin vieras, joten tutustuminen Moodlen kehitystyöhön tarjoi mahdollisuuksiin ja vaatimuksiin vaatisivat pelkästään jo huomattavan työmäärän. Kurssien arkistointijärjestelmä on myös itsessään vaativa ja monisyinen toteutettava, joten kokonaisuutena tehtävän työmäärä muodostui kunnioitettavan kokoiseksi. Arkistointijärjestelmän osalta

opinnäytetyön ohjaajani, Riikka Räikkönen Oppimisteknologiakeskus Kymiedusta, pystyi antamaan selkeät määrittelyt järjestelmän tarpeista ja näin auttamaan suunnittelun alusta alkaen oikealle suunnalle.

Dokumentti ei ole ohje modulaarisen järjestelmän laajentamiseen, vaan pyrkii luomaan kuvan huomioitavista asioista ryhdyttyä toteuttamaan laajennusta. Eri järjestelmät poikkeavat toisistaan suuresti ja käsillä oleva toteutus antaa jokaiseen projektiin oman yksilöllisen etenemistien. Jokainen projekti on tie uuden oppimiseen.

2 MODULAARISET JÄRJESTELMÄT

2.1 Yleistä

Modulaarisuus on määritelty toiminnallisina yksiköinä, joilla on sisäinen tila ja määritelty rajapinta ulospäin (Clark, Baldwin 1999, 88). Ohjelmistotekniikassa modulaarisuus ymmärretään ohjelman jakamisena kokonaisuuksiin, jotka tuntevat oman tilansa ja joilla on selkeä rajapinta, jonka kautta ohjelma hallitsee kokonaisuutta. Rajapinta on joukko sovelluksen määrittelemiä aliohjelmiä, joiden avulla moduulit hoitavat kanssakäymisen muun sovelluksen kanssa. Ilman rajapintojen määrittelyä ei sovellusta ole mahdollista laajentaa ilman, että puututtaisiin sovelluksen lähdekoodiin. Moduuli kattaa yhden toiminnallisen ominaisuuden kokonaisuudesta. Moduuli voi ja usein koostuu useammasta pienemmästä kokonaisuudesta. Moduulia on vaikea määrittellä tarkasti, koska jokainen kerta muodostaa aina oman kokonaisuutensa ja ainutkertaisen tarpeensa.

Modulaarisella rakenteella tavoitellaan ohjelmistojen helppoa laajennettavuutta, yksinkertaisempaa ylläpitoa ja moduulien uudelleenkäyttöä. Kaikkien ominaisuuksien tavoittelemisen on varsin epärealistista, joten helpon ylläpidon lisäksi tulisi keskittyä joko laajennettavuuteen tai uudelleenkäyttöön. Laajennettavuus ja uudelleenkäyttö on mahdollista toteuttaa yhdessä, mutta se vaatii todella tarkkaa suunnittelua.

Ohjelmiston hyvin toteutettu modulaarinen rakenne antaa mahdollisuuden ohjelman helppoon laajentamiseen ja ylläpitoon. Kun jokainen moduuli pitää itse huolen tarvitsemastaan tiedosta ja kanssakäyminen muiden moduulien kanssa suoritetaan määrittelyn rajapinnan kautta, luodaan tilanne, jossa moduulin sisäistä rakennetta ja sen sisältämiä tietorakenteita voidaan muuttaa. Rajapinta on kuitenkin säilytettävä ennallaan.

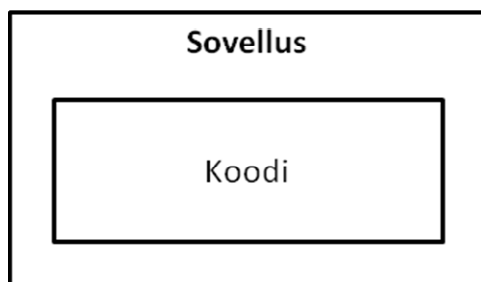
Rajapintaa voidaan laajentaa lisäämällä uusia aliohjelmiä. Ylläpidon kannalta tietenkin optimaalinen tilanne on se, jossa rajapinta pystytään määrittelemään niin hyvin jo alkuperäisessä toteutusvaiheessa, ettei tarvetta rajapinnan laajentamiseen ole.

Yksi suuria vaikuttavia tekijöitä rajapintojen ja modulaarisuuden kehittymiseen oli Microsoft Windows 3 käyttöjärjestelmän julkistaminen. Windows 3:n API-rajapinta ja käyttöliittymätason modulaarisuus toi asian kaikkien tietoon. Tähän asti sovelluskehityksessä oli toimittu muutamien aliohjelmakirjastojen ja sovelluksia tuottavien yritysten omien aliohjelmakirjastojen varassa. Nyt Microsoft antoi sovellusten toteuttamiseen rajapinnan, jonka kautta ohjelmien perustoiminnot toimivat samalla periaatteella ja näin siirtyminen ohjelmasta toiseen ei vaadi suurta uuden opettelua.

Tänä päivänä hyvinkin suositut web-palvelut (web services) tuovat myös mukaan mahdollisuutensa sovellusten laajentamiseen. Se voidaanko web-palvelut lukea mukaan modulaariseen ohjelmarakenteeseen, onkin toinen kysymys. Web-palvelut ovat verkon kautta tarjolla olevia palveluita ja tästä syystä en itse laskisi niitä mukaan modulaariseen ohjelmarakenteeseen.

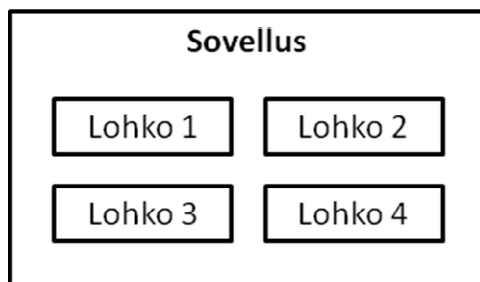
2.2 Modulaarisuuden kehitys

Ohjelmistokehitys on alkanut rakenteesta, jossa kaikki sovelluksen koodi on yhtenä suurena yksikkönä, (kuva 1). Uutta sovellusta aloitettaessa ei vanhoista ole pystytty käyttämään uudestaan kuin pieniä palasia. Samoin tarve sovelluksen laajentamiseen on ollut vaikeaa ja helpompi tie on ollut kokonaan aloittaa alusta. Modulaarisen ohjelmarakenteen ajatus alkoi nousta esiin jo 1970-luvun loppupuolella (Haikala & Märijärvi 2004, 308).



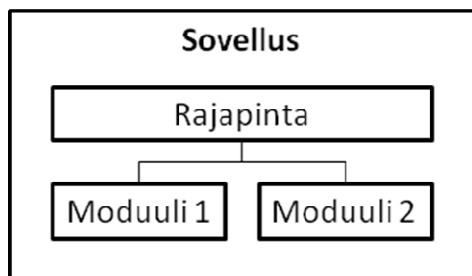
Kuva 1. Sovelluskohtainen lähdekoodi

Ensin sovellukset jaettiin sisäisesti lohkoiksi, joilla oli oma tehtävänsä (kuva 2). Tällaisella jaottelulla saavutettiin ensimmäinen askel kohti laajennettavuutta ja uudelleenkäyttöä. Ohjelman rakenteen selkiytyessä oli helpompi laajentaa ohjelmaa, kun tiedettiin minkä lohkon osalle laajennus haluttiin. Sovelluksen koodin uudelleenkäyttökin oli jo jossain määrin mahdollista ja uuden sovelluksen pohjaksi voitiin valita soveliaat lohkot. Varsinaisesta modulaarisesta rakenteesta ei vielä tässä vaiheessa voitu puhua, koska lohkoille ei ollut määritelty pysyvää rajapintaa, jonka kautta niitä käsiteltiin.



Kuva 2. Sovellus jaettu sisäisiin lohkoihin

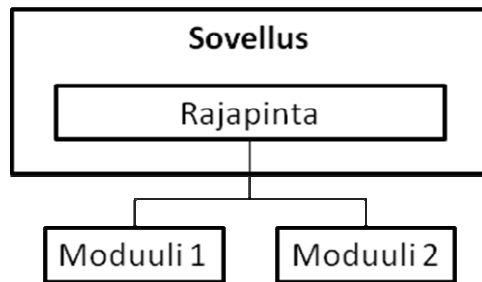
Seuraavassa kehitysvaiheessa luotiin muuttumaton rajapinta ohjelmalohkojen välille, (kuva 3). Sovellukselle luotiin sisäinen rajapinta, jonka kautta sovellusta voitiin laajentaa. Näin muutokset sovellukseen rajoittuivat pienelle alueelle ja laajennettu versio saatiin nopeammin käyttäjille. Moduulin sisäiset muutokset ja korjaukset eivät myöskään aiheuttaneet muutoksia sovelluksen muuhun toimintaan. Rakenteen etuja oli myös muodostunut tilanne, jossa samasta perusrakenteesta voidaan helposti muodostaa eri versioita sovelluksesta. Näin toimien pystytään hyödyntämään sovelluksissa samaa yhteistä runkoa ja erikoistaa kukin sovellus sen ominaisella osuudella. Tätä mallia hyödynnetään muun muassa sulautetuissa järjestelmissä.



Kuva 3. Sisäinen modulaarinen rakenne

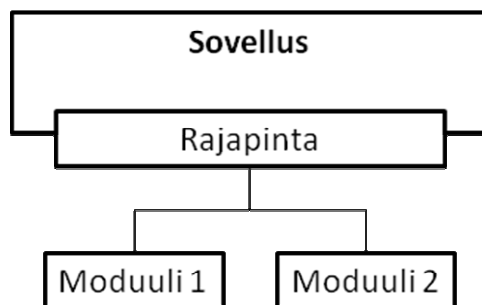
Tämän jälkeen nousi esiin mahdollisuus jakaa rajapinta myös muille kehittäjille. Varsin nopeasti ajatusta kehitettiin eteenpäin ja rajapinta päätettiin antaa myös muiden

kehittäjien käyttöön. Suljetun lähdekoodin sovelluksissa oli mahdollista luoda sisäinen rajapinta, joka on pelkästään toteuttaneen tahon tiedossa tai se voidaan avata maksua vastaan (kuva 4). Ensimmäisten joukossa tämän hyödyntäjinä suljetun lähdekoodin sovelluksista olivat pelivalmistajat, jotka antoivat pelaajille mahdollisuuden laajentaa pelien pelikenttiä.



Kuva 4. Suljettu rajapinta dynaamisella laajennettavuudella

Viimeisenä vaiheena oli antaa rajapinta kokonaan kaikkien halukkaiden käytettäväksi, (kuva 5). Tämä on mahdollista toteuttaa niin suljetun kuin avoimen lähdekoodin sovelluksissa, mutta avoimen lähdekoodin toteutuksissa koko lähdekoodin avoimena olemisen antaa vielä suuremman mahdollisuuden laajennettavuuteen. On mahdollista muokata sovellusta vastaamaan paremmin omia tarpeita ja luoda itse toteutettuihin laajennuksiin omia rajapintoja antamaan mahdollisuuksia uusille laajennuksille.



Kuva 5. Avoin modulaarinen rakenne

2.3 Moduulien tunnistaminen ja liittäminen

Toinen perusedellytys laajentumiselle on moduulin saattaminen sovelluksen käytettäväksi. Uudet moduulit voidaan ottaa käyttöön suoraan tai lisätä odottamaan käyttöön-ottoa. On myös mahdollista etsiä moduuleja Internetistä ja tuoda ne sovelluksen käyttöön. Sovelluksen on tunnistettava moduuli, jotta sen antamat laajennusominaisuudet voidaan hyödyntää. Moduulin tunnistaminen tapahtuu yleisimmin sovelluksen tarkas-

taessa määritellyssä hakemistossa tapahtuneita muutoksia. Tämä tapahtuu sovelluksen käynnistyessä tai web-sovellusten kohdalla tietty sivu ladattaessa. Seuraavassa vaiheessa käyttäjä valitsee käytettävät moduulit, jonka jälkeen moduulit asennetaan sovelluksen käyttöön.

On myös mahdollista tunnistaa ja ottaa käyttöön käytettävissä olevat moduulit automaattisesti. Tämä voidaan toteuttaa etsimällä dynaamisesti laajennusmoduuleja sovelluksen hakemistoista tai käyttäen tehdasmenetelmä, joka on sopiva tarkoituksiin, joissa ei tiedetä, mikä nimenomainen moduuli otetaan käyttöön sovelluksessa. Moduulit toteuttavat toiminnallisuuden tarvitsemat aliohjelmat ja näiden lisäksi oman erityisen aliohjelman moduulin luontia varten. Menetelmää käytetään esimerkiksi liityttäessä tietokannanhallintajärjestelmiin, tällöin käytettävä järjestelmä voidaan valita aina kulloisenkin tilanteen tai asiakkaan mukaan. Microsoftin COM-tekniikka käyttää tämän tyyppistä ratkaisua komponenttien liittämiseen sovellukseen. Siinä menetelmää on vielä viety siinä määrin eteenpäin, että sovellus voi kysyä komponentilta, tukeeko se sovelluksen tarvitsemaan rajapintaa. Näin vältetään turhilta virhetilanteilta, kun jo valmiiksi tiedetään mitä ominaisuuksia komponentti tukee.

2.4 Rajapintojen tarjoamat mahdollisuudet

Tarjotun rajapinnan perusteella voidaan rajoittaa, miten moduulien on mahdollista laajentaa sovellusta. Tarjottaessa suppea rajapinta, rajoitetaan moduulin vaikutusmahdollisuudet. Sovellukselle jää näin valta määrittää, miten paljon moduuli pääsee vaikuttamaan toimintaan. Moduulin toiminta voi olla laaja, mutta vaikutukset ja näkyvyys ovat rajatut ja säädellyt. Laaja rajapinta antaa tarttumapintaa sovellukseen ja vaikutusmahdollisuuksia on enemmän. Moduulilla on mahdollisuus päästä kiinni sovelluksen toimintoihin monissa oleellisissa kohdissa. Rajapinta voidaan myös toteuttaa tapahtuma-aliohjelmina, joiden avulla päästään kiinni sovelluksen tapahtumiin juuri ennen tai jälkeen niiden suorituksen.

Rajapinnan ollessa tarkasti määritelty ja liittyen sovelluksen tapahtumiin, mahdollistaa se helpon kiinnittymisen järjestelmän toimintoihin. Moduulilla on mahdollisuus vaikuttaa suoraan sovelluksen toimintaan ennen ja jälkeen tapahtumien. Tämä antaa moduulille monia erilaisia toimintamalleja hyödyntää sovelluksen tietoa ja muokata tiedon esitysmuotoja. Rajapinnan tulee tässä mallissa olla tarkkaan harkittu ja tulee huomioida, mitä tapahtumia annetaan rajapinnan käyttöön. Vastavuoroisesti moduulit

voivat tuottaa oman rajapinnan muiden moduulien käyttöön ja näin edelleen laajentaa sovelluksen käyttömahdollisuuksia. Tarkasti määritelty rajapinta rajoittaa vapaata hyödyntämistä täysimääräisesti, vastapainona kuitenkin hyvä laajennettavuus.

2.5 Uudelleenkäyttö

Modulaarinen rakenne mahdollistaa myös moduulien uudelleenkäytön muissa ohjelmistoprojekteissa. Uudelleenkäyttö nousee aina uudestaan esiin tasaisin väliajoin. Moduulien uudelleenkäytön huomioon ottaminen ohjelmistoprojektissa on varsin monisyinen asia. Yritettäessä luoda uudelleenkäytettäviä moduuleja yhdessä laajennettavuuden kanssa, törmätään nopeasti tilanteeseen, jossa kulutetaan aikaa yrittäessä ottaa huomioon uudelleenkäytön vaatimuksia moduulia suunnitellessa ja toteutettaessa (Lehtimäki 2006, 169). Projektien aikataulujen ollessa perustilanteessakin jo varsin tiukkoja, uudelleenkäytön huomioiminen aiheuttaa lähes varmasti aikataulun ylityksen. Kun uudelleenkäytettävää moduulia otetaan käyttöön omaan projektiin, huomataan kohta, ettei se kuitenkaan sovellu suoraan, vaan siihen on tehtävä projektikohtaisia muokkauksia.

Moduulien uudelleenkäyttö on toki tärkeä kohdealue. On vain huomattava, ettei se onnistu tai on äärimmäisen hankalaa ja harvinaista ohjelmistoprojektin ohessa. Uudelleenkäytettävät moduulit on parasta toteuttaa omassa projektissa, jossa luodaan aliohjelmista kokoelma toteuttamaan halutut toiminnot. Tällöin uudelleenkäytön vaatimuksia on helpompi lähestyä yleiseltä tasolta, ilman oman projektin vaatimuksia. Yleisellä tasolla toteutettaessa rajapinnasta on helpompi muodostaa kaikkia palveleva ja projektikohtaiset toiminnot jäävät toteutettavaksi kulloisenkin projektin sisällä.

3 MOODLE-OPPIMISYMPÄRISTÖ

3.1 Yleistä

Moodle on oppimisympäristö verkon kautta tapahtuvaan opiskeluun (kuva 6). Siitä käytetään myös nimityksiä oppimisalusta, kurssin hallintajärjestelmä (Course Management System, CMS), opetuksen hallintajärjestelmä (Learning Management System, LMS) tai virtuaalinen oppimisympäristö (Virtual Learning Environment, VLE).

The screenshot shows the Moodle homepage for Kymenlaakson ammattikorkeakoulu. The header includes the university logo and the text 'Moodle-oppimisympäristö'. A central message area reads: 'Tervetuloa Moodle-oppimisympäristöön. Kymenlaakson ammattikorkeakoulun oppimisympäristö on päivitetty uuteen versioon 23.6.2009. KyAMK:n opiskelijat ja henkilökunta voivat kirjautua Moodleen työasemien verkkotunnuksella (kuva Novell Login -ikkunasta). VirtuaaliAMK:n tai Avoimen AMK:n kautta opiskelevat saavat käyttäjätunnuksensa kurssin alkaessa sähköpostitse. Kirjautumisongelmissa ota yhteyttä Helpdeskiin osoitteessa <http://helpdesk.kyamk.fi> tai sähköpostitse moodle@kyamk.fi. Welcome to Moodle eLearning environment. Moodle of Kymenlaakso University of Applied Sciences is updated to a new version on 23rd of June 2009. Kymenlaakso UAS students and staff will sign in to Moodle with Novell user accounts that are same user ids which are used to log in to our network computers (picture of Novell Login window). If you have problems in logging in to Moodle, please read the "Contact information" page from Main Menu for further information.' The left sidebar has two main sections: 'Päävalikko' with links like 'Moodle-ohjeet opettajille', 'Moodle-ohjeet opiskelijoille', 'Etsi kursseja', 'Kurssikategoriat', 'Yhteyshenkilöt', 'Päivystys', 'Moodle instructions for teachers', 'Moodle instructions for students', 'Search courses', 'Course categories', and 'Contact information'; and 'Kurssikategoriat' with a list of subjects like 'Yhteiset perusopinnot', 'Kielioopinnot', 'Matemattis-luonnontieteelliset opinnot', 'Tekniikka, metsä ja liikenne', and 'Kansainvälinen liiketoiminta ja kulttuuri'. The right sidebar contains a 'Kirjautu' login form with fields for 'Käyttäjätunnus' and 'Salasana', and a 'Kirjautu' button. The top right corner shows 'Et ole kirjautunut. (Kirjautu)' and a language dropdown set to 'Suomi (fi)'.

Kuva 6. Kymenlaakson ammattikorkeakoulun Moodlen etusivu

Moodlen kehityksen aloitti Martin Dougiamas, joka Curtinin yliopiston WebCT-ympäristöä ylläpitäessään totesi sen olevan rakennettu työkalujen ympärille (Cole 2005, 5). Dougiamas halusi järjestelmän, joka lähtisi pedagogisesta näkökulmasta. Järjestelmän tulee muotoutua opetuksen mukaan eikä päinvastoin. Moodlen perusfilosofiana on sosiaalinen konstruktionismi (Moodle 2009). Siinä opetus sosiaalisena tahtumana rakentuu opiskelijoiden laatimien vastausten jakamiseen kaikille kurssilla olevien saataville ja yhteisissä keskusteluista saatuun palautteeseen. Tässä opettajan roolin ei ole tarkoitus jäädä perinteiseen rooliin, vaan hänen tulee myös ohjata ja hillitä keskustelua kursseilla.

Moodlen ensimmäinen versio julkaistiin vuonna 2002 ja kehitystyö jatkuu edelleen, versio 1.9.6 julkaistiin lokakuussa 2009. Tällä hetkellä työtä tehdään monella rintamalla, vanhoja versioita ylläpidetään ja uuden Moodle 2.0 version parissa työtä on jo tehty pitkään. Beta versio Moodle 2.0 on tarkoitus saada jakeluun huhtikuussa ja lopullinen versio heinäkuussa 2010. Moodlen rekisteröityneitä yhteisöjä löytyy 209 maasta ja se on käännetty 78 kielelle (lokakuu 2009).

3.2 Moodlen rakenne

Alusta alkaen Moodlea rakennettiin täysin avoimena ratkaisuna ja modulaarisena toteutuksena. Ydintoimintojen ympärille toiminnot on lisätty moduuleilla, näin noudat-


taen modulaarisuuden periaatteita. Moduulit ovat toiminnallisia kokonaisuuksia, jotka lisäävät toiminnon Moodleen. Toiminto kokonaisuus saattaa vaatia useita moduuleja toimiakseen, mutta perustilanteessa yksi toiminto koostuu yhdestä moduulista. Avoinmuutta noudattaen kaikkien halukkaiden toteuttamat moduulit on otettu vastaan. Kehityksestä vastaava ryhmä Dougiamasin johdolla päättää, milloin ja mitkä uusista moduuleista liitetään ytimeen eli perusasennukseen kuuluvat moduulit.

Moodle rakentuu kurssien ympärille ja moduulit on jaettu kahteen ryhmään, aktiviteetteihin ja lohkoihin. Aktiviteeteillä lisätään uusia oppimateriaalin käsittelytapoja tai opetus toteutuksia ja lohkoilla säädellään tai ilmaistaan Moodlen järjestelmän tilaa. Lohkot sijoittuvat näkymässä oikealle ja vasemmalle laidalle ja sisältävät lisätietoa kurssin aktiviteeteistä tai ohjaavat suoraan muualle Moodleen (Karevaara 2008, 18). Osa lohkoista on järjestelmänvalvojan asettamia, näitä opettajat eivät voi poistaa, ja osa opettajien asettamia. Aktiviteetit ovat välineet varsinaisen opetusaineiston liittämiseen kurssille ja ne sijaitsevat käyttöliittymässä keskellä. Aktiviteetit tarjoavat opettajalle monen tyyppisiä vaihtoehtoja kurssin toteuttamiseen, valittavissa on esimerkiksi: keskustelupalsta, tekstisivu, tehtävä, tentti ja koko ajan kasvava joukko muita aktiviteettejä. Opettaja määrittelee ne aktiviteetit, joita haluaa kurssillansa käyttää.

3.3 Aktiviteetit

Kurssi rakennetaan Moodleen aktiviteeteilla, joiden avulla oppimateriaalia tuodaan opiskelijoiden saatavilla. Aktiviteetteja on useamman tyyppistä, ja yhdessä ne muodostavat kattavan valikoiman työkaluja, joilla opettaja voi muodostaa erilaisia lähestymistapoja opetukseen. Perustapana voidaan pitää oppimateriaalin lisäämistä kurssille, josta opiskelijat voivat käydä sen lukemassa sopivaksi katsomanaan ajankohtana. Kun tähän lisätään keskustelualue, niin on jo saavutettu vuorovaikutus mahdollisuus opiskelijoiden kesken. Tehtävien palautus yhtenä tai useampana tiedostona täydentää opettajan perustyökaluvalikoiman.

Aktiviteettien avulla opettajan on mahdollista toteuttaa kurssi, jossa asiassa edetessä voidaan seurata opiskelijoiden edistymistä ja jopa vaihdella oppimateriaalin tarjontaa edistymisen mukaan. Kaikkea materiaalia ei tarvitse antaa kerralla, vaan sen voi ajoittaa edistymisen mukaan. Tenttityökalulla on mahdollista luoda kurssille välitenttejä, jotta opiskelijat voivat seurata kehittymistään ja arvosanaan vaikuttava lopputentti onnistuu samalla työkalulla (kuva 7).

3	Yhdistä maat ja pääkaupungit.						
Pistettä: 1	<table> <tr> <td>Ruotsi</td> <td>Valitse...</td> </tr> <tr> <td>Tanska</td> <td>Valitse...</td> </tr> </table>	Ruotsi	Valitse...	Tanska	Valitse...		
Ruotsi	Valitse...						
Tanska	Valitse...						
4	Mitkä seuraavista ovat tentin kysymysvaihtoehtoja?						
Pistettä: 1	Valitse ainakin yksi vastaus <table> <tr> <td><input type="checkbox"/></td> <td>a. Tehtävä</td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td>b. Monivalinta</td> </tr> <tr> <td><input type="checkbox"/></td> <td>c. Lasku</td> </tr> </table>	<input type="checkbox"/>	a. Tehtävä	<input checked="" type="checkbox"/>	b. Monivalinta	<input type="checkbox"/>	c. Lasku
<input type="checkbox"/>	a. Tehtävä						
<input checked="" type="checkbox"/>	b. Monivalinta						
<input type="checkbox"/>	c. Lasku						
5	Mitä lyhennettä käytetään Kymenlaakson ammattikorkeakoulusta?						
Pistettä: 1	Vastaus: <input type="text"/>						
6	Paljonko on piin arvo?						
Pistettä: 1	Vastaus: <input type="text"/>						
7	Suomi itsenäistyi 1917?						
Pistettä: 1	Vastaus: <table> <tr> <td><input type="radio"/></td> <td>Tosi</td> </tr> <tr> <td><input type="radio"/></td> <td>Epätosi</td> </tr> </table>	<input type="radio"/>	Tosi	<input type="radio"/>	Epätosi		
<input type="radio"/>	Tosi						
<input type="radio"/>	Epätosi						
<input type="button" value="Tallenna ilman palauttamista"/> <input type="button" value="Palauta kaikki ja lopeta"/>							
 Tämä sivu Moodle Docs -sivustolla							

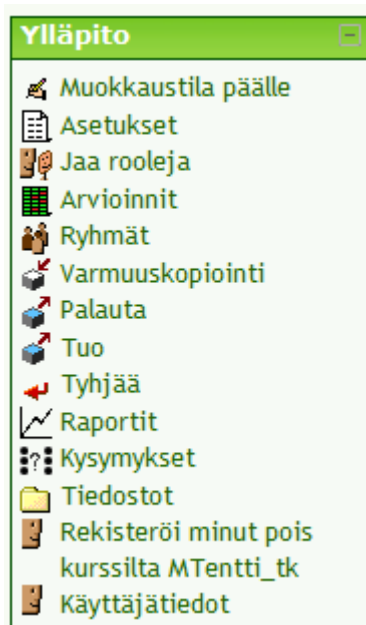
Kuva 7. Tentti-aktiiviteetti

Lisäksi tarjolla on monia yhteisöllisiä vaihtoehtoja, kuten chatit ja wikit. Oppiaineen huomioivia aktiiviteetteja löytyy myös tukemaan esimerkiksi kielten opiskelua, johon on tarjolla nauhurityökalu, jonka avulla on mahdollista seurata omaa edistymistä tai toimittaa opettajalle harjoitusten palautuksia. Valmiita aktiiviteetteja löytyy Moodlen kotisivuilta suuri määrä ja uusia on koko ajan työn alla. Tästä pitää huolen suuri joukko innostuneita Moodlen käyttäjiä, joilla on myös taito luoda uusia aktiiviteetteja. Eikä mikään estä rakentamasta tai teettämästä juuri omaan tarpeeseen sopivaa aktiiviteettiä.

3.4 Lohkot

Lohkot muodostavat merkittävän osan Moodlen toiminnallisuutta. Lohkojen avulla kaikki käyttäjät pystyvät omien oikeuksien puitteissa seuraamaan ja vaikuttamaan Moodlen toimintaan. Opettajille ja opiskelijoille lohkot kertovat heihin liittyviä järjes-

telmän tietoja, kuten omat kurssit. Kurssikohtaisesti näytetään viimeisimmät uutiset ja tapahtumat. Kalenterista voidaan seurata tulevia tapahtumia yleisellä, kurssi tai henkilökohtaisella tasolla. Opettajalle tärkein lohko on kurssin ylläpitolohko (kuva 8), jonka avulla hän pystyy muokkaamaan kurssiin liittyviä tietoja. Lohkot ovat peruskäyttäjien väylä vuorovaikutukseen Moodlen kanssa.



Kuva 8. Kurssisivun ylläpito lohko

Lohkoilla on myös mahdollista toteuttaa opiskelua tukevia toimintoja. Lohkojen kautta käytetään blogeja ja blogi-tunnisteita, portfolioon siirrytään lohkon kautta. Lohkoilla voi näyttää tenttien tuloksia ja siirtyminen verkon kautta toiseen Moodleen tai esimerkiksi Second Life virtuaali-maailmaan on toteutettu lohkoilla.

Järjestelmänvalvoja määrittelee muille käyttäjille näkyvät lohkot. Lohkot voidaan asettaa pakollisiksi, jolloin käyttäjät eivät voi niitä poistaa tai näkyviksi, jolloin ne voi halutessaan poistaa omasta kurssinäkymästä. Lohkojen poisto on sallittu muista käyttäjistä vain opettajille, ja he saavat lohkot myös takaisin näkyviksi kurssisivuilla. Lohkojen yleiset asetukset ovat järjestelmänvalvojan, mutta joillakin lohkoilla on asetuksia, jotka ovat kurssikohtaisia ja näitä opettajat pääsevät muokkaamaan.

3.5 Moodlen kurssijärjestelmä ja perusjärjestelmän hallinta

Moodlen kurssijärjestelmän perusyksikkö on kurssi, joka on sijoitettu kurssikategoriaan. Kurssikategoriat muodostavat hierarkkisen puun, johon kaikki kurssit sijoittuvat.

Kurssikategoriat ovat oppilaitoskohtaisesti määriteltävissä, eikä niitä ole mitenkään rajoitettu. Kurssilla on joukko asetuksia, jotka järjestelmänvalvoja asettaa kurssia luodessaan, näihin kuuluvat kurssin nimi, kuvaus kurssista ja opettaja. Opettaja voi jälkepäin muuttaa kurssin tietoja. Opiskelijat voidaan lisätä kurssille käsin tai he voivat käydä itse liittymässä kurssille, joka on suositeltavin tapa. Opiskelijat voivat myös halutessaan poistaa itsensä kurssilta. Kun kurssi on luotu, voi opettaja aloittaa oppimateriaalin lisäämisen kurssille.

Järjestelmänvalvojalla on hallittavanaan myös suuri joukko Moodlen toiminnan kannalta oleellisia asetuksia. Nämä ovat kuitenkin lähes poikkeuksetta kertaalleen asetettavia. Järjestelmänvalvojan pääasialliset toimet keskittyvät kurssien ja käyttäjien hallintaan sekä järjestelmän toiminnasta kertovien raporttien seuraamiseen. Näistä kurssien hallinta ja ylläpito muodostavat suurimman osan työstä.

4 ARKISTOINTI

4.1 Yleistä

Arkistoinnin merkitys kasvaa koko ajan, kun samalla tuotetaan yhä enemmän digitaalista materiaalia. Tuotettua materiaalia on saatava talletettua siten, että siihen pystytään vaivattomasti palaamaan ja saada haluttu tieto käytettäväksi. Tällä hetkellä arkistointi ei ole saanut tarvitsemaansa huomiota ja monet erheellisesti kuvittelevat varmistuksen tarkoittavan samaa asiaa. Varmistus ja arkistointi omaavat kuitenkin eriävät tavoitteet.

Varmistuksessa tarkoituksena on taata toiminnan jatkuminen ja arkistoinnissa materiaalista erotellaan ydinosa ja taltioidaan se pysyvästi, (Järvinen 2009, 26). Varmistuksessa talletetaan yleensä kaikki saatavilla oleva tieto ja sen palautus tehdään samassa muodossa kuin se oli talletettu. Varmistustallenteen osalta toivotaan, ettei sitä tarvitsisi koskaan käyttää. Arkistoinnissa olennaista on oleellisen tiedon tallettaminen ja talletetusta tiedosta on saatava käsiteltäväksi haluttu tieto sopivan kokoisena pakettina. Ei ole kuitenkaan tarkoituksenmukaista olettaa, että saisi vain sen tiedon jota tarvitsee, vaan palautettava otos on laajempi kokonaisuus, josta halutut tiedot täytyy itse erottaa. Arkistoitu tieto on talletettu käytettäväksi ja jää odottamaan käyttöä.

Toinen arkistoinnin kannalta tärkeä, mutta usein pienelle huomiolle jäävä asia on, miten ja minkälaiselle medialle arkistointi suoritetaan. Tallennusmedian valinta ei ole aivan yhdentekevä ratkaisu, arkistoista ei paljon iloa jos niitä ei pystytä lukemaan. Tässä suhteessa paperiarkisto on vielä edellä, paperille taltioitua tietoa pystyy lukemaan vielä vuosisatojen päästä. Tietotekniikassa tallennusmediat kehittyvät hurjalla vauhdilla ja tänä päivänä 5 ¼ tuuman levykkeen (lerpun) tietojen lukeminen vaatii jo hieman työtä, että löytää sopivan levykeaseman.

Aivan yhtä tärkeä asia arkistoinnin kannalta on arkistojen säilytys, kirjoitus säilyy kylmä paperillakin, jos vain paperia ei altisteta tuhoutumisvaaraan. Tallennusmedioiden osalta tilanne ei ole paljoakaan muuttunut, ne vaativat oikeanlaiset säilytysolosuhteet, jotta ne säilyvät. Tämän päivän tallennusmediat ovat oikeastaan herkempiä tuhoutumiselle verrattuna paperiin, pienikin naarmu CD-levyn pinnassa saattaa tehdä siitä lukukelvottoman.

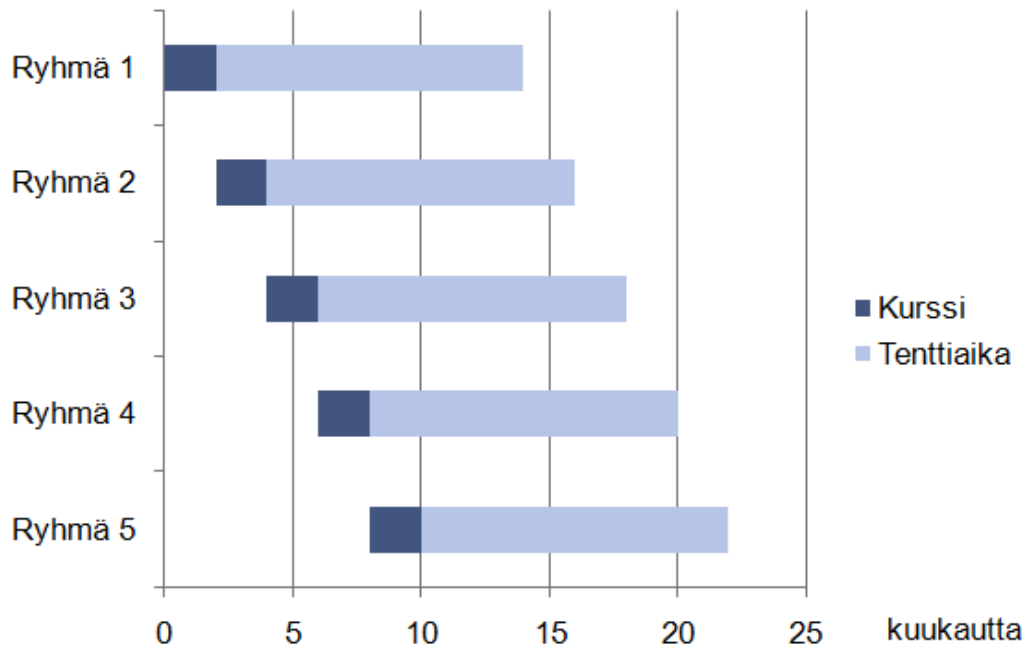
4.2 Arkistointi Moodlen näkökulmasta

Moodlen kohdalla arkistointi kohdistuu kursseihin. Kurssit sisältävät opettajan oppimateriaalin ja opiskelijoiden palauttamien harjoitukset ja työt. Opiskelijoiden palauttamien töiden on oltava tarkasteltavissa niin kauan kuin heidän opiskelunsa on kesken. Aina tasaisin välein tulee tilanteita, joissa opiskelijan kurssisuoritus on varmistettava ennen kuin hän voi valmistua. Opettajien materiaalit ovat yleensä heillä itsellään tallessa, mutta niidenkin osalta voi tulla tarve hakea vanhoista kurseista.

Kurssien arkistoinnilla tarkoitetaan Moodlen osalta toimintatapaa, jossa kurssit siirretään pois varsinaisesta oppimisympäristöstä varastoitavaksi siten, että kursseihin on helppo tarvittaessa palata. Moodlen ydinpakettiin ei kuulu varsinaista arkistointi toimintaa. Kurssien taltiointi ja palautus täytyy tällä hetkellä tehdä normaaleilla varmistusvälineillä.

Arkistointi tarpeeseen vaikuttavat ensinnäkin koko oppimisympäristöön kertyvä kurssien kokonaismäärä ja toisaalta yksittäisen opettajan omien kurssien määrä saattaa nousta vaikeasti käsiteltäväksi. Opettajan kurssimäärää kasvattaa ammattikorkeakoulun kurssijärjestelmä, jossa oppilailta on vuosi kurssin päättymisestä aikaa suorittaa tentti tai korottaa arvosanaa, (Kymenlaakson ammattikorkeakoulun tutkintosääntö 2009 - 2010 2009, 16). Opettajalle, joka opettaa kurssin jokaisessa lukuvuoden jaksos-

sa ja jokaiselle ryhmälle tarvitaan oma kurssinsa Moodle-oppimisympäristössä, on nopeasti melkoinen kurssi määrä hallittavana. Kuva 9 selventää, miten kurssit ja niitä seuraava tenttimiskelpoinen jakso vievät aikaa ja näin vaativat oppilaiden Moodle-oppimisympäristöön palauttamien tehtävien olevan saatavilla.



Kuva 9. Kurssien käytettävissä oloaika

Kymenlaakson ammattikorkeakoulun kurssimäärä on jatkuvassa kasvussa. Moodle oppimisympäristön kurssiseurannasta katsottuna on kurssien määrä vuoden 2009 aikana kasvanut noin 1000 kurssilla. Uusien kurssien määrä voi olla tätäkin suurempi, koska osa vanhoista kursseista on poistettu. Kurssien arkistoinnille on todellinen tarve.

5 ARKISTOINTIJÄRJESTELMÄ

5.1 Lähtökohta

Arkistoinnin puuttuessa Moodlen perusominaisuuksista ja tarve arkistoinnille oli olemassa, aloitettiin ratkaisun kartoitus olemassa olevista lisämoduuleista. Lisämoduuleista ei kuitenkaan löytynyt tarkoituksiin sopivaa laajennusta, joten ratkaisua täytyi etsiä toisaalta. Varteenotettavaksi vaihtoehdoksi ilmeni Moodle Network -liityntä, jossa erilliset Moodle-toteutukset ovat yhteydessä toisiinsa verkon välityksellä. Vaihtoehtoa tutkittaessa havaittiin sen antavan mahdollisuuksia arkistoinnin suorittamiseen, mutta lopulta muutamat piirteet aiheuttivat tämänkin vaihtoehdon hylkäämisen. Oleel-

lisin hylkäämiseen vaikuttava tekijä oli, ettei käyttäjätunnuksia saada pysymään samoina kummassakin Moodle-toteutuksessa. Tämä taas aiheuttaa sekaantumisen vaaran haettaessa arkistosta tietyn opiskelijan suorituksia. Näiden tutkimusten perusteella tulimme siihen tulokseen, että oman arkistointijärjestelmän toteutusta olisi myös järkevää tutkia.

5.2 Arkistointijärjestelmän toimintojen kartoitus

Oman arkistointijärjestelmän toteutuskartoitus aloitettiin määrittelemällä arkistoon talletettavat tiedot sekä tarvittavat hakuehdot. Kurssi koostuu opettajan materiaalista, opiskelijoiden tehtävän palautuksista, käyttäjätiedoista sekä arvostelutiedoista, kurssin perusasetusten lisäksi. Nopeasti todettiin, että paras ratkaisu oli tallettaa kaikki kurssin tiedot arkistoon. Tietoja ei määrällisesti ole suurta määrää ja toisen kautta olisi ollut vaikea rajata jokin tietty osa-alue pois. Hakuehdot aiheuttivat enemmän pohdintaa, olihan niillä suuri vaikutus arkiston käyttöön. Mukaan päätettiin ottaa kurssin nimi ja lyhyttunnus, arkistointiaika sekä opettajat ja opiskelijat. Hakuehtojen kohdalla päädyttiin myös pitämään varaus ehtojen mahdolliseen muuttamiseen.

Seuraavaksi tuli miettiä käyttöliittymä vaihtoehdot, joita tarkemman harkinnan jälkeen jäi vain Moodlen lohko. Alussa lohkoja päätettiin toteuttaa kaksi, järjestelmänvalvojalle ja opettajalle omansa. Ajatuksena oli pitää näiden erilaiset toimet selkeästi erillään. Järjestelmänvalvojan lohkoon tulisivat arkistoinnin aloitus ja kurssien etsintä arkistosta. Opettajan lohkoon taas haluttiin arkistointipyynnön lähetys ja arkistoinnin opastus.

Tässä vaiheessa projektin mahdollisuuksia tarkasteltiin yhdessä toimeksiantajan edustajan, Riikka Räikkösen kanssa ja pohdittiin ratkaisun toteuttamismahdollisuuksia. Koska kaikki halutut ominaisuudet oli saatu mukaan toteutusmalliin, eikä kartoituksen aikana ilmennyt toteutuksen kohtuuttomia ongelmia kohtia, päätettiin aloittaa projekti arkistointijärjestelmän toteuttamiseksi.

5.3 Suunnittelun lähtökohtia

Arkistointijärjestelmän tulisi olla helppokäyttöinen, niin opettajan kuin järjestelmänvalvojan kannalta. Opettajan osalta tähän on yksinkertaista päästä toiminnon ollessa pyynnön lähetys. Järjestelmänvalvojan osalta toiminto sarjoihin täytyi kiinnittää huo-

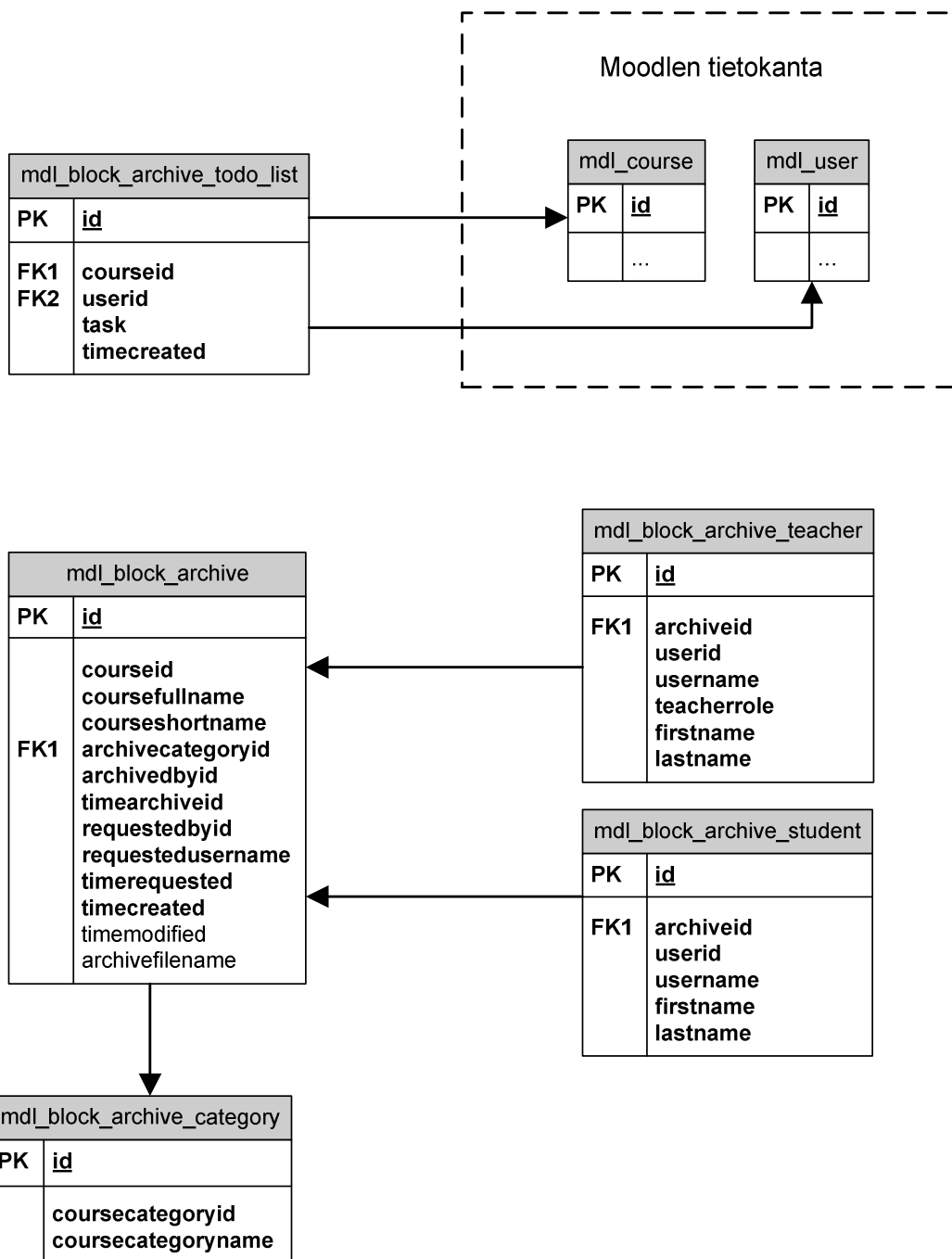
miota, jotta toimet saatiin suoritettua mahdollisimman vähin askelin. Yhdellä lomakkeella ei kuitenkaan toimintoja pystytä hoitamaan. Järjestelmä pyrittiin toteuttamaan mahdollisimman suuressa määrin Moodlen vakiokomponentteja käyttäen, näin säilyttäen yhteensopivuus myös tulevissa versioissa. Tämä oli toinen tekijä, jonka vuoksi arkistointiin metodina päätettiin käyttää Moodle varmistus- ja palautustoimintoja. Toinen tärkeä seikka oli se, että näin toimien arkistosta palautus on mahdollista myös ilman arkistointijärjestelmää.

Suunnittelun aikana tarkentuivat myös järjestelmään tulevat toiminnot, opettajalle tuli arkistointipyynnön lisäksi kurssin poistopyynnön lähetysmahdollisuus. Lisäksi täytyi huomioida kurssit, joilla on useampia opettajia, joissa kaikilta tulisi saada arkistointi toimeen hyväksyntä. Lisäksi haluttiin välittää lohkon kautta tietoa arkistoinnin tilasta sekä ilmoitus opettajille, kun arkistointi on suoritettu.

5.4 Tietokanta

Ensimmäisenä suunniteltiin ja toteutettiin arkistointijärjestelmän tietokanta, sen ollessa keskeisessä roolissa koko järjestelmän kannalta. Tietokantaan tallennetaan tiedot arkistoitavista kursseista ja arkistossa olevista kursseista (kuva 10). Osat ovat toiminnaltaan ja tavoitteiltaan erilaiset, ja nämä erot tuli huomioida tietokantaa suunniteltaessa. Arkistoitavia kursseja sisältävä osa on lähinnä operatiivinen, kun taas arkistoiduista kursseista tietoja tallentava osuus sisältää operatiivisia, mutta myös tietovarastotyyppisiä ominaisuuksia. Moodlen asettamat ohjeet ja vaatimukset tietokannalle täytyi myös ottaa huomioon suunnittelussa.

Arkistoitavat kurssit sisältävään tietokanta osaan kuuluu yksi taulu. Tauluun kerätään perustiedot arkistointipyynnöstä, kurssin id-tunnus, arkistoinnin pyytäjän id-tunnus ja valittu toimenpide. Näillä tiedoilla pystytään aloitettaessa arkistointia hakemaan Moodlen tietokannoista kaikki arkistoinnin kannalta vaadittavat tiedot. Näistä pystytään myös tekemään tarvittavat tarkastukset, onko muilta opettajilta pyydettävä suostumusta tai soviteltava ristiriitatilanteessa, jossa yksikin opettajista on kieltänyt arkistoinnin. Arkistointi toimenpiteiden jälkeen pyynnöt poistetaan taulusta, perustilanteen ollessa se, että arkistointipyynnöttaulu on tyhjä.



Kuva 10. Kurssien arkistointilohkon tietokantataulut

Arkistoitujen kurssien tietokantaosuus sisältää neljä taulua. Taulujen avulla on pyritty muodostamaan mahdollisimman joustava kokoonpano, jotta kurssien etsintä arkistosta olisi yksinkertaista ja nopeaa. Arkistoiduille kurseille on varsinainen päätaulu ja opettajille, opiskelijoille sekä kurssikategorioille ovat omat taulunsa. Kurssikategorioiden osalta tällaiseen ratkaisuun päädyttiin, koska kurssikategoria saatetaan poistaa ja nyt palautus pystytään suorittamaan alkuperäisen mukaiseen kategoriaan. Onhan varsin mahdotonta päätellä, mihin uuteen kategoriaa palautettava kurssi tulisi sijoittaa. Nyt järjestelmänvalvojan tehtäväksi jää kurssin sijoittaminen oikeaan kurssikategori-

aan. Opettajien ja opiskelijoiden kohdalta sijoitustapaa harkittiin omien taulujen ja varsinaisen arkistotaulun välillä. Omiin tauluihin päädyttiin kuitenkin, jotta välttyään turhalta toistolta. Hakujen kannalta varsinaiseen arkistotauluun sijoittaminen olisi ollut parempi ratkaisu, mutta koska kyseessä on kuitenkin arkisto eikä tietovarasto, hakuja tietokantaan ei tehdä jatkuvasti. Levytilan säästö on tässä tapauksessa parempi etenemistapa.

Varsinaiseen arkistotauluun on kerätty tiedot kurssista, arkistoinnin suorittajasta ja pyytäjistä sekä näiden tapahtumahetkistä ja tietenkin varsinaisen arkistotiedoston nimi. Kurssin ja arkistoinnin pyytäjän tietoja on otettu mukaan sekä id-tunnisteena että teksti muodossa. Näin on helpotettu hakujen suoritusta, että oleelliset hakuavaimet löytyisivät arkistotietokantataulusta. Muutoin vaihtoehtona olisi tietojen hakeminen arkistotiedostosta, joka olisi vaativa ja hidaskäyttöinen suorittaa. Opettaja- ja opiskelijatauluissa on myös käytetty tietovarastotyylistä lähestymistapaa, käyttäjätunnus ja nimitietoja on sisällytetty mukaan tekstimuodossa hakujen nopeuttamiseksi. Kategoriataulussa on tiedot arkistoitujen kurssien arkistointiaikaisista kategorioista id-tunnisteista ja kategorianimet.

Tietokantojen luonti suoritetaan Moodlella omalla XMLDB-editorilla. Editori löytyy Moodlen järjestelmänvalvojan työkaluista. Editorin käyttö on selkeää ja tuloksena syntyvä XML-tiedosto, install.xml, muodostaa yleisselitteisen asennustiedoston Moodlelle. Näiden ohjeiden pohjalta muodostetaan tietokantataulut, riippumatta käytetystä tietokannanhallintajärjestelmästä. Näin ohjelmankehittäjän ei tarvitse huolehtia eri tietokannanhallintajärjestelmien eroista luodessaan tauluja omien lohkojen tai aktiviteettien käyttöön.

5.5 Lohkonäkymät

Suunniteltu kahden lohkon toteutus havaittiin nopeasti hankalaksi, ja se monimutkaisti rakennetta tarpeettomasti, joten molemmat käyttäjien toiminnot päätettiin toteuttaa yhdellä loholla. Seuraavaksi pohdittiin, pystytäänkö selkeä ja yksinkertainen toimintojen hallinta suorittamaan yhdellä yhteisellä näkymällä. Tämä osoittautui varsin nopeasti mahdolliseksi, mikä johtuu järjestelmänvalvojan ja opettajan toimintojen eroavaisuuksista. Omien näkymien suunnittelu järjestelmänvalvojalle ja opettajille oli paras vaihtoehto. Järjestelmänvalvojan näkymä on näkyvissä vain sivuston etusivulla ja opettajien näkymä kurssisivulla.

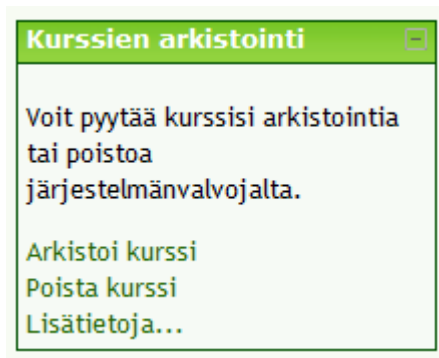
Järjestelmänvalvojan näkymä sisältää tiedon arkistointia odottavista kursseista ja linkit arkistoinnin aloittamiseen, mikäli arkistoitavia kursseja on odottamassa ja hakuun arkistoiduista kursseista (kuva 11). Lisäksi näytetään arkistoitujen kurssien kokonaislukumäärä. Järjestelmänvalvojalla on myös oma näkymänsä arkistointilohkoon kurssisivulla, mikäli hän ei ole opettajana kurssilla. Normaalien toimintojen lisäksi järjestelmänvalvojalle annetaan mahdollisuus lisätä kurssi pakotetusti arkistoitavien kurssien joukkoon.



Kuva 11. Järjestelmänvalvojan lohkonäkymät

Kurssisivulla lohkonäkymä sisältää opettajille arkistointi- ja poistopyynnön lähetyksen linkit (kuva 12). Järjestelmänvalvojan ollessa opettajana kurssilla, on hänellä myös mahdollisuus esittää pakotettu arkistointipyynnö. Näkymässä näytetään myös tiedot esitetystä arkistointipyynnöstä ja useamman opettajan kurssilla myös tieto, kuka pyynnön on esittänyt sekä hyväksyntä- ja kiello-vahvistukset. Arkistointipyynnön esittäjälle ja hyväksyjälle annetaan ilmoituksen lisäksi mahdollisuus peruuttaa pyyntö. Peruutusmahdollisuus on voimassa niin kauan, kun arkistointia ei ole aloitettu. Mikäli usean kurssin opettajista joku esittää arkistoinnille kielloa, lukittuu kurssin arkistointitila ristiriitatilanteeseen ja kiellon esittäjälle annetaan mahdollisuus perua kiello, muille näytetään vain tiedotus tilanteesta. Toissijaiselle opettajalla näytetään tiedonanto arkistoinnintilasta, mikäli arkistointia on pyydetty. Muussa tilanteessa näytetään ohjeistus arkistoinnin pyytämiseen ja arkistoinnista vastaavan sähköpostiosoite, johon pyynnöt tulee osoittaa.

Opettajien lohkoissa on myös Lisätietoja-linkki, josta löytyy opettajille ohjeistus. Ohjeissa käydään lyhyesti läpi arkistointilohkon perustoiminnot. Ohjesivu näytetään sekä normaaleille, että toissijaisille opettajille. Ohjenäyttö on toteutettu Moodlen normaalilla ikkuna-näkymällä.



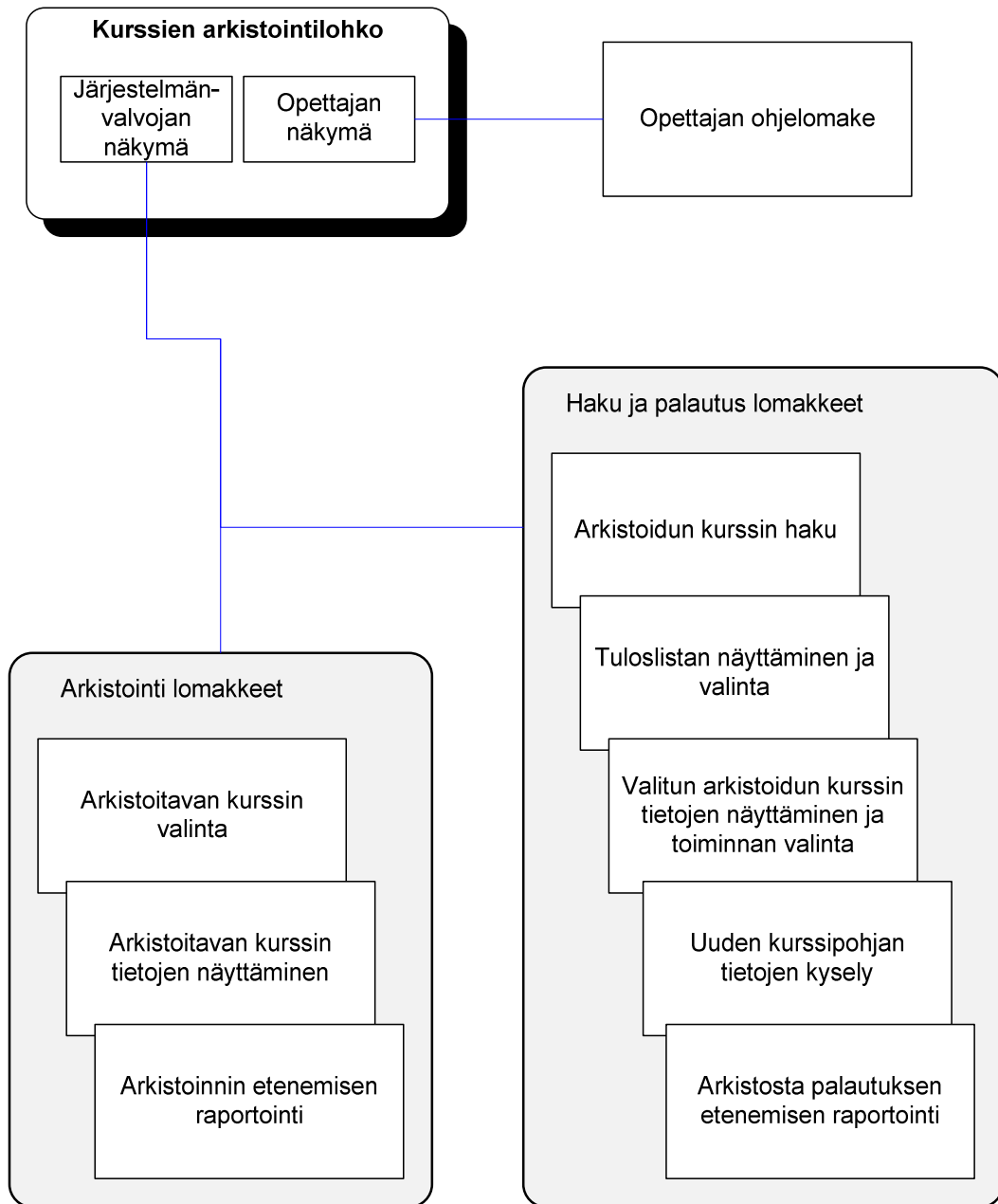
Kuva 12. Opettajan lohkonäkymä

Lohkojen toimintaa Moodlen `block_base`-luokkaan, joka muodostaa lohkoille toiminnallisuudet. Luokka tarjoaa rajapinnan, jonka kautta muodostetaan yhteys Moodleen ja saadaan lohko näkyviin sekä ohjaus toiminnot. Jokaisessa lohossa tulee ylikirjoittaa vähintään metodit: `init` ja `getContent`, mutta tarjolla myös muita toimintoja helpottavia metodeja.

5.6 Arkistointijärjestelmän ikkunanäkymät

Arkistointijärjestelmän arkistointi- ja palautustoiminnot on toteutettu hyödyntäen Moodlen lomakkeita. Lomakeluokasta periyttiin oma lomakeluokka, johon lisättiin järjestelmän tarvitsemat erikoisominaisuudet, kuten kirjautuneena olon tarkistus, käyttäjällä riittävät oikeudet toimintoihin sekä ulkoasun yhteiset osuudet. Moodlea on mahdollista käyttää ilman sisään kirjautumista, mutta arkistointi vaatii kirjautumisen. Siksi jo tietoturvan kannalta on välttämätöntä tarkastaa, että käyttäjä on kirjautunut sisään Moodleen. Toinen tietoturvan kannalta oleellinen tarkistus on riittävien käyttöoikeuksien tarkastus. Arkistointi on järjestelmän kannalta kriittinen kohta, joten se on sallittu vain järjestelmänvalvojan oikeuksin.

Ikkunanäkymät toistavat tiettyjä osia joka lomakkeella, joten on tarkoituksen mukaista liittää perusluokkaan, josta varsinaiset lomakkeet periytetään. Näin saadaan kaikille yhteiset ylä- ja alatunnisteet sekä leivänmururivi varmasti jokaiselle lomakkeelle, ja ne myös näyttävät samanlaisilta. Raportti-ikkunoiden reaaliaikainen näytön päivitys on myös toteutettu perusluokassa. Kuva 13 esittelee arkistointijärjestelmän näkymät ja ikkunat sekä niiden väliset suhteet.



Kuva 13. Arkistointijärjestelmä näkymät ja niiden käytettävissä olevat lomakkeet

5.7 Kurssien arkistointi

5.7.1 Arkistointipyyntö

Opettajan lähettämä arkistointipyyntö käynnistää koko arkistointiprosessin. Pyyntö talletetaan tietokantaan, josta tieto päivitetään järjestelmänvalvojan etusivulla olevaan lohkonäkymään. Vastaavasti mikäli kurssilla on useita opettajia, päivitetään heidänkin lohkonäkymään tieto arkistoinnin tilasta. Heille annetaan myös mahdollisuus hyväksyä tai kieltää arkistointitoimi ja tämäkin tieto talletetaan tietokantaan. Toissijaisten opettajien osalta lohkon yleiset asetukset määräävät kuinka tietoa näytetään heille.

Vaihtoehtoja on kolme: arkistoinnista ei kerrota heille mitään, näytetään arkistoinnin tila ja sähköpostiosoite, johon voi pyynnön välittää tai kohdellaan kuten normaalia opettajaa. Arkistointipyynnö ei vielä aiheuta mitään muutoksia kurssiin, se vain kirjaa tiedon tietokantaan.

5.7.2 Varsinainen arkistointi

Kurssien arkistointi aloitetaan siirtymällä arkistoitavan kurssin valintaan. Kurssin valintasivulla on pudotusvalikko arkistointia odottavista kursseista (kuva 14). Sivun toteutus selkeyden saavuttamiseksi omana kokonaisuutena, eikä yhdessä valitun kurssin tietojen näytön kanssa. Jos valinta olisi toteutettu samalle sivulle valitun kurssin tietojen näytön kanssa, olisi ollut mahdollista unohtaa päivittää näkymä ja aloittaa arkistointi väärälle kurssille.

The screenshot shows the Moodle interface for course archiving. At the top, there is a header with the logo of Kymenlaakson ammattikorkeakoulu (University of Applied Sciences) and the text 'Moodle-oppimisympäristö'. Below the header, there is a green navigation bar with 'EvoMoodle ► Kurssien arkistointi' and a user status indicator 'Olet kirjautunut nimellä Ylläpito Käyttäjä. (Kirjaudu ulos)'. The main content area is titled 'Kurssien arkistointi' and contains a section 'Arkistoitavat kurssit'. Under this section, there is a dropdown menu for 'Lähetetyt pyynnot' with the selected option 'FOK007 - Funktionaalinen ohjelmointi'. Below the dropdown are two buttons: 'Näytä valitun kurssin tiedot' and 'Peruuta'. At the bottom of the main content area, there is a small information icon and the text 'Tämä sivu Moodle Docs -sivustolla'. The footer of the page is a green bar with 'EvoMoodle ► Kurssien arkistointi'.

Kuva 14. Arkistoitavan kurssin valinta

Kun arkistoitava kurssi on valittu, näytetään sen tiedot. Valitusta kurssista näytetään perustiedot ja pyynnön esittäjä sekä haluttu toimenpide. Kaikki kurssin opettajat listataan ja näytetään ovatko he hyväksyneet tai kieltäneet kurssin arkistoinnin (kuva 15). Arkistointijärjestelmä ei tarkista suostumuksia, joten vaikka opettaja on kieltänyt arkistointitoimen, järjestelmänvalvoja voi tästä huolimatta arkistoida kurssin. Lisäksi näytetään kurssiin kohdistettu arkistointitoimenpide, joka voi olla arkistointi, järjestelmänvalvojan pakotettu arkistointi tai kurssin poisto Moodlesta. Kurssin poisto suoritetaan Moodlen normaalilla kurssin poistotyökalulla, eikä toimenpide liity tämän jälkeen arkistointijärjestelmään.

Kymenlaakson ammattikorkeakoulu
University of Applied Sciences

Moodle-oppimisympäristö

EvoMoodle ► FOK007 ► Kurssin arkistointi

Olet kirjautunut nimellä vilpito käyttäjä. (Kirjautu ulos)

Kurssin arkistointi

Arkistoitavan kurssin tiedot

Koko nimi: Funktionaalinen ohjelmointi
 Lyhenne: FOK007
 Katogoria: Ohjelmointi
 Pyynnön esittäjä: Mika Hatakka
 Toimenpide: Arkistointi
 Arkistointi pyydetty: 20.04.2010, 19:42
 Kurssin muut opettajat: Siru Särö
 Vierailevat opettajat: Teppo Töpö

Suorita toimenpide Peruuta

Tämä sivu Moodle Docs -sivustolla

EvoMoodle ► FOK007 ► Kurssin arkistointi

Kuva 15. Arkistoitavaksi valitun kurssin tiedot

Varsinainen arkistointi aloitetaan keräämällä kaikki tiedot, niin pyyntötietokannasta kuin kurssijärjestelmästä ja tarkastetaan, että arkistoitava kurssi on olemassa. Kurssin arkistointitapahtuma koostuu kurssin pakkaamisesta, pakatun tiedoston siirrosta arkistohakemistoon ja kurssin poistamiseen Moodlen kurssijärjestelmästä sekä tietojen liittämistä arkistotietokantaan.

Kurssin pakkaaminen hoidetaan Moodlen varmistus-toiminnolla. Varmistus on toteutettu omaksi kiinteäksi kokonaisuudeksi, eikä tue modulaarisuutta. Tämän takia jouduttiin toteuttamaan omat liittynät varmistus-toimintoon. Yhteensopivuuden takaamiseksi vaiva oli kuitenkin varsin pieni.

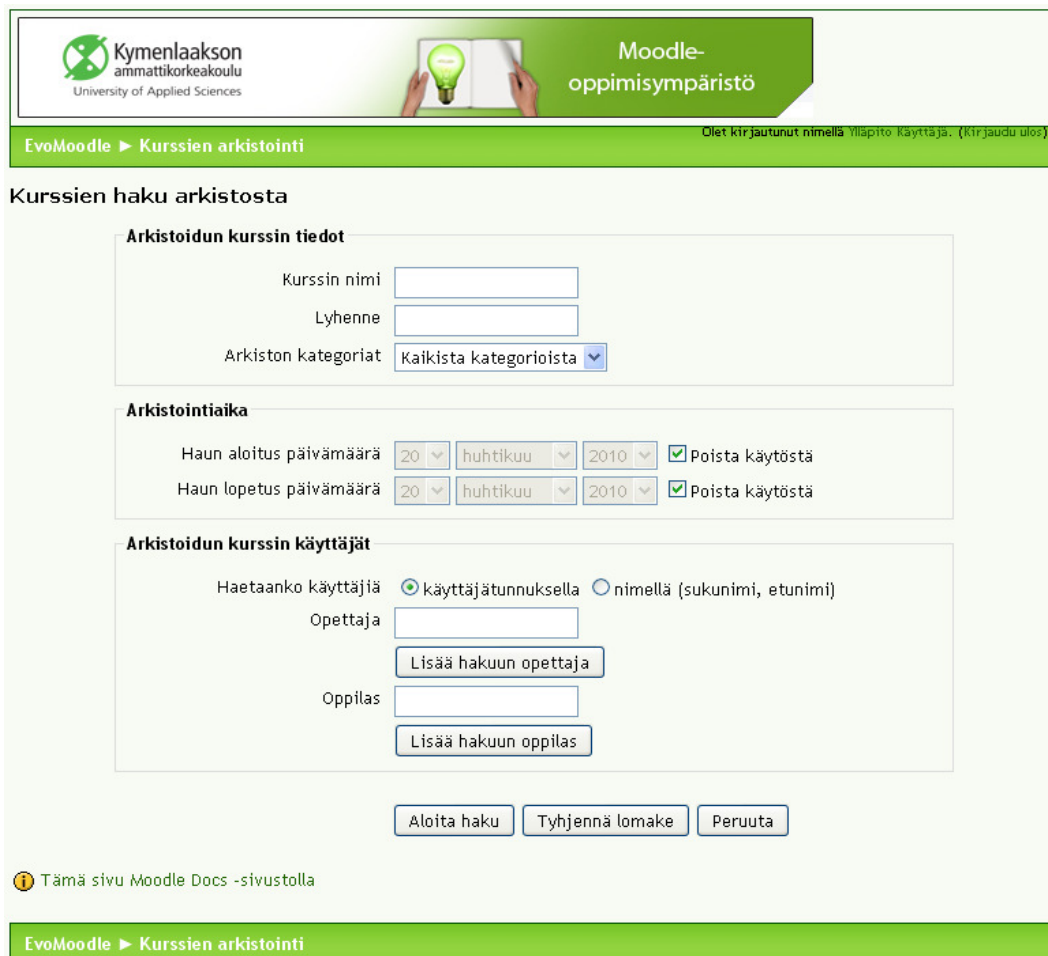
Pakkausvaihe koostuu alustavien tietojen kokoamisesta, jota seuraa tietojen muunnos XML-muotoon ja tämän jälkeen suoritetaan varsinainen pakkaustoimenpide. Pakkauksen jälkeen tarkistetaan, että tiedosto on olemassa oikeassa paikassa ja oikealla nimellä. Arkistotiedosto siirretään arkistohakemistoon ja aloitetaan arkistoinnin lopetus-toimet. Arkistoinnin aikana muodostetut väliaikaiset tiedostot ja hakemistot poistetaan, päivitetään arkistotietokanta, poistetaan kaikki kurssiin viittaavat toimet pyyntötietokannasta ja lähetetään opettajille sähköpostitse ilmoitus arkistointitoimenpiteen suorituksesta. Viimeisenä vaiheena kurssi poistetaan Moodle kurssijärjestelmästä.

Arkistointi vaiheen aikana muodostetaan raportti arkistoinnin tilasta. Raporttiin kirjataan myös arkistoinnin aikana tapahtuneet virheet. Mikäli arkistoinnin aikana ilmenee virheitä, lopetetaan arkistointi ja luodaan järjestelmänvalvojalle ilmoitus virheestä. Kurssi palautetaan myös tilaan, jossa se oli ennen arkistoinnin alkamista.

5.8 Kurssin palautus arkistosta

5.8.1 Kurssin haku arkistosta

Kurssiarkistolle on toteutettu hakutoiminto, jonka avulla arkistotietokannasta etsitään annetuilla ehdoilla kursseja (kuva 16). Kurssin haku aloitetaan antamalla hakuehdot. Hakuehdot ovat toisiaan tarkentavia eli kaikkien hakuehtojen on täsmättävä. Tyhjiä hakukenttiä ei huomioida. Hakuehdot on jaettu kolmeen osaan: arkistoidun kurssin tietoihin, arkistointiaikaan ja arkistoidun kurssin käyttäjiin. On myös mahdollista aloittaa haku ilman yhdenkään ehdon asettamista, tällöin tulosjoukko koostuu kaikista arkistotietokannassa olevista kursseista.



The screenshot shows the Moodle course search interface for archived courses. At the top, there is a header for 'Kymenlaakson ammattikorkeakoulu' and 'Moodle-oppimisympäristö'. Below the header, the page title is 'EvoMoodle ► Kurssien arkistointi'. The main content area is titled 'Kurssien haku arkistosta' and contains three sections of search criteria:

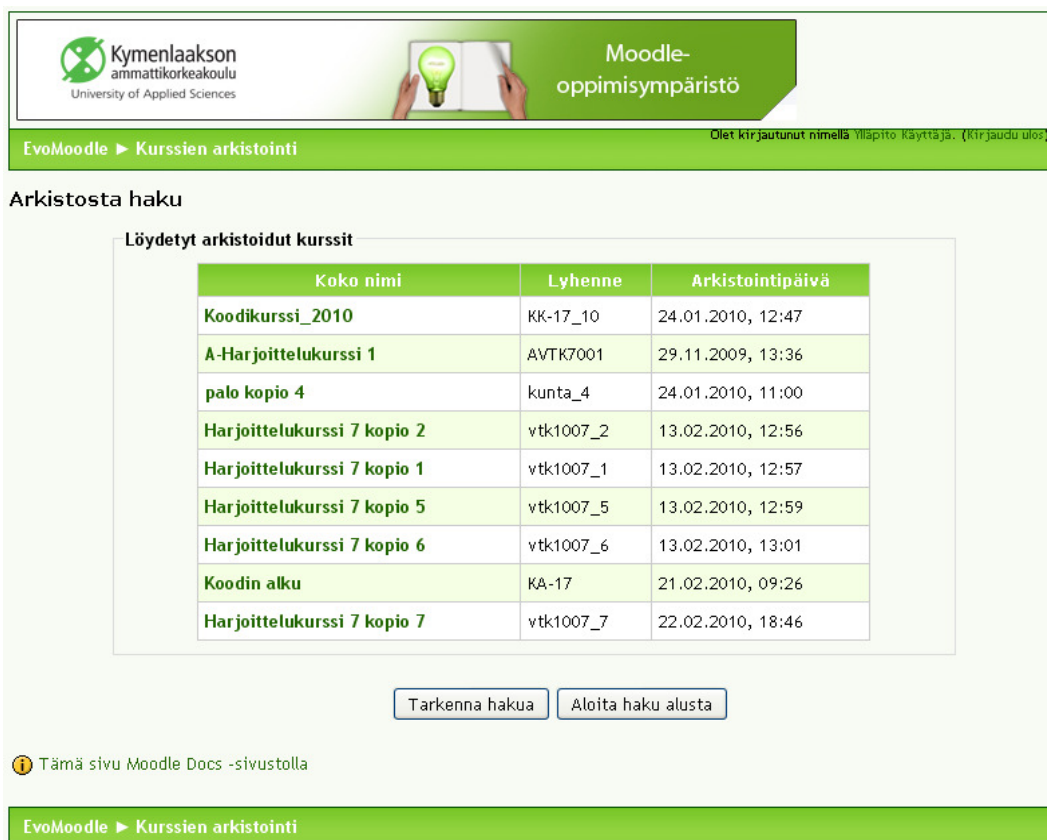
- Arkistoidun kurssin tiedot:** Includes input fields for 'Kurssin nimi', 'Lyhenne', and a dropdown menu for 'Arkiston kategoriat' set to 'Kaikista kategorioista'.
- Arkistointiaika:** Includes date pickers for 'Haun aloitus päivämäärä' and 'Haun lopetus päivämäärä', both set to '20' and 'huhtikuu' of the year '2010'. There are checkboxes for 'Poista käytöstä' next to each date field.
- Arkistoidun kurssin käyttäjät:** Includes radio buttons to select search criteria: 'käyttäjätunnuksella' (selected) or 'nimellä (sukunimi, etunimi)'. Below are input fields for 'Opettaja' and 'Oppilas', each with a 'Lisää hakuun' button.

At the bottom of the search section, there are three buttons: 'Aloita haku', 'Tyhjennä lomake', and 'Peruuta'. A footer note indicates 'Tämä sivu Moodle Docs -sivustolla'.

Kuva 16. Arkistosta haku

Arkistoidun kurssin tiedoista hakuun voidaan käyttää kurssin nimeä, lyhennettä ja kategoriaa. Kategoriasta on huomioitava, että se käyttää arkistoitujen kurssien alkuperäisiä kategorioita. Jos kategorioiden ovat myöhemmin muuttuneet, ei uusilla kategorioilla saada aikaan osumia. Hakua on mahdollista tarkentaa arkistointiajan perusteella, on mahdollista käyttää sekä alku- että loppuaika tai vain toista ajoista. Oletuksena arkistointiaika on suljettu pois haluehdoista. Käyttäjien haku voidaan valita toimimaan käyttäjätunnuksen tai nimen mukaan. Käytettäessä hakua nimellä, tulee huomioida nimen esitysmuoto. Sukunimi annetaan aina ensin, etunimeä voidaan käyttää tarkentamaan ehtoa, tällöin se erotetaan pilkulla sukunimestä. Pelkkää etunimeä ei hakuehdoksi voi määrittää. Hakuun voidaan käyttää sekä opettajia että opiskelijoita. Painonapilla voidaan lisätä uusia kenttiä, mikäli hakuun tarvitaan useampia käyttäjiä.

Mitä tarkemmat rajaukset pystytään antamaan, sitä tarkemman tuloksen haku antaa. Jos hakuun saadaan tulokseksi vain yksi kurssi, siirrytään suoraan kurssin tietojen näyttöön, muuten siirrytään näyttämään tuloslistaus. Tuloslistauksessa on arkistoiduista kursseista tiedot kurssin nimestä, lyhenteestä ja arkistointipäivästä (kuva 17). Kurssin nimestä siirrytään kurssin tietojen näyttöön. Mikäli hakuun ei löydy yhtään sopivaa kurssia, tulostetaan hakusivulle ohje tarkentaa hakuehtoja.



Kymenlaakson ammattikorkeakoulu
University of Applied Sciences

Moodle-oppimisympäristö

EvoMoodle ► Kurssien arkistointi Olet kirjautunut nimellä Ylläpito Käyttäjä. {Kirjautu ulos}

Arkistosta haku

Löydetty arkistoidut kurssit

Koko nimi	Lyhenne	Arkistointipäivä
Koodikurssi_2010	KK-17_10	24.01.2010, 12:47
A-Harjoittelukurssi 1	AVTK7001	29.11.2009, 13:36
palo kopio 4	kunta_4	24.01.2010, 11:00
Harjoittelukurssi 7 kopio 2	vtk1007_2	13.02.2010, 12:56
Harjoittelukurssi 7 kopio 1	vtk1007_1	13.02.2010, 12:57
Harjoittelukurssi 7 kopio 5	vtk1007_5	13.02.2010, 12:59
Harjoittelukurssi 7 kopio 6	vtk1007_6	13.02.2010, 13:01
Koodin alku	KA-17	21.02.2010, 09:26
Harjoittelukurssi 7 kopio 7	vtk1007_7	22.02.2010, 18:46

Tarkenna hakua Aloita haku alusta

Tämä sivu Moodle Docs -sivustolla

EvoMoodle ► Kurssien arkistointi

Kuva 17. Arkistosta haun tuloslista

Arkistoidun kurssin tietojen näytössä on kaikki oleellinen tieto kurssista (kuva 18). Siinä näkyvät kurssin nimi, lyhenne ja kategoria, arkistoinnin suorittaja ja arkistointiaika sekä kuka arkistoinnin on pyytänyt. Kurssin opettajat ja opiskelijat on listattu omiin osioihinsa, opettajat on myös jaoteltu varsinaisiin ja toissijaisiin opettajiin. Toiminto vaihtoehtoina ovat arkistoidun kurssin palautus tarkkana kopiona tai kurssipohjana, jolloin käyttäjät ja käyttäjien tiedot jätetään pois. Lisäksi on mahdollista palata tarkentamaan hakua tai palata tuloslistalle.

The screenshot shows the Moodle course archive interface. At the top, there is a header for 'Kymenlaakson ammattikorkeakoulu' (University of Applied Sciences) and 'Moodle-oppimisympäristö'. Below the header, the breadcrumb 'EvoMoodle > Kurssien arkistointi' is visible. The main content area is titled 'Näytä valitun kurssin tiedot' (Show details of the selected course). It contains several sections:

- Näytä arkistoidun kurssin tiedot** (Show details of the archived course):
 - Koko nimi: Funktionaalinen ohjelmointi
 - Lyhenne: FOK007
 - Kategoria: Ohjelmointi
 - Arkistoinnin suoritti: Ylläpito Käyttäjä
 - Arkistointi suoritettu: 20.04.2010, 20:04
 - Arkistotiedoston nimi: 2010S/varmuuskopio-fok007-20100420-2004.zip
 - Pyynnön esittäjä: Mika Hatakka
- Opettajat** (Teachers):
 - Opettajat: Mika Hatakka, Siru Särö
 - Vierailevat opettajat: Teppo Töpö
- Opiskelijat** (Students):
 - Opiskelijat: Tahvo Niitti, Otto Opiskelija
- Kurssin palautus** (Course return):
 - Buttons: 'Palauta arkistosta' (Return from archive), 'Palauta uutena kurssipohjana' (Return as new course template)
 - Buttons: 'Tarkenna hakua' (Refine search), 'Palaa tuloslistaan' (Return to results list)

At the bottom, there is a footer with the text 'Tämä sivu Moodle Docs -sivustolla' (This page on Moodle Docs website) and another breadcrumb 'EvoMoodle > Kurssien arkistointi'.

Kuva 18. Arkistoidun kurssin tietojen näyttö

5.8.2 Arkistosta palautus

Tarvittaessa tietoja arkistoidun kurssin suorituksista tai muusta arkistoidusta materiaalista tai haluttaessa käyttää arkistoitua kurssia uuden pohjana, on kurssi palautettava arkistosta. Palautus prosessi etenee kolmessa osassa, ensin alustetaan ympäristö palau-

tusta varten, seuraavaksi kerätään palautuksen suoritusta varten tarvittavat tiedot ja lopuksi suoritetaan varsinainen palautus. Ympäristön alustuksessa varmistetaan, että palautuksen aikana tarvittavat istuntomuuttujat ovat tyhjä, luodaan tarvittavat väliaikaiset hakemistot ja tiedostot sekä kopioidaan palautettava kurssi arkistosta.

Palautustoiminnon seuraavassa vaiheessa valitaan arkistoidulle kurssille palautustapa vaihtoehdoista: Palauta arkistosta tai Palauta uutena kurssipohjana. Palauta arkistosta luo Moodlen kurssijärjestelmään kopion kurssista sellaisena, kuin se on arkistoitu. Kaikki kurssin tiedot, opiskelijoiden palautukset, opettajan materiaalit ja kurssin tapahtumat palautetaan. Samoin opettajat ja opiskelijat liitetään palautettuun kurssi kopioon. Poikkeuksen muodostavat valmistuneet opiskelijat, koska heillä ei ole enää aktiivista tiliä Moodlella, heitä ei liitetä kurssiin. Heidän tietonsa ovat kyllä edelleen arkistossa, mutta ilman erikoistoimia niitä ei saada palautettua. Valmistuneet ovat opintonsa suorittaneet, joten heidän tietoihinsa ei ole tarvetta palata. Muualle siirtyneiden opettajien kohdalla käytäntö on sama. Kurssia palautettaessa kurssin nimi ja lyhenne saattavat muuttua, mikäli samanniminen kurssi on jo olemassa, tällöin noudatetaan Moodlen normaalia nimeämiskäytäntöä, jossa kurssin nimen perään lisätään alaviiva ja juokseva numero.

Uutena kurssipohjana palautettaessa jätetään kurssiin vain kurssimateriaali, kaikki muut jätetään palauttamatta. Opiskelijoita ja opettajia ei liitetä kurssille, eikä myöskään opiskelijoiden palautuksia. Kurssille kysytään uusi nimi, lyhytnimi ja kategoria (kuva 19). Käytettäessä arkistoitua kurssia uuden pohjana, on tarkastettava kenellä on tekijänoikeudet kurssin sisältöön.

Kurssin palautustavan valinnan jälkeen alustetaan palautukseen tarvittavat tietorakenteet palautettavista tiedoista. Yhteisen rakenteen vuoksi myös Palauta arkistosta -valinta kierrätetään uusien kurssitietojen kyselylomakkeen kautta. Tällä helpotetaan ylläpitoa ja mahdollisten muutosten suorittamista. Viimeisenä vaiheen on kurssin varsinainen palautus Moodlen kurssijärjestelmään. Toimenpide suoritetaan Moodlen palautus-toimintolla. Palautus-toiminto, kuten varmistus-toimintokin on Moodlella oma kokonaisuutensa, johon oli toteutettava oma liityntä. Palautus on suoraviivainen toimenpide, jossa ensin tarkistetaan, että kaikki tarvittavat tiedot kerätty. Tämän jälkeen aloitetaan palautus ja raportti tulostetaan käyttäjälle toiminnan edetessä. Palautuksen päätteeksi poistetaan väliaikaiset tiedostot ja hakemistot ja luodaan painike palautetul-

le kurssille siirtymiseen. Toiminnon aikana tapahtuneista virheistä tulostetaan raporttiin virhe ilmoitus.

The screenshot shows the Moodle course archive interface. At the top, there is a header for 'Kymenlaakson ammattikorkeakoulu' (University of Applied Sciences) and 'Moodle-oppimisympäristö'. Below the header, a navigation bar indicates 'EvoMoodle ► Kurssin palautus arkistosta'. The main content area is titled 'Kurssin palautus arkistosta' and contains a form for creating a new course. The form is titled 'Näytä arkistoidun kurssin tiedot' and includes input fields for 'Kurssin nimi', 'Lyhenne', and a dropdown menu for 'Kurssikategoriat' (currently set to 'Muut'). Below the form are two buttons: 'Luo uusi kurssi' and 'Peruuta'. At the bottom of the form area, there is an information icon and the text 'Tämä sivu Moodle Docs -sivustolla'. A footer bar at the bottom of the page repeats the navigation path 'EvoMoodle ► Kurssin palautus arkistosta'.

Kuva 19. Uuden kurssin tietojen kysely

6 SAAVUTETUT EDUT JA HAITAT

Arkistointijärjestelmästä tuli kiinteä osa Moodlea. Toimintojen logiikka on sama kuin muuallakin toimiessa, joten toimintatavat ovat jo entuudestaan tuttuja. Opettajan osalta toiminnot ovat yksinkertaiset, valitaan arkistointi ja tämän jälkeen toiminta siirtyy omista käsistä. Peruutus on tietenkin mahdollista, kunnes järjestelmänvalvojan ottaa pyynnön käsiteltäväksi. Vielä tämänkin jälkeen kurssi on mahdollista palauttaa, mikäli kyseessä oli arkistointipyyntö. Poisto on peruuttamaton sen jälkeen, kun pyyntö on toteutettu.

Järjestelmänvalvojan osalta toiminta kurssien arkistoinnista on nyt suoraviivaisempaa. Arkistoitavat kurssit keräytyvät tietokantaan, jossa ne odottavat, kunnes järjestelmänvalvojalla on aikaa niitä toteuttaa. Koska tietoja arkistoitavista kursseista ei tarvitse erikseen etsiä, on niiden käsittely mahdollista aina, kun pienikin hetki aikaa löytyy. Kurseja voi arkistoida vaikka yhden kerrallaan tai tyhjentää koko tietokannan. Toiminta on helppo sopeuttaa kulloinkin vallitsevaan työtilanteeseen. Lisäksi järjestelmänvalvojalle on mahdollisuus kursseja selatessaan aloittaa yksittäisen kurssin arkistointi suoraan kurssisivulta. Tällöin esimerkiksi toiseen oppilaitokseen siirtyneen opet-

tajan tai ilman opettajaa olevat kurssit voidaan arkistoida pelkäämättä, että arvokkaita tietoja menetettäisiin.

Arkistoituja kursseja on helppo etsiä ja hakuehdot on valittavissa varsin vapaasti. Hakuehdot kattavat kurssin perustiedot ja tiedot käyttäjistä, niin opettajista kuin opiskelijoista. Halutun kurssin palautus onnistuu suoraan, joko tarkkana kopiona tai uutena kurssipohjana, eikä vaadi erityisiä toimenpiteitä järjestelmänvalvojalta.

Koska arkistointijärjestelmä saatiin koekäyttöön vasta maaliskuussa 2010, ei vielä ole muodostunut tietoa mahdollisista haitoista tai selkeistä puutteista. Asennuksen yhteydessä tuli esiin, että mikäli arkistohakemisto on tarkoitus sijoittaa Moodlen järjestelmähakemistojen ulkopuolelle, vaaditaan siihen pääkäyttäjän oikeudet palvelimeen. Arkistohakemiston luonti on siis hyvä pyytää jo ennen moduulin asennusta. Toteutuksen aikana esiin tulleita kehitysmahdollisuuksia on käsitelty seuraavassa luvussa.

7 KEHITYSMAHDOLLISUUDET

7.1 Yleistä

Arkistointijärjestelmää on mahdollista kehittää monellakin tavalla. Järjestelmää voidaan laajentaa muodostamaan laajempi kurssien arkisto- ja mallipohjavarasto. Näin mahdollistetaan tehokas kurssien uudelleen käyttö ja turha kurssipohjien kopiointi jää pois. Arkistointijärjestelmää on mahdollista kehittää myös pelkkänä arkistona luomalla käyttäjille ympäristö, jossa arkistointijärjestelmä pitää sisällään koko arkistoinnin kaareen kurssien arkistoinnista aina arkistoitujen kurssien siirtoon pysyviin varmenneformaateihin sekä riittävän raportoinnin. Moodlen osaltakin löytyy kehitettävää varmistus- ja palautusrutiinien osalta.

7.2 Kehitysmahdollisuudet arkistointijärjestelmään

Raportointi on ominaisuus, joka olisi hyvä lisätä lohkoon. Raportoinnin avulla pystytään muodostamaan selkeä kuva arkiston käytöstä ja käyttäjistä. Raportointi esitetyistä arkistointipyynnöistä ja niiden esittäjistä sekä arkistoiduista kursseista käyttäen sopivia suodattamia, antavat järjestelmänvalvojalle hyvät työkalut seurata arkistointia.

Arkistoon voisi kehittää lisäosan, jonka avulla arkistoitavasta kurssista muodostettaisiin mallipohja. Tällöin saataisiin luotua kurseista pohja-arkisto, josta opettajille olisi mahdollista muodostaa lista valittavista kurssipohjista. Opettajalle näytettävät kurssit voisivat olla hänen omista kurseista muodostettuja tai muiden opettajien yleiseen mallipohjapankkiin luovuttamia. Opettajat pääsisivät näin suoraan luomaan itselleen kurssipohjan, ilma järjestelmänvalvojan puuttumista kurssin luontiin.

Vientiominaisuus olisi hyvä olla helpottamassa arkistoitujen kurssien siirtämistä toisiin Moodle-järjestelmiin. Opettajan siirtyessä uuteen kouluun opettamaan hänelle tarjottaisiin mahdollisuus ottaa mukaan kurssipohjat omista arkistoiduista kurseista. Samalla tulisi ottaa huomioon myös muut siirto formaatit, kuten SCROM, jota käytetään muualla.

Arkistoitujen kurssien määrän kasvaessa tulee tarve siirtää arkistoja talteen esimerkiksi CD-ROM-levyille. Tällöin arkistoinnin kannalta olisi optimaalista, mikäli arkistotietokantaan saadaan tieto siirrettyjen kurssien sijainnista. Arkistojen siirron pysyville formaateilla ollessa osa arkistointijärjestelmää tietojen pysyminen ajan tasalla saadaan hoidettua automaattisesti.

Yhtenä kehittämismahdollisuutena on täysin oman arkistointi metodin kehittäminen. Tällä saavutettaisiin riippumattomuus Moodlen varmistus- ja palautus-toimintojen kehityksestä. Haittapuolena on tietysti arkistointijärjestelmän välttämättömyys palautettaessa kurseja arkistosta.

7.3 Kehittämismahdollisuudet Moodleen

Moodlen varmistus- ja palautusrutiinien käyttäminen edellyttää lähes samanlaisen koodin käyttämistä. Kurssin aktiviteettien sisältämien tuen tarkistus olisi hyvä siirtää kirjastorutiineihin. Samalla helpotettaisiin varmistuksen ja palautuksen jatkokehityksen hyödyntämien myös tulevissa Moodlen versioissa ja yhteensopivuus vanhojen ja uusien versioiden välillä aiheuttaisi vähemmän ongelmia. Samalla myös muiden varmistus tai palautus rutiineja hyödyntävien moduulien tulisi parantaa yhteensopivuutta myös tulevien Moodlen versioiden kanssa.

Moodlessa modulaarisuuden laajentaminen koskemaan myös enemmän järjestelmänhallinnan toimintoja avaisi uusia mahdollisuuksia järjestelmätason toimintojen kehittä-

tämiseen. Tämän työn osalta tuli hyvin esille varmistus- ja palautus-toimintojen kiinteä rakenne Moodlen ydintoimintoihin, tämä aiheuttaa paikoin koodin toistoa. On tietysti hyvä, että varmistus ja palautusrutiinit ovat kiinnitetty tiukasti ytimeen, mutta kiinnittämällä hieman huomiota rajapinnan ratkaisuihin, mahdollistettaisiin rutiinien kehittämisestä seuraava, omien moduulien muutoksien parempi hallinta.

8 JOHTOPÄÄTÖKSET

Jo varhaisessa vaiheessa kaksi asiaa nousi esiin: hyvä modulaarisuutta hyödyntävä lopputulos edellyttää hyvää suunnittelua ja kohdealusta tulee tuntea hyvin. Nämä seikat ovat tiukasti sidoksissa toisiinsa. Ilman kohdealustan tuntemusta suunnittelu jää väkisinkin puolitiehen. Projektin edetessä tuli hyvin esiin, että se oli ensimmäinen suurempi toteutus Moodleen. Projektin ensimmäisen kolmanneksen kohdalla kävi selväksi, että Moodlen tuntemus aloitettaessa ei ollut riittävä. Tuossa vaiheessa tuntemusta ympäristöstä alkoi olla riittävästi, jotta voi todeta siihenastisen työn olevan varsin alkeellista. Mutta aikataulu ei enää antanut mahdollisuutta kirjoittaa siihenastista työtä uusiksi, vaan oli elettävä sen kanssa. Samassa vaiheessa tuli esiin myös suunnitelmien uudelleen tarkistus projektin loppuosalle. Suunnitelma ei enää ollut ajan tasalla, mikä johtui puutteellisesta ympäristön tuntemuksesta projektin alkaessa.

Työn edetessä kuitenkin nousi esiin se työ, joka oli tehty jo aiemmin, kun järjestelmän ominaisuuksia oli mietitty tilaajan kanssa. Toteutustason suunnitelmia oli helppo päivittää, koska perustoiminnot olivat selkeästi selvillä. Toimintojen tarkka määrittely ja tarkastaminen, että puhutaan asiakkaan kanssa samoista asioista samoilla termeillä, luo tukevan pohjan, jolle järjestelmän rakentaminen on hyvä aloittaa.

Arkistojärjestelmän osalta Moodlen modulaarinen rakenne näkyi erityisesti lohkon ja lomakkeiden osalla. Lohko oli helppo liittää Moodleen ja roolimalli antoi hyvät mahdollisuudet toteuttaa molemmille pääkäyttäjätahoille omat erikoistetut näkymänsä. Peruslomakkeiden osalta komponenttien sijoittelu ja toiminnot kävivät varsin vaivattomasti. Omien perustoiminnallisuuksien lisäämien oli yksinkertaista, hyödyntäen perusluokkien erikoistamista.

Suurimmat ongelmakohdat tulivat esiin Moodlen varmistus- ja palautus-toimintojen kohdalla. Molemmat ovat varsin omineen toimivia moduuleja, niitä ei ole liitetty tiukasti Moodleen ytimeen, jolloin niitä olisi helppo hyödyntää. Ne eivät myöskään toimi

selkeästi modulaarisina yksikköinä, mikä myös tukisi niiden hyväksikäyttöä. Ne ovat kokoelma aliohjelmiä, joiden kohdalta toteutus on jäänyt puolitiehen. Niiden käyttö aiheuttaa omaan lähdekoodiin useita toistoja, joita on äärimmäisen vaikea saada yhdistettyä. Toiminta on jaoteltu useisiin aliohjelmiin, jotka tallentavat tietoja oman mallinsa mukaan, ilman johdonmukaisuutta. Näiden toimintojen osalta olisi toivottavaa, että niiden modulaarisuus ja uudelleenkäyttö ominaisuudet paranisivat tulevissa Moodlen versioissa. Näin toimien uusi Moodle-versio aiheuttaa väistämättä arkistointijärjestelmään päivityksiä.

Arkistointijärjestelmä on varsin pitkälle itsenäinen, modulaarinen toiminnallinen yksikkö. Se asentuu omaksi kokonaisuudeksi ja toimii hyvässä vuorovaikutuksessa muun Moodlen kanssa. Varmistus- ja palautus-toimintojen osalta tukeutuminen Moodlen toimintoihin rikkoo itsellistä toimintaa, mutta kokonaisuuden kannalta näin on kuitenkin parempi. Saavutetaanhan sillä arkistojen osalta riippumattomuus myös arkistointijärjestelmästä. Arkistosta voidaan palauttaa suoraan Moodlen omalla palautustoiminnolla kurssit takaisin oppimisympäristöön. Pienistä puutteista huolimatta voidaan todeta, että arkistointijärjestelmä tukee ja lisää Moodlen modulaarisuutta, ja ilman sitä tällaisen järjestelmän toteuttaminen olisi ollut monin verroin hankalampaa, ellei jopa mahdotonta.

LÄHTEET

Clark, K. B. & Baldwin, C. Y. 1999. Design Rules: The Power of Modularity. Cambridge: MIT Press.

Cole, J. 2005. Using Moodle: Teaching with the Popular Open Source Course Management System. Sebastopol: O'Reilly.

Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. Helsinki: Talentum

Järvinen, P. 2009. Digiarkistointi: säilytä muistot ja tiedostot. Jyväskylä: Docendo.

Karevaara, S. 2008. Moodlen perusteet: opettajan ja opiskelijan opas. Helsinki: Finn Lectura.

Kymenlaakson ammattikorkeakoulun tutkintosääntö 2009 - 2010. 2009. Saatavissa: <https://elack.kyamk.fi/TWeb/tfile?id=30529>. [viitattu 24.10.2009].

Lehtimäki, T. 2006. Ohjelmistoprojektit käytännössä. Helsinki: Readme.fi.

Moodle. 2009. Philosophy. Saatavissa: <http://docs.moodle.org/en/Philosophy>. [viitattu 24.10.2009].

MOODLEN ARKISTOINTILOHKON KÄYTTÖOHJE

Mika Hatakka

Kymenlaakson ammattikorkeakoulu

SISÄLLYS

1	OPETTAJA	3
1.1	Kurssien arkistointi lohko	3
1.2	Pyynnöt	3
1.3	Pyynnön poisto	4
1.4	Pyynnön vahvistus	4
1.5	Ilmoitus ristiriidasta	5
1.6	Vieraileva opettaja	6
2	JÄRJESTELMÄNVALVOJA	7
2.1	Asetukset	7
2.2	Lohko	8
2.3	Arkistointi	9
2.4	Palautus	10

1 OPETTAJA

1.1 Kurssien arkistointi lohko

Omille kursseille on mahdollista pyytää järjestelmänvalvojalta kurssin arkistointia tai poistoa. Oikeus on kurssin ensisijaisilla opettajilla, toissijaisen opettajan oikeudet riippuvat asetuksista. Useita opettajia sisältävän kurssin pyyntöä käsiteltäessä, järjestelmänvalvoja pyytää suostumuksen kaikilta kurssin ensisijaisilta opettajilta. Tapauskohteisesti pyydetään myös toissijaisilta opettajilta suostumus.

Kurssia arkistoidaessa siirretään se täydellisenä arkistoon ja poistetaan Moodlesta, jolloin kurssi poistuu myös opettajan kurssilistasta. Kurssi on myöhemmin mahdollista palauttaa takaisin täydellisenä kopiona alkuperäisestä tai kurssipohja ilman käyttäjiä ja heidän kurssille liittämiään tuotoksia. Poistettaessa kurssi se poistetaan lopullisesti, eikä sitä ole mahdollista palauttaa takaisin.

Toimenpiteen suorituksen jälkeen kaikille kurssin opettajille ilmoitetaan, että toimenpide on suoritettu. Tämän jälkeen arkistoituun kurssiin kohdistuvat toimenpiteet tulee lähettää järjestelmänvalvojalle sähköpostitse.

1.2 Pyyntöt

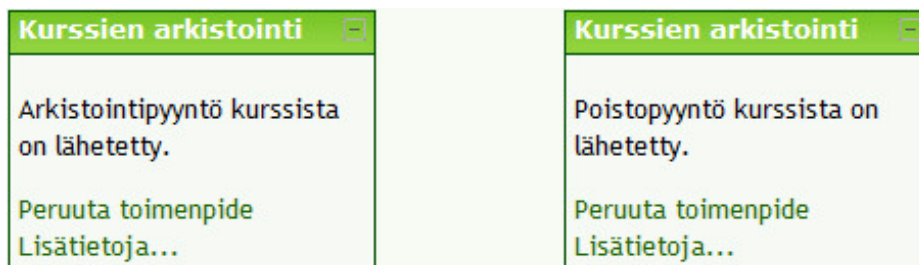
Kurssista voidaan esittää arkistointi- tai poistopyyntö (kuva 1). Pyyntö kirjataan järjestelmään ja järjestelmänvalvoja suorittaa halutun toimenpiteen.



Kuva 1. Arkistointilohkon opettajan perusnäky

1.3 Pyynnön poisto

Pyynnön esityksen jälkeen opettajalla on mahdollisuus perua pyyntö (kuva 2). Peruutus mahdollisuus säilyy kunnes järjestelmänvalvoja ottaa pyynnön käsiteltäväkseen, tämän jälkeen automaattinen peruutus ei ole mahdollista.



Kuva 2. Pyynnön peruutus

1.4 Pyynnön vahvistus

Mikäli kurssilla on useita opettajia, on ensisijaisilla opettajilla mahdollisuus hyväksyä tai kieltää pyyntö. Vieraileville opettajille näkyy vain ilmoitus pyynnön esittämisestä ja kuka pyynnön on esittänyt asetusten näin salliessa.



Kuva 3. Pyynnön vahvistus kurssin toisilta opettajilta

1.5 Ilmoitus ristiriidasta

Yhdenkin opettajan kieltäessä pyynnön, tulee muille opettajille tieto kiellosta tilanteesta ja kiellon esittäjä (kuva 4). Tilanteen voi purkaa ainoastaan kiellon esittäjä perumalla kiellon (kuva 5) tai järjestelmänvalvoja poistamalla esitetyn pyynnön järjestelmästä.



Kuva 4. Arkistointi kielletty



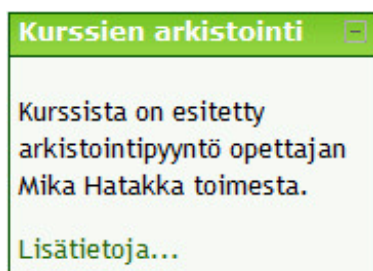
Kuva 5. Arkistoinnin kiellon peruutus

1.6 Vieraileva opettaja

Vierailevan opettajan oikeudet kurssin arkistointiin ovat säädeltävissä järjestelmänvalvojan asetuksissa. Näissä vierailevalta opettajalta voidaan piilottaa kaikki arkistointiin liittyvä tai näyttää informaatiota ja ohjeita kurssin arkistoinnista (kuva 6) ja sen tilasta (kuva 7). Kolmantena vaihtoehtona on, että vierailevalla opettajalla on samat oikeudet ja mahdollisuudet kuin ensisijaisella opettajalla.



Kuva 6. Vierailevan opettajan arkistointi tiedote



Kuva 7. Kurssin arkistoinnin tila

2 JÄRJESTELMÄNVALVOJA

2.1 Asetukset

Järjestelmänvalvoja pääsee muokkaamaan arkistointilohkon käyttäytymistä asetuksilla (kuva 8). Asetuksilla vaikutetaan hakemistoon, johon arkistoidut kurssit talletetaan, sähköpostiosoitteeseen, johon arkistointiin liittyvät kysymykset ja pyynnöt lähetetään sekä vierailevan opettajan oikeuksista.

Kurssien arkistointi

Arkistohakemisto Moodlen datahakemistossa Oletus: Ei
block_archive_dir2data Arkistohakemisto voidaan määritellä Moodlen datahakemistoon.

Arkistohakemisto Oletus: moodlearchive
block_archive_directory Hakemisto johon arkistoidut kurssit tallennetaan. Hakemistopolku on annettava juurihakemistosta alkaen(/home/moodlearchive). Kuitenkin hakemiston ollessa Moodlen datahakemistossa, määritellään vain datahakemiston alainen polun osa (moodlearchive).

Arkistojen sähköposti Oletus: Tyhjä
block_archive_archiveemail Sähköposti osoite, johon arkistointia koskevat kysymykset ja pyynnöt lähetetään

Näytetään kaikille opettajille Oletus: Ei
block_archive_show2noned Arkistointilohko näytetään myös vieraileville opettajille.

Opettajilla samat oikeudet arkistoida Oletus: Ei
block_archive_asteacher Kaikilla opettajilla, myös vierailevilla, on oikeus esittää arkistointi toimenpide pyyntöjä.
Huom! edellyttää että 'Näytetään kaikille opettajille' valinta on valittuna.

Kuva 8. Arkistointilohkon järjestelmänvalvojan asetukset

Arkistohakemisto voidaan määritellä Moodlen datahakemiston alle, jolloin valitaan sijaintipaikaksi datahakemisto ja annetaan pelkkä hakemiston nimi. Mikäli hakemisto ei sijaitse datahakemistossa, tulee hakemisto määritellä absoluuttisesti juurihakemistosta alkaen. Hakemisto olisi hyvä sijoittaa muualle kuin Moodlen datahakemistoon, tällöin siihen sitoutuva tila ei häiritse Moodlen normaalia toimintaa. Arkistohakemiston sijoitus Moodlen datahakemistoon on tilanteita varten, joissa ulkopuolisiin hakemistoihin ei ole käyttöoikeuksia. Arkistohakemiston ollessa Moodlen hakemistojen ulkopuolel-

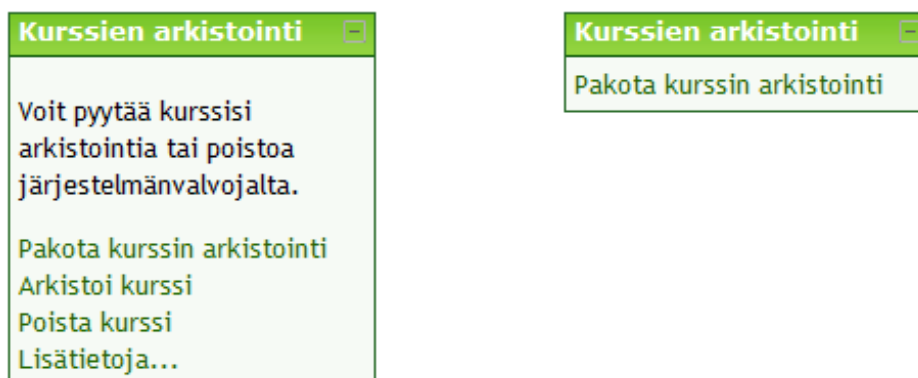
la, on se luotava ennen lohkon asennusta. Jos arkistohakemisto sijoitetaan Moodlen datahakemistoon, se luodaan ensimmäisen arkistoinnin yhteydessä, mikäli sitä ei vielä ole luotu.

Sähköpostiosoitteeksi on laitettu osoite, johon kaikilla arkistointia suorittavilla on pääsy. Arkistointia varten on hyvä luoda oma sähköpostitili pelkästään siihen tarkoitukseen. Tällöin arkistointiin liittyvä posti ei sekaannu muuhun ylläpidolle tulevaan posttiin.

Vierailevien opettajien arkistoinnin käyttömahdollisuuksia voidaan hallita asetuksilla. Oletus on, että vieraileville opettajille ei edes näytetä arkistointilohkoa kurssisivulla. Kuitenkin organisaatioissa roolilla on erilaisia käyttöjä, joten vieraileville opettajille on lisätty kaksi muuta toimintatasoa. Vieraileville opettajille voidaan näyttää arkistointilohko ja sen kautta tietoa arkistoinnista ja arkistoinnin tilasta kurssilla. Toinen vaihtoehto on antaa vierailevalle opettajalle samat oikeudet käyttää arkistointia kuin ensisijaisella opettajalla.

2.2 Lohko

Kurssisivulla järjestelmänvalvojalla on kahden tyyppinen perusnäky (kuva 9). Hänen ollessaan kurssilla opettajana näky on opettajan perusnäky, johon on lisätty linkki välittömään arkistoinnin aloittamiseen. Järjestelmänvalvojan ollessa kurssilla toissijaisena opettajana, näkyy hänelle normaali ilmoitus lisätynä linkillä välittömään arkistointiin. Muilla kursseilla järjestelmänvalvojalle näkyy vain välitön kurssin arkistoinnin aloitus linkki.



Kuva 9. Järjestelmänvalvojan lohkonäkymät kurssilla

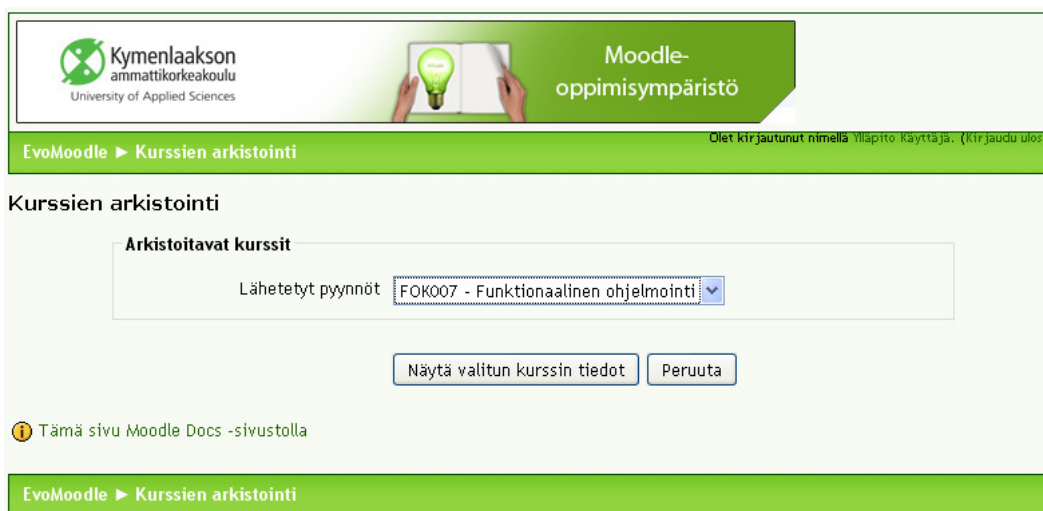
Etusivulla järjestelmänvalvojan lohko näkyvässä (kuva 10) kerrotaan arkistoitavien kurssien määrä, ja jos näitä on, linkki arkistoinnin aloitukseen sekä linkki kurssien haakuun.



Kuva 10. Järjestelmänvalvojan lohkonäkymät etusivulla

2.3 Arkistointi

Arkistointiin siirrytään etusivulta, kun arkistoitavia kursseja on tai valittaessa kurssisivulta välitön kurssin arkistointi. Etusivulta tultaessa tullaan ensin kurssin valinta sivulle, josta päästään arkistoitavan kurssin perustietojen näyttöön (kuva 11).



Kuva 11. Arkistoitavan kurssin valinta

Välittömän arkistoinnin valinnan jälkeen siirrytään suoraan arkistoitavan kurssin tietojen näyttöön (kuva 12). Haluttaessa jatkaa toimenpidettä valitaan 'Suorita toimenpide'-painike ja käynnistetään toimenpide. Mikäli kyseessä on kurssin poisto, poistetaan kurssi Moodlen omien ruutien mukaan, joka alkaa varmistus kysymyksellä ja tämän jälkeen poistaa kurssin niin haluttaessa. Arkistoinnin ollessa kyseessä toiminta alkaa suoraan ja toiminnon aikana luodaan raportti arkistoinnin etenemisestä. Lopuksi kerrotaan toiminnon onnistumisen tila. Virheet ennen arkistotiedoston tekoa peruuttavat

arkistoinnin ja arkistointipyyntö jää tietokantaan. Arkistoitava kurssi poistetaan Moodlesta arkistointitoimen päätteeksi.

The screenshot shows the Moodle course archive page for FOK007. The page header includes the logo of Kymenlaakson ammattikorkeakoulu (University of Applied Sciences) and the Moodle-oppimisympäristö logo. The breadcrumb trail is EvoMoodle > FOK007 > Kurssin arkistointi. The main content area is titled 'Kurssin arkistointi' and contains a box for 'Arkistoitavan kurssin tiedot' (Course archive details). The details are as follows:


Koko nimi:	Funktionaalinen ohjelmointi
Lyhenne:	FOK007
Kategoria:	Ohjelmointi
Pyynnön esittäjä:	Mika Hatakka
Toimenpide:	Arkistointi
Arkistointi pyydetty:	20.04.2010, 19:42
Kurssin muut opettajat:	Siru Särö
Vierailevat opettajat:	Teppo Töpö

Below the details box are two buttons: 'Suorita toimenpide' and 'Peruuta'. At the bottom of the page, there is a note: 'Tämä sivu Moodle Docs -sivustolla'.


Kuva 12. Arkistoitavaksi valitun kurssin tiedot

2.4 Palautus

Palautus osasta näkyvin osa on arkistosta haku (kuva 13). Haku ehdot ovat: kurssin nimi, lyhenne, kategoria, arkistointiaika, opettajat sekä opiskelijat. Haku ehdoista voi käyttää halutun määrän, ehdot joita ei haluta mukaan, jätetään tyhjäksi. Opettajia ja opiskelijoita voidaan hakea käyttäjätunnuksen tai nimen mukaan, näistä toinen on va-
littava ja se koskee koko hakua. Hakuun kuuluvien opettajien ja opiskelijoiden määrää ei ole rajoitettu, uusia kenttiä voi lisätä yksi kerrallaan. Hakuehtoja ei ole rajoitettu ai-
neiston alkuun, vaan ne voivat sijaita missä kohtaa tahansa. Esimerkiksi hakuehdolla ”takka”, voi tämä sijaita missä kohtaa tahansa tutkittavaa tietoa, joten ehdon täyttävä:
takka, hatakka, jne. Haku ei myöskään erottele isoja ja pieniä kirjaimia.



**Kymenlaakson
ammattikorkeakoulu**
University of Applied Sciences



**Moodle-
oppimisympäristö**

EvoMoodle ► Kurssien arkistointi
Olet kirjautunut nimellä **Yliopisto Käyttäjä**. (Kirjautu ulos)

Kurssien haku arkistosta

Arkistoidun kurssin tiedot

Kurssin nimi

Lyhenne

Arkiston kategoriat

Arkistointiaika

Haun aloitus päivämäärä Poista käytöstä

Haun lopetus päivämäärä Poista käytöstä

Arkistoidun kurssin käyttäjät

Haetaanko käyttäjiä käyttäjätunnuksella nimellä (sukunimi, etunimi)

Opettaja

Oppilas

🔔 Tämä sivu Moodle Docs -sivustolla

EvoMoodle ► Kurssien arkistointi

Kuva 13. Arkistosta haku

Mikäli haun tuloksena löytyy vain yksi tulos, avataan tämän tiedot suoraan näytölle. Muussa tapauksessa avataan tuloksista lista, jossa on arkistoidun kurssin nimi, lyhenne ja arkistointipäivä (kuva 14). Tästä listasta voidaan valita kurssi, jonka tiedot näytetään ja listalle päästään palaamaan tietojen näytöstä (kuva 15).

Kymenlaakson ammattikorkeakoulu
University of Applied Sciences

Moodle-oppimisympäristö

EvoMoodle ► Kurssien arkistointi Olet kirjautunut nimellä Yläpito Käyttäjä. (Kirjautu ulos)

Arkistosta haku

Löydetty arkistoidut kurssit

Koko nimi	Lyhenne	Arkistointipäivä
Koodikurssi_2010	KK-17_10	24.01.2010, 12:47
A-Harjoittelukurssi 1	AVTK7001	29.11.2009, 13:36
palo kopio 4	kunta_4	24.01.2010, 11:00
Harjoittelukurssi 7 kopio 2	vtk1007_2	13.02.2010, 12:56
Harjoittelukurssi 7 kopio 1	vtk1007_1	13.02.2010, 12:57
Harjoittelukurssi 7 kopio 5	vtk1007_5	13.02.2010, 12:59
Harjoittelukurssi 7 kopio 6	vtk1007_6	13.02.2010, 13:01
Koodin alku	KA-17	21.02.2010, 09:26
Harjoittelukurssi 7 kopio 7	vtk1007_7	22.02.2010, 18:46

Tarkenna hakua Aloita haku alusta

Tämä sivu Moodle Docs -sivustolla

EvoMoodle ► Kurssien arkistointi

Kuva 14. Hakutulostusta

Kun on löydetty arkistosta kurssi, joka halutaan palauttaa ja tämä on avattu, valitaan 'Palauta arkistosta' -painike ja kurssin palautus käynnistyy. Palautuksen aikana näytölle tulostetaan palautuksen etenemisen tila. Onnistuneesta palautuksesta tiedotetaan palautuksen lopuksi. Mikäli palautus päättyy virheeseen, ilmoitetaan tästä virhe ilmoituksella ja palautus keskeytetään. Palautuksen jälkeen kurssi jää edelleen arkistoon.

Haluttaessa luoda arkistossa olevasta kurssista uusi kurssi, edetään kuten edellä aina palautuksen käynnistykseen. Uuden kurssipohjan luonti aloitetaan 'Palauta uutena kurssipohjana' -painikkeella. Tämän siirrytään lomakkeelle, jossa annetaan kurssille uusi nimi, lyhenne ja kurssikategoria (kuva 16). 'Luo uusi kurssi' -painikkeella käynnistetään palautusprosessi, jonka tuloksena syntyy uusi kurssipohja valitusta kurssista ilman opettajia ja opiskelijoita. Opiskelijoiden tiedot eivät siirry kurssipohjaan.



**Kymenlaakson
ammattikorkeakoulu**
University of Applied Sciences



**Moodle-
oppimisympäristö**

EvoMoodle ► Kurssien arkistointi
Olet kirjautunut nimellä Ylläpito Käyttäjä. (Kirjaudu ulos)

Näytä valitun kurssin tiedot

Näytä arkistoidun kurssin tiedot

Koko nimi: Funktionaalinen ohjelmointi
 Lyhenne: FOK007
 Katogoria: Ohjelmointi
 Arkistoinnin suoritti: Ylläpito Käyttäjä
 Arkistointi suoritettu: 20.04.2010, 20:04
 Arkistotiedoston nimi: 2010S/ varmuuskopio-fok007-20100420-2004.zip
 Pyynnön esittäjä: Mika Hatakka

Opettajat

Opettajat: Mika Hatakka
 Siru Särö
 Vierailevat opettajat: Teppo Töpö

Opiskelijat

Opiskelijat: Tahvo Niitti
 Otto Opiskelija

Kurssin palautus

Palauta arkistosta
Palauta uutena kurssipohjana

Tarkenna haku
Palaa tulostistaan


📘 Tämä sivu Moodle Docs -sivustolla

EvoMoodle ► Kurssien arkistointi

Kuva 15. Palautettavaksi valitun kurssin tiedot



**Kymenlaakson
ammattikorkeakoulu**
University of Applied Sciences



**Moodle-
oppimisympäristö**

EvoMoodle ► Kurssin palautus arkistosta
Olet kirjautunut nimellä Ylläpito Käyttäjä. (Kirjaudu ulos)

Kurssin palautus arkistosta

Näytä arkistoidun kurssin tiedot

Kurssin nimi

Lyhenne

Kurssikategoriat

Luo uusi kurssi
Peruuta

📘 Tämä sivu Moodle Docs -sivustolla

EvoMoodle ► Kurssin palautus arkistosta

Kuva 16. Kurssipohjaksi palautuksen uusien tietojen kysely

KURSSIEN ARKISTOINTILOHKO
OHJELMAN TEKNINEN KUVAUS
Mika Hatakka
Kymenlaakson ammattikorkeakoulu

SISÄLLYS

1	KURSSIEN ARKISTOINTILOHKO	7
2	LOHKOHAKEMISTO ARCHIVE	7
2.1	Tiedosto block_archive.php	7
2.1.1	Yleistä	7
2.1.2	Metodi init()	8
2.1.3	Metodi has_config()	8
2.1.4	Metodi get_content()	8
2.1.5	Metodi add_request_link(\$where, \$admin = false)	9
2.1.6	Metodi admin_content()	9
2.1.7	Metodi admin_manager()	9
2.2	Pyynnön talletussivu request_sen.php	9
2.3	Asetuslomake settings.php	9
2.4	Metodikirjasto archivelib.php	9
2.4.1	Yleistä	9
2.4.2	check_parameter(\$param, \$default=null)	10
2.4.3	get_archive_directory()	10
2.4.4	function get_useable_categories()	10
3	LOMAKETIEDOSTOT LOHKOHAKEMISTOSSA ARCHIVE	10
3.1	Yleistä	10
3.2	Luokkatiedosto archive_info_form.php	11
3.2.1	Muodostin __construct()	11
3.2.2	Metodi definition()	11
3.2.3	Metodi definition_after_data()	11
3.3	Määrittystiedosto archive_info.php	11
3.4	Luokkatiedosto archive_select_form.php	12
3.4.1	Metodi definition()	12
3.5	Määrittystiedosto archive_select.php	12
3.6	Luokkatiedosto archive_selection_form.php	12
3.6.1	Metodi definition()	12

3.6.2 Metodi definition_after_data()	12
3.7 Määrittystiedosto archive_selection.php	12
3.8 Lomaketiedosto archive_do.php	13
3.9 Luokkatiedosto archive_search_form.php	13
3.9.1 Metodi definition()	13
3.9.2 Metodi definition_after_data()	13
3.10 Määrittystiedosto archive_search.php	13
3.11 Luokkatiedosto result_list_form.php	14
3.11.1 Muodostin __construct(\$table = ")	14
3.11.2 Metodi definition()	14
3.12 Määrittystiedosto result_list.php	14
3.13 Luokkatiedosto search_result_form.php	14
3.13.1 Metodi definition()	14
3.13.2 Metodi definition_after_data()	14
3.14 Määrittystiedosto search_result.php	15
3.15 Luokkatiedosto archive_restore_form.php	15
3.15.1 Metodi definition()	15
3.15.2 Metodi definition_after_data()	15
3.16 Määrittystiedosto archive_restore.php	15
3.17 Lomaketiedosto restore_do.php	15
4 LUOKKAHAKEMISTO ARCHIVE/CLASSES	15
4.1 Yleistä	15
4.2 Luokkatiedosto requestdb.php	16
4.2.1 Yleistä	16
4.2.2 Muodostin __construct(\$courseid = null)	16
4.2.3 Metodi request_count()	16
4.2.4 Metodi remove_by_courseid(\$courseid)	16
4.2.5 Metodi remove_forced()	16
4.2.6 Metodi get_display(\$userid)	17
4.2.7 Metodi insert_item(\$courseid, \$userid, \$state)	17
4.2.8 Metodi delete_item(\$courseid, \$userid)	17

4.2.9	Metodi update_state(\$state, \$courseid, \$userid)	17
4.2.10	Metodi get_item(\$userid, \$courseid = null)	17
4.2.11	Metodi get_item_by_id(\$id)	17
4.2.12	Metodi get_all_items()	17
4.2.13	Metodi get_action_items()	17
4.2.14	Metodi get_other_items(\$courseid, \$userid)	17
4.2.15	Metodi get_who(\$what)	18
4.2.16	Metodi get_timecreated(\$userid)	18
4.2.17	Metodi get_task(\$userid)	18
4.2.18	Metodi check_request()	18
4.3	Luokkatiedosto archivecourse.php	18
4.3.1	Yleistä	18
4.3.2	Muodostin __construct(\$course, \$request)	18
4.3.3	Metodi get_archived_categories(\$addlineall = true)	18
4.3.4	Metodi update_archived_categories(\$id, \$name)	19
4.3.5	Metodi move_to_archive()	19
4.3.6	Metodi remove_course()	19
4.3.7	Metodi update_archive_filename(\$filename)	19
4.3.8	Metodi read_archivedata(\$id)	19
4.3.9	Metodi get_archivedata()	19
4.3.10	Metodi send_confirmation_email(\$arkistoiija)	19
4.3.11	Metodi get_record()	19
4.3.12	Metodi fill_object(\$course, \$request)	20
4.3.13	Metodi write_to_db()	20
4.3.14	Metodi update_coursearchive()	20
4.3.15	Metodi delete()	20
4.3.16	Metodi archive_exists(){	20
4.3.17	Metodi restore_write(\$where)	20
4.4	Luokkatiedosto courseuser.php	20
4.4.1	Muodostin __construct(\$user, \$archiveid = null)	20
4.4.2	Metodi write_to_db(\$tablename)	21
4.4.3	Metodi read_from_db(\$tablename, \$id)	21
4.4.4	Metodi delete_by_archive(\$archiveid, \$tablename)	21
4.4.5	Metodi set_archiveid(\$id)	21

4.4.6	Metodi get_id()	21
4.5	Luokkatiedosto courseteacher.php	21
4.5.1	Muodostin __construct(\$user = null, \$archiveid = null)	21
4.5.2	Metodi get_course_teachers(\$context, \$userid = null)	21
4.5.3	Metodi get_teachers(\$courseid, \$reqteacher, \$type = self::EDITINGTEACHER)	22
4.5.4	Metodi get_archives_teachers(\$archiveid)	22
4.5.5	Metodi write_to_db()	22
4.5.6	Metodi read_from_db(\$id)	22
4.5.7	Metodi delete_by_archive(\$archiveid)	22
4.6	Luokkatiedosto coursestudent.php	22
4.6.1	Muodostin __construct(\$user = null, \$archiveid = null)	22
4.6.2	Metodi get_course_students(\$context, \$userid = null)	22
4.6.3	Metodi write_to_db()	23
4.6.4	Metodi read_from_db(\$id)	23
4.6.5	Metodi delete_by_archive(\$archiveid)	23
4.7	Luokkatiedosto archivebackup.php	23
4.7.1	Muodostin __construct(\$course)	23
4.7.2	Metodi get_prefs()	23
4.7.3	Metodi show_info()	23
4.7.4	Metodi execute_backup()	24
4.7.5	Metodi check_backup()	24
4.7.6	Metodi create()	24
4.7.7	Metodi check_archive_subdir()	24
4.7.8	Metodi create_archive_dir(\$tunnus)	24
4.8	Luokkatiedosto archiverestore.php	24
4.8.1	Muodostin __construct(\$id = null)	24
4.8.2	Metodi get_archivedata()	25
4.8.3	Metodi set_restore_preferences(\$to_link = null)	25
4.8.4	Metodi update_restore_preferences(\$userdata = 1)	25
4.8.5	Metodi update_restore_preferences_new(\$newcoursedata)	25
4.8.6	Metodi execute_restore()	25
4.8.7	Metodi parse_param(\$var)	25

4.8.8	Metodi set_php_memory_settings()	25
4.8.9	Metodi cleanup()	26
4.9	Luokkatiedosto archivesearch.php	26
4.9.1	Muodostin __construct(\$sdata)	26
4.9.2	Staattinen metodi get_archived_count()	26
4.9.3	Metodi find()	26
4.9.4	Metodi get_result_table(\$url)	26
4.9.5	Metodi get_nextprev_table()	26
4.9.6	Metodi set_limits(\$firstrow)	26
4.9.7	Metodi is_multi_page()	26
4.9.8	Metodi get_cntresult()	27
4.9.9	Metodi get_first_archive_id()	27
4.9.10	Metodi add_data(\$par, \$val)	27
4.9.11	Metodi make_search_sql()	27
4.9.12	Metodi add_needed_and(&\$sql, \$param = "AND")	27
4.9.13	Metodi parse_name(\$name)	27
4.10	Luokkatiedosto archiveform.php	27
4.10.1	Muodostin __construct(\$restrict = true)	27
4.10.2	Metodi definition()	28
4.10.3	Metodi show_with_navigation(\$title, \$course = null, \$caption = null, \$title_extra = "")	28
4.10.4	Metodi start_page_with_navigation(\$title, \$course = null,	28
4.10.5	Metodi end_page(\$course = null)	28
4.10.6	Metodi start_nav_box_page(\$title, \$course = null,	28
4.10.7	Metodi end_box_page(\$course = null)	28
4.10.8	Metodi check_rights()	28
5	TIETOKANTAHAKEMISTO ARCHIVE/DB	28
5.1	Kuvaustiedosto install.xml	28
5.2	Oikeustiedosto access.php	29
5.3	Kuvaustiedosto upgrade.php	29
6	KIELIHAKEMISTO ARCHIVE/LANG	29
6.1	fi_utf8/block_archive.php	29
6.2	en_utf8/block_archive.php	29

1 KURSSIEN ARKISTOINTILOHKO

Tämän dokumentin on tarkoitus olla johdatus ohjelman rakenteeseen ja osiin, joista se koostuu. Dokumentin avulla on helppo löytää metodien tiedosto ja sijainti hakemistorakenteessa. Tiedostoista, luokista ja metodeista on annettu yleiskuvaus, tyhjentävää esitystä ei ole yritetty lähteä hakemaan. Metodien toiminnallisuus selviää lähdekoodia tutkimalla.

Ohjelman hakemistorakenne noudattaa Moodlen perusrakennetta. Hakemistonimi on samalla lohkon nimi ja Moodle käyttää tätä tietoa kaikkeen tiedon vaihtoon lohkon kanssa. Samoin db ja lang -hakemistot ovat Moodlen määritysten mukaiset. Omat luokat on selvyuden vuoksi sijoitettu classes-hakemistoon. Luokkien sijoitteluun ei Moodlen dokumentaatiossa ollut määrättyä hakemistoa.

Arkistointilohko on toteutettu hyödyntäen Moodle 1.9.x version backup- ja restore-toimintoja, joten toiminta seuraavan suuremman versiopäivityksen yhteydessä huolellisesti varmistettava. Suuria muutoksia tuskin on odotettavissa, mutta pientenkin muutosten vaikutukset voivat seuraamuksiltaan vakavia. Lisäksi on käytetty Moodlen tietokantakirjastoja, joiden kohdalla tuskin tulee ongelmia, koska ne on toteutettu joustavammiksi, eikä tarvetta muutoksiin ole. Tarvittaessa uudet ominaisuudet voidaan lisätä uusina metodeina.

2 LOHKOHAKEMISTO ARCHIVE

2.1 Tiedosto block_archive.php

2.1.1 Yleistä

Lohkon ydin tiedosto, joka luo lohkon sisällön ja ohjaa toiminnan toteuttavilla lomakkeille. Lohko käyttää Moodlen tietokantakirjastoa (ddllib.php) tietokanta toimintoihin. Lisäksi käytetään arkistointi moduulin requestdb ja courseteacher -luokkia sekä yhteisiä metodeja sisältää archivelib.php-kirjastotiedostoa. Arkistointilohko muodostetaan periyttämällä liityntärajapinta Moodleen perusluokasta block_base.

2.1.2 Metodi init()

Lohkon alustus metodi, jossa määritellään lohkon nimi ja versio. Versio tietoa tarvitaan päivitysversioiden tunnistukseen.

2.1.3 Metodi has_config()

Metodi palauttaa arvon true, jolla kerrotaan Moodlelle, että lohkolla on yleisiä asetuksia. Asetukset asetetaan settings.php -lomakkeella.

2.1.4 Metodi get_content()

Metodi on lohkon keskeisin, sillä siinä luodaan lohkon näkymä. Näkymä muodostamisessa käytetään hyväksi nykyisen sivun ja käyttäjän tietoja. Metodi käyttää globaaleja muuttujia \$CFG ja \$USER näkymän muodostamiseen. Näkymä palautetaan stdClass-objektina tai tyhjänä merkkijonona, mikäli näkymää ei näytetä.

Sivun tiedoista tarkistetaan, onko sisältö jo olemassa, ollaanko kurssisivulla ja onko kurssi olemassa. Käyttäjistä tarkistetaan, että on opettajana kurssilla tai järjestelmänvalvoja. Arkistolohkon osalta tarkastetaan, että arkistointilohkon pyyntötaulu on olemassa. Mikäli näitä ehtoja ei täytetä, ei lohkoa näytetä. Käyttäjän ollessa järjestelmänvalvoja, tarkistetaan onko hän opettajana kurssilla vai ei ja lohkon näkymä muodostetaan tämän mukaan.

Näkymän muodostuksen monimutkaisin osa on valittavissa olevien toimenpiteiden määrittäminen. Lohko aloitetaan lyhyellä opasteella. Yhden opettajan kurssilla vaihtoehtoja ovat kurssin arkistointi tai poisto pyynnön esitys ja valinnan jälkeen peruutus mahdollisuus, kunnes pyynnöt on käsitelty. Useamman opettajan kurssilla tulevat mukaan vaihtoehdot pyynnön hyväksyntä tai kieltäminen. Lisäksi on huomioitava vierailevien (toissijaisten) opettajien tiedonannot. Järjestelmänvalvojalla on Pakota kurssin arkistointi -valinta, joka on myös vaihtoehtona hänen ollessaan kurssin opettajana. Lopuksi lisätään vielä lisätietoja-linkki, josta aukeaa ohjesivu opettajalle.

2.1.5 Metodi add_request_link(\$where, \$admin = false)

Palauttaa linkin määriteltyyn ositteeseen, huomioiden onko järjestelmänvalvojan vai opettajan lohko kyseessä. Linkin otsikko käännetään käytetylle kielelle.

2.1.6 Metodi admin_content()

Luo järjestelmänvalvojan näkymän. Aluksi tarkastetaan, että järjestelmänvalvojalla on oikeus arkistoida kurseja. Tämän jälkeen haetaan arkistoitavien kurssien lukumäärä ja luodaan linkki arkistoinnin aloitukseen ja kerrotaan montako kurssia odottaa arkistointia. Mikäli arkistoitavia kurseja ei ole odottamassa, ilmoitetaan tämä tekstikentällä. Seuraavaksi luodaan linkki arkistoitujen kurssien hakuun, jonka aloitetaan arkistosta palautus. Lopuksi palautetaan näkymä stdClass-objektina.

2.1.7 Metodi admin_manager()

Luo kurssien arkistointilohkoon linkin, jolla järjestelmänvalvoja voi suorittaa pakotetun kurssin arkistoinnin. Tämän avulla järjestelmänvalvoja pystyy siivoamaan unoh-tuneita tai orpoja kurseja.

2.2 Pyynnön talletussivu request_sen.php

Sivulla varmistetaan, että käyttäjä on kurssilla opettajana ja tämän jälkeen pyyntö kirjataan arkistointilohkon pyyntötietokantaa ja ohjataan toiminta takaisin kurssisivulle.

2.3 Asetuslomake settings.php

Luodaan lomake arkistointilohkon asetusten asetusta varten. Moodle osaa automaattisesti hakea lohkon hakemistosta settings.php -nimellä.

2.4 Metodikirjasto archivelib.php

2.4.1 Yleistä

Metodikirjastoon archivelib.php on kerätty kaikille lomakkeille ja luokille yhteisiä metodeja. Nämä yhteiset metodit eivät kuulu mihinkään yksittäiseen luokkaan, joten ne on ollut parempi sijoittaa omaan metodikirjastoon.

2.4.2 check_parameter(\$param, \$default=null)

Tarkastaa onko parametri asetettu ja palauttaa tämän. Mikäli parametria ei ole asetettu palautetaan annettu oletus arvo tai mikäli oletusta ei ole annettu palautetaan ”null”.

2.4.3 get_archive_directory()

Palauttaa arkistohakemiston, tutkii onko sijoituspaikka järjestelmässä vai moodledata-hakemiston alla.

2.4.4 function get_useable_categories()

Palauttaa taulukon Moodlen käytössä olevista kategorioista.

3 LOMAKETIEDOSTOT LOHKOHAKEMISTOSSA ARCHIVE

3.1 Yleistä

Lomakeluokilla perustuvat archiveform-luokkaan, johon on kerätty kaikille lomakkeille yhteiset piirteet ja toiminnot. Lomakkeella näytettävät komponentit määritellään definition-metodissa. Muodostinta __construct käytetään lähinnä tarvittaessa peruslomakkeesta erikoisversioita, kuten esimerkiksi kaikille näytävissä lomakkeissa. Metodissa definition_after_data määritellään lomakkeen ominaisuudet, jotka ovat riippuvaisia lomakkeelle tuotavasta tiedosta. Käytettäessä definition_after_data-metodia, on aina ensimmäiseksi kutsuttava perusluokan vastaavaa parent::definition_after_data()-metodia.

Lomake luodaan ja siitä ohjataan eteenpäin kolmen perustarkastuksen kautta. Ensin tarkastetaan onko valittu peruutus. Seuraavaksi tarkistetaan välitetäänkö lomakkeelta tietoa. Kolmas vaihtoehto on lomakkeen luonti näytettäväksi. On tapauksia, joissa välitettävää tietoa esikäsitellään ja tällöin lomakkeen luonti ei tapahdu if-silmukassa vaan sen ulkopuolella, jolloin siihen mennään myös tilanteissa, jossa peruutus tai hyväksyntä painiketta on painettu, eikä näistä ole siirrytty eteenpäin.

```
if($form->is_cancelled()) {
    // Lomakkeen tietojen asetus peruttu
} else if(($data = $form->get_data()) !== null) {
```

```
        // Lomakkeella asetettua tietoa
    } else {
        // Lomakkeen luonti näytettäväksi
    }
```

3.2 Luokkatiedosto archive_info_form.php

3.2.1 Muodostin __construct()

Muodostimessa annetaan perusluokalle parametrina tieto, ettei sivun näyttäminen vaadi erityisiä oikeuksia, vaan kuka tahansa kirjautunut käyttäjä saa nähdä sen.

3.2.2 Metodi definition()

Metodissa lomakkeelle lisätään piilotettu kenttä muuttujalle "id" sekä painike takaisin kurssille siirtymistä varten.

3.2.3 Metodi definition_after_data()

Mikäli lomakkeelle on määritelty muuttujalla "table" näytettävää tietoa, lisätään lomakkeelle muuttujan sisältämä html-muotoinen sisältö.

3.3 Määrittystiedosto archive_info.php

Muodostaa opastussivun lohkon toiminnasta opettajille. Opastesivu näytetään kaikille kirjautuneille, mutta muiden kuin kurssien opettajien täytyy tällöin tietää sivun tarkka URL.

Opastetiedot sivulle luodaan perus HTML hyväksikäyttäen. Tulostettavat teksti ajetaan käännösmetodin get_string kautta, ja näin saadaan uusien kielten lisäys yksinkertaiseksi. Arkistoijan sähköpostiosoite luetaan Moodlen asetustiedoista, jonne asetussivulla on lisätty. Paluu-painikkeella palataan takaisin kurssille, josta sivulle saavuttiin.

3.4 Luokkatiedosto archive_select_form.php

3.4.1 Metodi definition()

Luo lomakkeella pudotusvalikon, josta arkistoitava kurssi valitaan ja Valinta-painikkeeseen, jolla siirrytään arkistoinnin seuraavaan vaiheeseen.

3.5 Määrittystiedosto archive_select.php

Muodostaa ilmentymän archive_select_form -luokasta ja näyttää lomakkeen. Ohjaa lomakkeelta saadun tiedon mukaan toiminnan eteenpäin.

3.6 Luokkatiedosto archive_selection_form.php

3.6.1 Metodi definition()

Luodaan lomakkeelle arkistoivan kurssin perustietojen näyttämiseen tarvittavat komponentit sekä arkistoinnin jatkamiseen tai peruttamiseen painikkeet.

3.6.2 Metodi definition_after_data()

Poistetaan opettajien ja toissijaisten opettajien näyttämiseen varatut kentät mikäli näitä ei määritelty.

3.7 Määrittystiedosto archive_selection.php

Aluksi tarkistetaan, että on annettu todellinen kurssi, joka halutaan arkistoida ja tämän jälkeen varmistetaan, että kurssi on myös arkistoitavien listalla. Näiden ehtojen täytyessä haetaan ja näytetään arkistoitavaksi valitun kurssin tiedot.

Valittaessa lomakkeella toiminnon peruutus, tyhjennetään pyyntötietokannasta pakko arkistoitavat kurssit. Kurssin pakko arkistointi suoritetaan suoraan, eikä niitä jätetä tietokantaan odottamaan. Valittaessa arkistoinnin aloitus tarkistetaan, onko toimenpide ollut arkistointi vai poisto. Poiston ollessa kyseessä ohjataan toiminta Moodlen omaan poisto toimintoon. Arkistoinnin ollessa kyseessä ohjataan toiminta lomakkeelle, jossa varsinainen arkistointi suoritetaan.

3.8 Lomaketiedosto archive_do.php

Lomake archive_do tulostaa suoraa raporttia toiminnan edetessä, tästä syystä sille ei ole luotu omaa lomakeluokkaa, vaan on johdettu suoraan luokasta archiveform. Toiminta aloitetaan varmistamalla, että arkistoita kurssi on oikea kurssi ja tämän jälkeen haetaan kurssin tiedot. Kurssin tietoja käytetään muodostamaan ilmentymä arkistointiluokasta archivecourse, joka huolehtii arkistointitietokannasta. Varsinaista arkistointia varten luodaan ilmentymä archive_backup-luokasta. Kun arkistointi on suoritettu onnistuneesti, päivitetään tiedot arkistointitietokantaan, lähetetään ilmoitus kurssin opettajille ja poistetaan kurssi. Toimista tulostetaan näytölle reaaliaikainen raportti.

3.9 Luokkatiedosto archive_search_form.php

3.9.1 Metodi definition()

Luodaan kurssin haun lomake. Arkistossa olevien kategorioiden nimet haetaan käyttäen archivecourse-luokan staattista metodia get_archived_categoriesd(). Lomakkeella käytetään suoraan MoodleQuickForm-luokan metodeja luotaessa radiopainike -valinta etsitäänkö nimellä vai käyttäjätunnuksella. Samoin luokkaa on käytetty luotaessa opettajien ja opiskelijoiden hakukenttien lisäys mahdollisuus. Toimintopainikkeet on myös luotu yksitellen, koska tarvitsemme laajemmat mahdollisuudet kuin peruspainikkeiden lisäys metodi sallii.

3.9.2 Metodi definition_after_data()

Poistaa ”Ei tuloksia” tekstin tilanteissa, joissa on hakutuloksia.

3.10 Määrittystiedosto archive_search.php

Hakulomakkeen muodostus aloitetaan tyhjentämällä searchdata -istuntomuuttuja. Haku ja palautus toiminnossa käytetään istuntomuuttujaa tietojen välitykseen. Seuraavaksi tutkitaan, mitä kolmesta painikkeesta on painettu ja edetään sen mukaisiin toimintoihin.

3.11 Luokkatiedosto result_list_form.php

3.11.1 Muodostin __construct(\$table = ")

Muodostimessa asetetaan luokan yksityiseen resulttable-muuttujaan välitetty tulos-
taulukko.

3.11.2 Metodi definition()

Luodaan lomakekentät htm-muodossa perustuen luokan resulttable-muuttujaan. Toi-
mintopainikkeet luodaan yksitellen lomakkeen tarpeiden mukaa.

3.12 Määrittystiedosto result_list.php

Toiminta aloitetaan tutkimalla onko haku tuottanut tuloksia. Tuloksista muodostetaan
archivesearch-luokan avulla htm-taulukko, johon tulokset on sijoitettu. Taulukko vä-
litetään lomake-luokan muodostimelle lomaketta luotaessa. Loput toimet ohjaavat
toiminnan eteenpäin lomakkeelta välitetyn tiedon mukaan.

3.13 Luokkatiedosto search_result_form.php

3.13.1 Metodi definition()

Luodaan arkistoidun kurssin tietojen näkymä. Toimintopainikkeet luodaan yksitellen
lomakkeen tarpeiden mukaa. Painikkeet on jaettu kahteen ryhmään, ensimmäisessä
ovat palautus painikkeet ja toisessa normaalit paluu painikkeet.

3.13.2 Metodi definition_after_data()

Poistetaan vierailevien opettajien näyttökomponentti, mikäli kurssilla ei näitä ole.
Tarkastetaan onko tultu suoraan hausta, tällöin asetetaan "Paluu listalle" -painike ei
valittavissa olevaan tilaan.

3.14 Määrittystiedosto search_result.php

Näytetään palautettavaksi valitun kurssin tiedot ja lomakkeelta tulevan tiedon mukaan ohjataan eteenpäin. Valittaessa toinen palautus vaihtoehdoista, tutkitaan kumpi valittu ja tämän mukaan suoritetaan palautuksen esivalmistelut ennen eteenpäin ohjausta.

3.15 Luokkatiedosto archive_restore_form.php

3.15.1 Metodi definition()

Luodaan kysely lomakkeella malli, jolla pyydetään arvot, mikäli kurssista palautetaan uutta kurssipohjaa.

3.15.2 Metodi definition_after_data()

Poistetaan ”Lyhytnimi käytössä” ja ”Ei kategorialla” -ilmoitukset mikäli näitä ei ole asetettu.

3.16 Määrittystiedosto archive_restore.php

Luo tarvittaessa palautuksessa tarvittavien tietojen kyselylomakkeen. Suoritettaessa alkuperäisen suoraa palautusta toiminta ohjataan suoraan eteenpäin. Palautettaessa uudeksi kurssipohjaksi suoritetaan kysely uusista tiedoista ja tämän jälkeen jatketaan eteenpäin.

3.17 Lomaketiedosto restore_do.php

Aloitetaan palautuksen suoritus luomalla ilmentymä archiverestore-luokasta. Palautus prosessi käynnistetään luokan execute-metodilla, joka palauttaa linkin eteenpäin, jos palautus onnistui tai virheilmoituksen epäonnistuessaan.

4 LUOKKAHAKEMISTO ARCHIVE/CLASSES

4.1 Yleistä

Luokat sisältävät ohjelman varsinaisen toiminnallisuuden. Luokista courseuser on abstraktiluokka, johon on kerätty kurseille osallistuvien yhteiset ominaisuudet. Lo-

makkeiden perusluokka archiveform on tarkoitettu pääasiassa pohjaksi muille lomakkeille, mutta toteutettu siten, että siitä voidaan johtaa tarvittaessa ilmentymä myös suoraan. Luokat hyödyntävät varsin laajasti Moodlen varmistus ja palautus toimintoja.

4.2 Luokkatiedosto requestdb.php

4.2.1 Yleistä

Luokka huolehtii tietokannan pyyntötaulun kirjoitus- ja lukutoiminnoista. Pyyntötauluun kerätään opettajien arkistointi- ja poistopyynnot kurseista. Järjestelmänvalvojan pakotetut arkistointipyynnotkin kirjataan hetkellisesti. Pakotetut pyynnot poistetaan, mikäli niitä ei heti toteuteta. Pyynnot poistetaan taulusta sen jälkeen, kun ne on toteutettu.

4.2.2 Muodostin __construct(\$courseid = null)

Annettaessa kurssitunnus muodostimelle, luetaan pyyntötaulusta kaikki kurssiin liittyvät pyynnot luokan sisäiseen muuttujaan.

4.2.3 Metodi request_count()

Staattinen metodi palauttaa odottamassa olevien kurssien lukumäärän, samaan kurssiin viittaavista pyynnöistä huomioidaan ainoastaan yksi eikä pakotettuja pyyntöjä lasketa.

4.2.4 Metodi remove_by_courseid(\$courseid)

Staattinen metodi poistaa kaikki annettuun kurssiin viittaavat tietueet pyyntötaulusta.

4.2.5 Metodi remove_forced()

Staattinen metodi poistaa pyyntötaulusta kaikki tietueet, jotka ovat pakotettuja arkistointeja. Niitä ei kerätä tietokantaan, vaan ovat vain merkinä toimenpiteen aloittamisesta.

4.2.6 Metodi get_display(\$userid)

Valitsee kurssin arkistoinnin tilan mukaan näytettävän moodin. Moodiin vaikuttavat opettajien määrä ja tyyppi sekä kuinka monta opettajaa kurssilla on.

4.2.7 Metodi insert_item(\$courseid, \$userid, \$state)

Lisää pyyntötauluun uuden tietueen.

4.2.8 Metodi delete_item(\$courseid, \$userid)

Poistaa pyyntötaulusta tietueet, jotka täyttävät annetut ehdot.

4.2.9 Metodi update_state(\$state, \$courseid, \$userid)

Muuttaa tieteen kurssiin kohdistuvan toimenpiteen.

4.2.10 Metodi get_item(\$userid, \$courseid = null)

Metodi hakee ehdot täyttävät tietueet.

4.2.11 Metodi get_item_by_id(\$id)

Hakee tunnuksen mukaisen tietueen.

4.2.12 Metodi get_all_items()

Metodi palauttaa kaikki tietueet pyyntötaulusta.

4.2.13 Metodi get_action_items()

Palauttaa pyyntötaulusta kaikki tietueet, joissa toimenpide on arkistointi tai poisto.

4.2.14 Metodi get_other_items(\$courseid, \$userid)

Palauttaa pyyntötaulusta kaikki tietueet, joissa viittaus annettuun kurssiin, eivätkä ole annetun käyttäjän.

4.2.15 Metodi `get_who($what)`

Palauttaa ensimmäisen tavatun käyttäjän, joka on esittänyt pyynnön annettuun kurssiin.

4.2.16 Metodi `get_timecreated($userid)`

Palauttaa annetun käyttäjän toimenpiteen luontiajan ilmentymässä olevasta pyyntötietue joukosta.

4.2.17 Metodi `get_task($userid)`

Palauttaa annetun käyttäjän toimenpiteen ilmentymässä olevasta pyyntötietue joukosta.

4.2.18 Metodi `check_request()`

Tarkistaa tietokannasta, ettei ole kurssi tietueita, joihin ei liittyisi toimenpide tietueita. Orvot kurssi tietueet ilman toimenpide tietuetta poistetaan.

4.3 Luokkatiedosto `archivecourse.php`

4.3.1 Yleistä

Luokka ylläpitää arkistotietokantaa, johon kuuluvat tiedot arkistoivan kurssin tiedoista, kurssilla olevien opettajien ja opiskelijoiden tiedoista sekä arkistossa olevista kurssikategoriasta.

4.3.2 Muodostin `__construct($course, $request)`

Luokan muodostimessa haetaan tieto-objektiin arkistoivan kurssin tiedot, mikäli annetut kurssitunnus ja pyyntötunnus ovat olemassa, muuten luodaan tyhjä tieto-objekti.

4.3.3 Metodi `get_archived_categories($addlineall = true)`

Staattinen metodi, joka lukee tietokannasta arkistoitujen kurssikategorioiden nimet.

4.3.4 Metodi update_archived_categories(\$id, \$name)

Lisää uuden kurssikategorian arkistotietokantaan, mikäli sitä ei siellä ole.

4.3.5 Metodi move_to_archive()

Ensin tarkastetaan, onko kurssitunnuksella jo arkistoitu, näin ollessa palautetaan virheilmoitus. Muuten kirjoitetaan ilmentymän tieto-objektin tiedot arkistotietokantaan. Epäonnistumisesta palautetaan virheilmoitus.

4.3.6 Metodi remove_course()

Poistetaan kurssi Moodlesta ja lisätään lokiin poisto. Lopuksi poistetaan pyyntö pyyntötaulusta.

4.3.7 Metodi update_archive_filename(\$filename)

Päivittää arkiston tiedostonimen tieto-objektiin ja arkistotietokantaan.

4.3.8 Metodi read_archivedata(\$id)

Luetaan arkistotietokannasta arkistoidun kurssin tiedot sekä tiedot opettajista ja opiskelijoista ilmentymän tieto-objektiin.

4.3.9 Metodi get_archivedata()

Muodostaa tieto-objektin tiedoista tulosvedoksen näytettäväksi hakutulossivulle.

4.3.10 Metodi send_confirmation_email(\$arkistoija)

Lähetetään arkistoidun kurssin opettajille sähköposti-ilmoituksen kurssin arkistointitoimenpiteen toteuttamisesta.

4.3.11 Metodi get_record()

Metodi palauttaa ilmentymän tieto-objektin tiedot.

4.3.12 Metodi fill_object(\$course, \$request)

Metodissa täytetään ilmentymän tieto-objektin tiedot annetun kurssitunnuksen ja pyyntötunnuksen osoittamista tiedoilla.

4.3.13 Metodi write_to_db()

Suorittaa varsinaisen ilmentymän tieto-objektin tietojen kirjoituksen arkistotietokantaan.

4.3.14 Metodi update_coursearchive()

Metodi päivittää arkistotietokannan tietueen aikaleiman.

4.3.15 Metodi delete()

Poistetaan ilmentymän tieto-objekti arkistotietokannasta.

4.3.16 Metodi archive_exists(){

Metodi tutkii onko ilmentymän tieto-objektin mukaisella kurssitunnuksella jo olemassa oleva tietue arkistotietokannassa.

4.3.17 Metodi restore_write(\$where)

Metodilla peruutetaan tehdyt kirjoitukset arkistotietokantaan. Käytetään virhetilanteissa, joissa ei haluta jättää tietoja toteutumattomista arkistoinneista teitokantaan.

4.4 Luokkatiedosto courseuser.php

On ohjelman ainoa abstraktiluokka. Luokka sisältää käyttäjien yhteiset ominaisuudet ja metodit. Luokassa ei ole abstrakteja metodeja.

4.4.1 Muodostin __construct(\$user, \$archiveid = null)

Muodostimessa asetetaan annetut käyttäjän perustiedot tai jos tietoja ei välitetty, tieto-objekti jää tyhjäksi.

4.4.2 Metodi write_to_db(\$tablename)

Tallettaa tieto-objektin annettuun tietokanta tauluun.

4.4.3 Metodi read_from_db(\$tablename, \$id)

Lukee parametrina annetusta tietokannasta halutun tunnuksen mukaisen tietueen, joka palautetaan paluu arvona.

4.4.4 Metodi delete_by_archive(\$archiveid, \$tablename)

Poistaa annetusta tietokantataulusta kaikki tietueet, jotka viittaavat annettuun arkistotunnukseen.

4.4.5 Metodi set_archiveid(\$id)

Asettaa ilmentymän tieto-objektin arkistotunnuksen.

4.4.6 Metodi get_id()

Palauttaa ilmentymän sisällään pitämän käyttäjän tunnuksen.

4.5 Luokkatiedosto courseteacher.php

Luokka courseteacher on johdettu courseuser-luokasta ja täydentää perusluokkaan opettajasta tarvittavat tiedot.

4.5.1 Muodostin __construct(\$user = null, \$archiveid = null)

Perusluokalle välitettävien tietojen lisäksi otetaan talteen opettajan rooli, varsinainen vai toissijainen opettaja.

4.5.2 Metodi get_course_teachers(\$context, \$userid = null)

Staattinen metodi palauttaa kaikki kurssin opettajat, varsinaiset ja toissijaiset, taulukona Moodlen tietokannasta. Jos käyttäjätunnus on määriteltä, palautetaan vain käyttäjätunnuksen määrittelemä opettaja, mikäli hän on opettajan kurssilla.

4.5.3 Metodi `get_teachers($courseid, $reqteacher, $type = self::EDITINGTEACHER)`

Staattinen metodi palauttaa taulukkona kaikki kurssin opettajat annetusta roolista, pois lukien `reqteacher`-muuttujalla määritelty opettaja.

4.5.4 Metodi `get_archives_teachers($archiveid)`

Staattinen metodi palauttaa kaikki arkistotietokannassa olevat kurssin opettajat.

4.5.5 Metodi `write_to_db()`

Kirjoittaa perusluokan kautta tiedot arkistotietokantaan.

4.5.6 Metodi `read_from_db($id)`

Lukee arkistotietokannasta parametrina annetun käyttäjän tiedot ilmentymän tietobjektiin, joka lopuksi palautetaan kysyvälle metodille.

4.5.7 Metodi `delete_by_archive($archiveid)`

Metodilla poistetaan ilmentymän tiedot arkistotietokannasta.

4.6 Luokkatiedosto `coursestudent.php`

Luokka `coursestudent` on johdettu `courseuser`-luokasta ja täydentää perusluokkaan opiskelijasta tarvittavat tiedot.

4.6.1 Muodostin `__construct($user = null, $archiveid = null)`

Perusluokalle välitetään tiedot opiskelijasta.

4.6.2 Metodi `get_course_students($context, $userid = null)`

Staattinen metodi, jolla haetaan Moodlen tietokannasta kurssin opiskelijat.

4.6.3 Metodi write_to_db()

Kirjoitetaan ilmentymän tiedot arkistotietokantaan.

4.6.4 Metodi read_from_db(\$id)

Lukee arkistotietokannasta parametrina annetun opiskelijan tiedot ilmentymän tietobjektiin, joka lopuksi palautetaan kysyvälle metodille

4.6.5 Metodi delete_by_archive(\$archiveid)

Metodilla poistetaan ilmentymän opiskelijan tiedot arkistotietokannasta.

4.7 Luokkatiedosto archivebackup.php

Luokka suorittaa kurssin varsinaisen arkistoinnin käyttäen hyväksi Moodlen varmistusrutiineja. Näin aikaan saatu arkistotiedosto on mahdollista palauttaa Moodlen omilla palautusrutiineilla, mikäli arkistosta palautus ei ole käytettävissä. Arkistotiedosto sijoitetaan arkistohakemistoon, joka voidaan sijoittaa minne tahansa levyjärjestelmään tai Moodlen data-hakemistoon, mikäli muihin hakemistoihin ei ole oikeuksia.

4.7.1 Muodostin __construct(\$course)

Muodostimessa liitetään arkistoiva kurssi ilmentymään, tyhjennetään vanhat tiedot istunnosta ja tarkastetaan, että arkistohakemisto on ajan tasalla, hakemistot vuosittain ja syksyille ja keväälle vielä omat hakemistot. Ympäristö muuttujat asetetaan ja luodaan asetuksille objekti ja luodaan perustiedot arkistointia varten kurssista.

4.7.2 Metodi get_prefs()

Metodi palauttaa ilmentymään kerätyt luettelointi tiedot arkistoitavasta kurssista.

4.7.3 Metodi show_info()

Metodilla näytetään tiedot arkistoitavasta kurssista ja täyttää samalla luettelointia. Näytettävä taulukko kerätään table-muuttujaan, joka palautetaan kutsuvalle metodille. Kutsuva metodi voi näin toimittaessa näyttää tiedot tai jättää näyttämättä. Luetteloin-

nin kartuttamisen takia täytyy kutsua. Metodin perusrakenne noudattaa Moodlen toiminta malleja.

4.7.4 Metodi execute_backup()

Metodi suorittaa varsinaiseen arkistoinnin. Luo raamit raportille ja siirtyy käyttämään Moodlen varmistus metodia.

4.7.5 Metodi check_backup()

Arkistoinnin tarkastus metodi, jossa tutkitaan onko arkistotiedosto olevassa halutussa hakemistossa.

4.7.6 Metodi create()

Metodi käy läpi Moodlessa käytettävissä olevat aktiviteetit ja luo kurssin käyttämistä arkistointiluettelon, joka talletetaan istuntopuuttuun.

4.7.7 Metodi check_archive_subdir()

Metodi tarkistaa vuosi ja lukukausitasolla, että arkistohakemisto on olemassa. Hakemiston puuttuessa ja hakemiston ollessa Moodlen datahakemiston alla, luodaan arkistohakemisto.

4.7.8 Metodi create_archive_dir(\$tunnus)

Metodi suorittaa varsinaisen arkistohakemiston luonnin ja kirjoittaa hakemiston nimen Moodlen parametritietokantaan.

4.8 Luokkatiedosto archiverestore.php

Luokassa suoritetaan arkistosta palautus prosessi. On johdettu luokasta archivecourse.

4.8.1 Muodostin __construct(\$id = null)

Kutsutaan perusluokan muodostinta, luoden ilmentymän ilman tietoa. Jos arkistoidun kurssin tunnus on annettu, luetaan tiedot arkistotietokannasta.

4.8.2 Metodi `get_archivedata()`

Luo arkistoidusta kurssista näytettävän tietonäytön tiedot.

4.8.3 Metodi `set_restore_preferences($to_link = null)`

Valmistellaan aloitettavan palautus luomalla tarvittavat väliaikaiset hakemistot, puhdistetaan mahdolliset vanhat tiedot pois, asetetaan ympäristömuuttujat ja ohjataan toiminta eteenpäin.

4.8.4 Metodi `update_restore_preferences($userdata = 1)`

Luodaan palautuksen luettelotiedot kurssille palautettavista aktiviteeteistä ja kurssin muista tiedostoista. Parametrilla `userdata` määritellään ollaanko luomassa tarkkaa kopiaita vai uutta kurssipohjaa. Tiedot talletetaan istuntomuuttujaan.

4.8.5 Metodi `update_restore_preferences_new($newcoursedata)`

Metodissa tehdään kurssipohjaksi palautuksen vaatimat muutokset luettelotietoihin.

4.8.6 Metodi `execute_restore()`

Suoritetaan varsinainen palautus prosessi. Tarkastetaan, että tarvittavat tiedot on annettu ja tämän jälkeen käynnistetään palautus, joka tehdään Moodlen palautus rutiinilla. Palautuksen jälkeen suoritetaan tarpeettomien tietojen puhdistus.

4.8.7 Metodi `parse_param($var)`

Tarkistetaan onko annettu parametri asetettu, tyhjä vai onko arvo `true` vai jotain muuta. Palautetaan arvon 1 kun tuotu arvo on `true`, muuten palautetaan 0.

4.8.8 Metodi `set_php_memory_settings()`

Tarkastetaan ja asetetaan ympäristömuuttujat väliaikaisesti toimintoon paremmin sopiviksi.

4.8.9 Metodi cleanup()

Poistaa käytetyt väliaikaiset tiedostot ja hakemistot sekä tyhjentää istuntopuuttujan.

4.9 Luokkatiedosto archivesearch.php

Luokan metodeilla suoritetaan hakuja arkistotietokantaan.

4.9.1 Muodostin __construct(\$sdata)

Muodostimessa välitetään hakuehdot ilmentymään.

4.9.2 Staattinen metodi get_archived_count()

Palauttaa arkistotietokannassa olevien arkistoitujen kurssien lukumäärän.

4.9.3 Metodi find()

Metodissa suoritetaan varsinainen haku arkistotietokantaan.

4.9.4 Metodi get_result_table(\$url)

Metodi muodostaa taulukkolistauksen haun tuloksista.

4.9.5 Metodi get_nextprev_table()

Muodostaa taulukkolistauksen haun tuloksista lisäten sivutuksen, kun haku tuloksia on enemmän kuin on määritelty.

4.9.6 Metodi set_limits(\$firstrow)

Metodi määrittelee haun ensimmäisen rivin tulosjoukossa.

4.9.7 Metodi is_multi_page()

Metodi palauttaa true, jos tuloksia enemmän kuin yhdelle sivulle on määritelty tai ensimmäinen rivi ei ole tulosjoukon ensimmäinen.

4.9.8 Metodi get_cntresult()

Saantimetodi haun tulosten lukumäärälle.

4.9.9 Metodi get_first_archive_id()

Metodi palauttaa hakutuloksen ensimmäisen tietueen arkistotunnuksen.

4.9.10 Metodi add_data(\$par, \$val)

Lisää asetetun hakusanan hakuehtoon.

4.9.11 Metodi make_search_sql()

Muodostaa hakuehdoista hakulauseen ja suorittaa haun arkistotietokantaan.

4.9.12 Metodi add_needed_and(&\$sql, \$param = "AND")

Lisää hakuehtojen väliin tarvittaessa "AND"-sanan.

4.9.13 Metodi parse_name(\$name)

Lisää hakuun käyttäjien ehdot oikeassa muodossa, sen mukaan onko valittu nimi vai käyttäjätunnushaku.

4.10 Luokkatiedosto archiveform.php

Luokka on kurssien arkistoinnin lomakkeiden perusluokka. Luokasta on mahdollista johtaa myös lomake suoraan, käytetty raporttityyppisissä lomakkeissa, joissa sisältöä luodaan toiminnan edetessä. Luo lomakkeille yhtenäisen muodon ja tarkastaa oikeudet tarvittaessa.

4.10.1 Muodostin __construct(\$restrict = true)

Muodostimessa tarkastetaan, että käyttäjä on kirjautunut sisään Moodleen ja haluttaessa tarkastaa oikeustason riittävyyden.

4.10.2 Metodi definition()

Metodi täytyy olla vaikka ei käytetä.

4.10.3 Metodi show_with_navigation(\$title, \$course = null, \$caption = null, \$title_extra = "")

Luo yhdessä lomaketiedoston kanssa sivu kokonaisuuden, jossa ovat myös yhteiset osat: ylätunniste, leivänmurujono ja alatunniste.

4.10.4 Metodi start_page_with_navigation(\$title, \$course = null,

Metodi lisää ylätunnisteen ja leivänmurujono sivulle.

4.10.5 Metodi end_page(\$course = null)

Metodi lisää alatunnisteen sivulle.

4.10.6 Metodi start_nav_box_page(\$title, \$course = null,

Metodilla aloitetaan raporttityylinen sivu. Lisää ensin ylätunnisteen ja leivänmurujono sivulle ja aloittaa tämän jälkeen laatikon johon sisältö tulostetaan.

4.10.7 Metodi end_box_page(\$course = null)

Lopettaa raporttityylinen sivun sulkemalla laatikon ja lisäämällä alatunnisteen.

4.10.8 Metodi check_rights()

Tarkastetaan, että käyttäjällä on riittävät oikeudet toimintojen suoritukseen.

5 TIETOKANTAHAKEMISTO ARCHIVE/DB

5.1 Kuvaustiedosto install.xml

Tiedosto sisältää XML-muotoisen esityksen lohkon tietokannasta, jonka perusteella asennusohjelma luo lohkolle tietokannan.

5.2 Oikeustiedosto access.php

Tiedossa määritellään arkistointilohkon tarvittavan oikeus määreen vaatimukset.

5.3 Kuvaustiedosto upgrade.php

Valinnainen tiedosto, jossa on tietokannan päivitys ohjeet.

6 KIELIHAKEMISTO ARCHIVE/LANG

6.1 fi_utf8/block_archive.php

Suomen kielen tekstit, lohkon alkuperäinen kieli.

6.2 en_utf8/block_archive.php

Lohkon tekstit englanniksi käännettynä.