

POHJOIS-KARJALAN AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutusohjelma

Ville Eronen

OPINTOJEN KUORMITTAVUUTTA SEURAAVAN VERKKOSO-
VELLUKSEN KEHITTÄMINEN

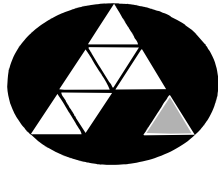
Opinnäytetyö
Joulukuu 2012

Sisältö

Sanasto	5
1 Johdanto.....	7
2 Teoria	8
2.1 Yleistä verkkosovelluskehyksistä	8
2.2 Muut tekniikat	13
3 Verkkosovelluskehysten vertailu	15
3.1 Yii Framework.....	15
3.2 Zend Framework.....	20
3.3 CodeIgniter	23
3.4 CakePHP	27
3.5 Symfony	31
4 Toteutus	36
4.1 Vaatimukset.....	37
4.2 Tietokanta.....	38
4.3 Käyttäjähallinta	39
4.4 Tietoturva	40
5 Tulokset.....	41
5.1 Kurssit	41
5.2 HOPS	43
5.3 Opettajan näkymä.....	46
6 Testaus ja jatkokehitystarpeet	48
7 Loppusanat.....	49
Lähteet	50

Liitteet

Liite 1	Verkkosovelluksen käyttötapauskaavio ja kirjallinen kuvaus
Liite 2	Tietokantakuvaus
Liite 3	Sekvenssikaavioita verkkosovelluksen toiminnoista



POHJOIS-KARJALAN
AMMATTIKORKEAKOULU

OPINNÄYTETYÖ
Joulukuu 2012
Tietojenkäsittelyn koulutusohjelma

Länsikatu 15
80100 JOENSUU
p. 050 311 6310

Tekijä
Ville Eronen

Nimeke
Opintojen kuormittavuutta seuraavan verkkosovelluksen kehittäminen

Toimeksiantaja
PKAMK Tietojenkäsittelyn koulutusohjelma

Tiivistelmä

Opinnäytetyön aiheena oli toteuttaa opiskelijoiden kuormituksen seurantaverkkosovellus PKAMK:n tietojenkäsittelyn koulutusohjelmalle. Verkkosovelluksen tarkoituksena oli helpottaa opiskelijoiden henkilökohtaisten opintosuunnitelmien suunnittelua ja luomista sekä auttaa ajanhallintaa havainnollistamalla opiskeluun käytettävä aika kaavioina.

Verkkosovellus toteutettiin Yii Framework -verkkosovelluskehityksen avulla ja ohjelmointiin käytettiin PHP-ohjelmointikieltä, HTML-kuvauskieltä sekä JavaScript-komentosarjakieltä. Verkkosovelluksen käyttöliittymässä käytettiin jQuery-JavaScript -kirjastoa ja kuormituksen seurantaan käytetyt kaaviot toteutettiin RGraph-JavaScript -kirjaston avulla.

Opinnäytetyössä tehtiin myös vertailu viiden eri verkkosovelluskehityksen välillä. Vertailussa tarkasteltiin verkkosovelluskehitysten asennusprosesseja, tietokantatukea ja sen käytettävyyttä sekä verkkosovelluskehitysten rakennetta.

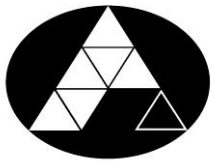
Työn tuloksena syntyi jatkokehitykseen kelpaava verkkosovellus, josta voidaan kehittää valmis sovellus. Raporttiin kirjattiin verkkosovelluksen toteutuksen eri vaiheita ja syitä sille, miksi tiettyihin ratkaisuihin päädyttiin. Jatkokehitystä varten tuotettiin tietokantakuvaus, käyttötapauskaavio ja sekvenssikaavioita eri toiminnoista.

Kieli
suomi

Sivuja 53
Liitteet 3
Liitesivumäärä 11

Asiasanat

verkkosovellus, verkkosovelluskehitys, verkkosovelluskehitys, opintosuunnitelma



NORTH KARELIA
UNIVERSITY OF APPLIED SCIENCES

THESIS
December 2012
Degree Programme in Business Information Technology
Länsikatu 15
FIN 80100 JOENSUU
FINLAND
Tel. 358-50 311 6310

Author
Ville Eronen

Title
Development of a web application for tracking workload of studies

Commissioned by
NKUAS Degree Programme in Business Information Technology

Abstract

The subject of the thesis was to create a student workload tracking web application for Degree Programme in Business Information Technology of NKUAS. The purpose of the web application was to make designing and creating study plans easier and help time management by visualizing study time by graphs.

The web application was created with Yii Framework web application framework and for programming PHP, HTML and JavaScript were used. JQuery JavaScript library was used for the user interface of the web application and RGraph JavaScript library was used for the workload tracking graphs.

A comparison between five different web application frameworks was also carried out. Installation processes, database support and the usability as well as structures of the frameworks were examined and compared.

As the result of the thesis a web application fit for further development was created from which a complete application can be developed. Different stages and reasons for why certain decisions were made were recorded in the report. For further development database description, use case diagram and sequence diagrams of certain features were produced.

Language
Finnish

Pages 53
Appendices 3
Pages of Appendices 11

Keywords

web application, web application development, web application framework, study plan

Sanasto

AJAX	Asynchronous JavaScript and XML eli ryhmä toisiinsa liittyviä verkkokehitystekniikoita vuorovaikutteisten verkkosovellusten toteuttamiseen.
BSD-lisenssit	Vapaiden ohjelmistolisenssien perhe. Alkuperäinen BSD-lisenssi on tunnettu Berkeley Software Distribution -käyttöjärjestelmän lisenssinä.
CRUD	Create, Retrieve, Update, Delete ovat tietokannan neljä perusoperaatiota, eli tiedon lisääminen, hakeminen, päivitys ja poisto.
CSS	Cascading Style Sheets on tyyliohje merkintäkielellä kirjoitetuille dokumenteille. Tuettuja kieliä ovat HTML, XHTML, XML, SVG ja XUL.
DAO	Data access object on objekti, joka tarjoaa rajapinnan tietokantaan.
DRY	Don't repeat yourself, eli "Älä toista itseäsi" on ohjelmistokehityksen periaate, jossa pyritään vähentämään kaikenlaista toistamista esimerkiksi ohjelmakoodissa tai dokumentaatiossa.
FTP	File Transfer Protocol on verkkoprotokolla, jota käytetään tiedon siirtoon asiakaslaitteen ja palvelimen välillä.
GPL	GNU General Public License on Free Software Foundationin ylläpitämä vapaa ohjelmistolisenssi.
HOPS	Henkilökohtainen opetussuunnitelma.
HTML	HyperText Markup Language on merkintäkieli verkkosivujen esittämiseen.
HTTP	Hypertext Transfer Protocol on hypertekstin siirtoprotokolla.
MIT-lisenssi	Massachusetts Institute of Technologyn julkaisema vapaa ohjelmistolisenssi.
MVC	Model-view-controller eli malli-näkymä-käsittelijä on arkkitehtuuri, joka erottaa käyttöliittymän ohjelmakoodista.

ORM	Object-relational mapping on ohjelmointitekniikka jolla muutetaan tietoa epäyhteensopivien tyyppijärjestelmien välillä jollakin olio-orientoituneella ohjelmointikielellä.
OSL	Open Software License on Lawrence Rosenin luoma vapaa ohjelmistolisenssi.
PDO	PHP Data Objects on PHP-rajapinta tietokantojen käsittelyyn.
PHP	PHP: Hypertext Preprocessor on palvelinpuolen ohjelmointikieli, jolla tehdään dynaamisia verkkosivuja.
RAD	Rapid application development eli nopean kehityksen malli on ohjelmistokehityksen malli, joka auttaa toteuttamaan ohjelmistoprojekteja hyvin nopealla aikataululla, ilman laaja-alaista suunnittelua ja laadusta tinkimällä.
SQL	Structured Query Language on kyselykieli relaatiotietokannan käsittelyyn.
YAML	YAML Ain't Markup Language on merkintäkieli.
YUM	Yii-user-management on kolmannen osapuolen kehittämä Yii Framework -käyttäjänhallintalisäosa.
XML	Extensible Markup Language on merkintäkieli, jota käytetään tietorakenteiden kuvaamiseen ja tiedon tallentamiseen.

1 Johdanto

Opinnäytetyöni käsittelee opiskelijoiden kuormituksen seurantaverkkosovelluksen kehittämistä. Käytännössä sovellus toimii siten, että opiskelija syöttää sovellukseen henkilökohtaisen opetussuunnitelmansa ja sovellus laskee kuormitavuuden kullekin vuodelle. Sovellus palvelee mahdollisesti jatkokehityksen jälkeen tietojenkäsittelyn koulutusohjelman opiskelijoita opintojen suunnittelussa ja kursseille ilmoittautumisessa. Opinnäytetyön tavoitteena on kuvata edellä mainitun verkkosovelluksen kehitystä ja tehdä samalla vertailuja eri verkkosovelluskehityksien välillä.

Opinnäytetyön toimeksiantajana on Pohjois-Karjalan ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma. Valitsin tämän aiheen, koska olen opinnoissani keskittynyt verkkosovelluskehitykseen ja tässä työssä käyttämäni tekniikat ovat uusia, joita minulla ei ole ollut mahdollista opiskella aiemmin. Näen tämän opinnäytetyön olleen oivallinen tilaisuus opiskella näitä tekniikoita ja soveltaa oppimani asiat käytäntöön.

Sovellus on lisensoitu ”Creative Commonsin Nimeä 3.0 Muokkaamaton” -lisenssillä, joka antaa vapauden jakaa ja muokata sovellusta sillä ehdolla, että teoksen alkuperäisen tekijän nimi mainitaan. Tällä tavoin pyrin mahdollistamaan sovelluksen vapaan jatkokehittämisen tulevaisuudessa. Luovutan myös sovelluksen lähdekoodin koulutusohjelman käyttöön, jotta sitä voitaisiin jatkokehittää tulevilla projektikursseilla tai muissa opinnäytetöissä.

Valitsin verkkosovelluksen toteuttamiseen tekniikoiksi Hypertext Markup Language:n 5:n version (HTML5), PHP: Hypertext Preprocessorin (PHP), jQuery -JavaScript-kirjaston ja MySQL-relaatiotietokantaohjelmiston. PHP:llä toteutin sovelluksen logiikkapuolen ja tietokantayhteydet. HTML5:llä toteutin sivuston perusrakenteet ja kuormituksen mittaamista ilmentäviin kaavioihin oli käytetty RGraph-työkalua, joka vaatii toimiakseen HTML5:n Canvas-elementtiä. JQuery:llä on luotu sivuston navigointivalikot ja muita käyttöliittymän osia.

Toisessa luvussa kerrotaan verkkosovelluskehyksistä sekä sovelluksessa käytetyistä tekniikoista ja vertailen viittä eri PHP-verkkosovelluskehystä. Kolmannessa luvussa kerrotaan verkkosovelluksen suunnittelusta, toteutuksesta, dokumentoinnista sekä tuloksista. Raportoin myös verkkosovelluksen testauksesta ja kuinka se toimii sellaisenaan mobiililaitteilla, jotta sovelluksen jatkokehittäminen myös mobiiliyhteensopivaksi helpottuisi.

Työtäni motivoi myös se, että nykyiset kouluni käyttämät ratkaisut kursseille ilmoittautumisiin ja henkilökohtaisen opintosuunnitelman suunnitteluun ja luontiin ovat erillisiä ja hajallaan olevia palveluita (SoleOPS, Winha, VirtuaaliAMK), joita on jopa vaivalloista käyttää yhdessä. Toteutettavan verkkosovelluksen on tarkoitus tuoda nämä palvelut yhden sivuston alle ja tarjota ensimmäistä kertaa mahdollisuus luoda HOPS verkossa koulutusohjelmassani. Aiemmin HOPS on luotu erillisenä Microsoft Excel -tiedostona ja se on toimitettu opettajatutorille muistitikulla tai Moodle-työtilan kautta.

2 Teoria

2.1 Yleistä verkkosovelluskehyksistä

Verkkosovelluskehys (web application framework) on ohjelmistokehys (software framework) dynaamisten verkkosivustojen ja -palveluiden toteuttamista varten. Niiden tavoitteena on helpottaa kehittäjän työtä verkkosovellusta kehitettäessä ja nopeuttaa kehitysprosessia. Tavallisesti verkkosovelluskehys sisältää toiminnot käyttäjän istunnonhallintaan, tiedon pysyvyyden varmistukseen ja muuttijärjestelmään. (DocForge 2011.)

Verkkosovelluskehysten ydintoiminto on se, että staattisia sivuja ei tarvita juuri lainkaan, vaan verkkosovelluksen sivut generoidaan dynaamisesti pysyvästä tiedosta. Tämän ominaisuuden avulla esimerkiksi jokaista blogin sivua ei tarvitse erikseen ohjelmoida, vaan käytetään samaa muuttia kaikkiin blogiteksteihin. Näin sivuston ulkoasu pysyy yhtenäisenä, aikaa säästyy ja HTML:ää

osaamatonkin voi luoda uusia sivuja ilman suoraa tiedostojen muokkaamista. (DocForge 2011.)

Muita yleisiä ominaisuuksia ovat muun muassa istunnonhallinta ja käyttäjän autentikointi, eri turvallisuusominaisuudet, tiettyjen sivujen välimuistiin kirjoittaminen ja pääkäyttäjän hallintapaneeli. Istunnonhallintaan sisältyy toiminnot käyttäjän rekisteröintiin ja sisäänkirjautumiseen sekä joissakin tapauksissa valmiita geneerisiä pohjia käyttäjätunnuksille, joita voidaan laajentaa tarpeen vaatiessa. Turvallisuusominaisuuksia voivat olla käyttäjän syötteiden suodattaminen verkkosovelluksen turvallisuudelle haitallisista syötteistä, roolipohjainen käyttäjien hallinta ja arkaluontoisen tiedon salaaminen. Tiettyjen sivujen tallentaminen välimuistiin poistaa tarpeen sivun dynaamiseen generoimiseen ja siten parantaa palvelimen suorituskykyä. Hallintapaneelista pääkäyttäjät pääsevät yleensä luomaan käyttäjätunnuksia, muokkaamaan sivujen oikeuksia ja sisältöä sekä generoimaan navigointirakennetta. Joissakin verkkosovelluskehysissä hallintapaneelin kautta voi lisätä sivustolle käyttöliittymäelementtejä, kuten lomakkeita tai vaikkapa kalenterin. (DocForge 2010.)

Yleensä verkkosovelluskehys pohjautuu Model-View-Controller -arkkitehtuuriin (malli-näkymä-käsittelijä), joka erottaa käyttöliittymän koodista. MVC-arkkitehtuurissa malli hoitaa ainoastaan kaiken sovelluksen tiedon tallentamisen, ylläpidon ja käsittelyn. Näkymä edustaa sovelluksen esittämistä ja ulkoasua. Se viittaa malliin, hakee esitettävän tiedon mallin kyselymetodien avulla ja esittää lopulta haetun tiedon loppukäyttäjälle. Käsittelijä yhdistää näkymän ja mallin ”sieppaamalla” käyttäjän pyynnöt näkymästä ja lähettämällä ne eteenpäin mallille. Kun käyttäjän tekemä toiminto on käsitelty, käsittelijä lähettää oikean näkymän takaisin käyttäjälle. (Rose India 2008; DocForge 2011.)

Yii Framework

Yii (Yes it is) Framework on avoimeen lähdekoodiin perustuva, selkeään ohjelmointiin ja DRY-suunnitteluun rohkaiseva verkkosovelluskehys. Yii on ohjelmoitu PHP5-ohjelmointikielellä ja sen ovat toteuttaneet PRADO-sovelluskehysen kehitystiimistä lähteneet Qian Xue ja Wei Zhue. (Yii Software LLC 2012a.)

Yii käyttää DAO-suunnittelumallia. DAO:n avulla eri tietokantajärjestelmiin yhdistäminen onnistuu yhden ja saman rajapinnan kautta ilman tietoa käsittelevien tiedostojen muuttamista. Malligeneraattorilla voidaan luoda helposti ja nopeasti tieto-objekteja tietokannan tauluista tai verkkosivun lomakkeesta. Tämä säästää ohjelmointityössä todella paljon aikaa, etenkin jos käsiteltävänä on laajoja tietokokonaisuuksia. Muita Yiiin ominaisuuksia ovat tietokantakyselyn rakentaja, lomakkeen syöttö ja validointi, AJAX-lisäosat, käyttäjän autentikointi ja roolipohjainen käyttöoikeuksien hallinta. (Yii Software LLC. 2012b; 2012c.)

Zend Framework

Zend Framework on Yiiin tavoin MVC-pohjainen sovelluskehys, joka on ilmainen ja BSD-lisensoitu. Zend on myös toteutettu PHP5:llä ja sen ominaisuuksia ovat AJAX-tuki, Lucene-hakukonemoottori, syndikointi eli sisällön jakaminen eteenpäin esimerkiksi RSS-syötteiden avulla, www-sovelluspalveluiden (Web Services) luonti sekä olio-orientoitunut PHP5-luokkakirjasto. (Zend Technologies Ltd. 2012i; 2012j.)

Zend Frameworkista on ladattavissa kolme eri versiota: Zend Framework + Zend Server Community Edition, full package ja minimal package. Zend Server CE on sovelluspalvelin, joka integroituu Apache- tai IIS-palvelimeen ja se tarjoaa käyttäjälleen Java-liittimen, debuggauskäyttöliittymän, Bytecode accelerationin ja ohjelmointirajapinnan tietojen tallentamiseen välimuistiin. Java-liittimen avulla ohjelmoija voi kutsua PHP:llä Java-koodia Java EE-sovelluspalvelimelta. Debuggauskäyttöliittymä helpottaa ohjelmoijan työtä virheiden paikantamisessa tallentamalla reaaliaikaisesti sovelluksen toimia, jolloin mahdolliset virheet ja koodin suorittamista hidastavat tekijät saadaan selville. Bytecode acceleration ja välimuistiin tallentaminen tehostavat sovelluksen käyttöä, koska kovalevyiltä lukemisen tarve vähentyy. Tarjolla on myös maksullinen Zend Server, joka tarjoaa Community Edition verrattuna enemmän ominaisuuksia ja tuotetuen. (Zend Technologies Ltd 2012k; 2012l; 2012m; 2012n.)

CodeIgniter

CodeIgniter on yhdysvaltalaisen ohjelmistotalo EllisLabin verkkosovelluskehys. CodeIgniter on ilmainen ja avoimeen lähdekoodiin pohjautuva sovellus ja se on lisensoitu BSD-tyylisellä lisenssillä. Uudempi versio, 3.0, joka on tällä hetkellä ennakkoversio, on lisensoitu OSL-lisenssillä. CodeIgniter pohjautuu myös MVC-arkkitehtuuriin, vaikkakin löysästi: tietoa tallentavia, ylläpitäviä ja käsitteleviä malleja ei ole pakko käyttää. CodeIgniterissa on myös sivupohjamoottori (template engine), jonka avulla saadaan sivun koodista hieman helpommin luettavampaa tehokkuuden kustannuksella. EllisLab suosittelee, että sivupohjamoottoria ei käytettäisi sivujen esittämiseen, vaikka pseudokoodia olisi PHP:tä helpompaa kirjoittaa. (EllisLab Inc 2012b; 2012d; 2012f; 2012g.)

CodeIgniter tarjoaa perusomaisuuksien kuten käyttäjän istunnonhallinnan, lomakkeiden ja tiedon tarkistuksen lisäksi monia luokkia: muun muassa kalenterin, FTP-luokan, jonka avulla voi ladata tiedostoja ulkoiselle FTP-palvelimelle ja Zip-enkooderin, jonka avulla voidaan pakata tiedostoja .zip-paketeiksi (EllisLab Inc 2012c.)

CakePHP

CakePHP on Cake Software Foundationin kehittämä ilmainen, avoimen lähdekoodin sovelluskehys, joka noudattaa nopean kehityksen mallia (RAD). (Cake Software Foundation Inc 2012i). RAD auttaa toteuttamaan ohjelmistoprojekteja hyvin nopealla aikataululla, ilman laaja-alaista suunnittelua ja laadusta tinkimällä. (Custom Application Development Portal 2010). CakePHP on lisensoitu vapaalla ohjelmistolisenssillä MIT-lisenssillä, joka antaa käyttää sovellusta kaupallisiin tarkoituksiin. (Open Source Initiative 2009, Cake Software Foundation Inc 2012b).

Kuten muutkin sovelluskehukset CakePHP perustuu myös MVC-arkkitehtuuriin. CakePHP:ssä on sivupohjamoottori, joka toimii PHP-syntaksilla ja sisältää avustajia (helpers), joilla voi esimerkiksi toteuttaa lomakkeita, luoda oikein muodostettuja kuvia, linkkejä, taulukkoja ja header-tunnisteita HTML:lla. Avustajia

on myös JavaScript -koodin sisällyttämiseen, AJAXille, sivujen taittoon ja järjestämiseen sekä istuntoihin. CakePHP:n on integroituna tietokannan CRUD-operaatiot, joten sovelluskehittäjän ei tarvitse niitä itse ohjelmoida. Tämä säästää aikaa sekä antaa mahdollisuuden keskittyä muihin asioihin. (Cake Software Foundation Inc 2012d.)

Symfony

Symfony on ranskalaisen Sensio Labsin ylläpitämä verkkosovelluskehys, joka on kirjoitettu PHP:llä ja lisensoitu MIT-lisenssillä, kuten CakePHP:kin. Symphonyyn on ladattavissa kokoelma useita eri kirjastoja, joita voi käyttää vapaasti missä tahansa PHP-projekteissa. Näitä komponenteiksi nimettyjä kirjastoja voi käyttää yhdessä ja erikseen, vaikkei koko Symphonya olisi edes asennettuna. Symfonyn ydinkomponentteja ovat istunnon ja tiedoston latauksen hallintakomponentti, reitityskomponentti, jolla voidaan asettaa tietty verkko-osoite viemään tietylle projektin sivulle, lomakkeen luomis- ja käsittelykomponentti, syötettävän tiedon validointikomponentti, kaikenlaisen tietoturvan sisältävä komponentti ja käännöskomponentti. Lisäksi Symphonyssa on tuki sivupohjamoottorille. Symphonyyn on myös ladattavissa lukuisia kolmannen osapuolen komponentteja ja nämä toimivat samalla periaatteella kuin viralliset komponentitkin. Symphony itsessäänkin voidaan laskea komponentiksi ja se voidaan korvata jollain toisella komponentilla. (Sensio Labs 2012c; 2012e; 2012g.)

Symfony pyrkii nopeuttamaan ohjelmointityötä vähentämällä toistuvien ohjelmointitehtävien määrää uudelleenkäyttämällä geneerisiä moduuleita. Sensio Labs lupaa myös, että Symfonyn käyttö parantaa sovelluksen laatua, päivitettävyyttä ja ylläpidettävyyttä sekä alentaa yrityksen kuluja. (Sensio Labs 2012j.)

2.2 Muut tekniikat

JQuery

JQuery on voittoa tavoittelemattoman jQuery Foundationin kehittämä JavaScript-kirjasto, jonka tarkoitus on yksinkertaistaa komentosarjojen kirjoittamista HTML-dokumentteihin. JQueryyllä voidaan luoda animaatioita ja hoitaa tapahtumankäsittelyä sekä AJAX-interaktioita. JQuery on avoimen lähdekoodin projekti ja se on lisensoitu MIT-lisenssillä, mutta myös GPL-lisenssin käyttö on luvalista. JQuerya voidaan käyttää myös kaupallisissa projekteissa, kunhan tekijänoikeustiedot jätetään näkyviin. (jQuery Foundation 2012a; 2012d; 2012g.)

JQuerystä on ladattavissa kahta eri versiota: tuotantoversio ja kehitysversio. Tuotantoversiossa jQuery:n koodi on minimoitu, jotta se veisi vähemmän tilaa ja tämä johtaa pienempään tiedoston kokoon. Kehitysversiossa koodia ei ole pakattu ja koodi on helpommin luettavassa muodossa, minkä takia sitä on yksinkertaisempaa debugata ja tehdä muutoksia lähdekoodiin. JQuerya on mahdollista käyttää myös ulkopuoliselta palvelimelta, jos ei halua tai ei ole mahdollista ladata jQuerya palvelimelleen. Esimerkiksi Googlen ja Microsoftin palvelimilta löytyvät kopiot jQuerysta, johon voidaan vapaasti viitata omassa HTML-koodissa. (jQuery Foundation 2012b.)

Käytännössä jQuery toimii siten, että sen tiedostoon viitataan script-tunnisteen sisällä HTML:ssä, minkä jälkeen tehdään heti uusi script-tunniste, johon jQuery-koodia sijoitetaan. Tyypillisesti jQuery-koodi aloitetaan `$(document).ready()`-funktioilla, joka ottaa jQuery-koodin käyttöön heti, kun HTML-dokumentti on ladattu. (jQuery Foundation 2012h.)

JQueryyn on ladattavissa myös käyttöliittymäkirjasto jQuery UI, joka tarjoaa verkkosovelluskehittäjälle abstraktioita matalan tason vuorovaikutukseen sekä animaatioita, efektejä ja lisäosia, joille voi asettaa teemoja. Tarjolla olevia interaktioita ovat muun muassa objektin raahaaminen ja pudottaminen, objektin

koon muuttaminen, objektin valitseminen ja objektien järjestely. Virallisia jQuery UI:n lisäosia ovat haitarivalikko, ennakoiva tekstinsyöttö, painike, päivämäärävalikko, dialogi, edistymispalkki, liukusäädin ja välilehtikäyttöliittymä. JQuery UI:n efektit sisältävät lukuisia erilaisia animaatioita, joilla voi elävöittää verkkosovellusta. JQuery UI:n lataus on räätälöitävissä: käyttäjä voi valita haluamansa vuorovaikutukset, lisäosat ja efektit ladattavaan pakettiin. Lisäksi käyttäjä voi luoda haluamansa teeman jQuery UI:n lisäosille ThemeRoller-työkalun avulla tai valita galleriasta valmiin teeman. (jQuery Foundation 2012e; 2012f.)

RGraph

RGraph on JavaScript-kirjasto, jonka avulla voidaan muodostaa yli 20 erilaista kaaviotyyppeä käyttämällä HTML5:n canvas-elementtiä. Canvas-elementin avulla voidaan renderöidä mitä tahansa grafiikkaa lennosta. Koska kaaviot luodaan suoraan selaimessa eikä palvelimella, säästetään paljon palvelimen kuormituksessa ja nopeutetaan verkkosivujen toimintaa. RGraphin käyttö on ilmaista epäkaupallisessa käytössä eli henkilökohtaisessa, hyväntekeväisyys- tai koulutuskäytössä. Jos RGraphin haluaa kaupalliseen tai hallinnolliseen käyttöön, sen lisenssi maksaa yhdelle verkkotunnukselle 99 puntaa. (RGraph Licensing 2012a; 2012e; 2012d; World Wide Web Consortium 2012.)

RGraphilla voidaan luoda pylväs-, palkki-, viiva-, ympyrä- sekä rengaskaavioita. Lisäksi pystytään tekemään suunnittelu- ja hallintakaavioita, mittareita ja muita kaaviotyyppejä kuten LED-ruudukkoja ja säteittäisiä kaavioita. (RGraph Licensing 2012c.). RGraph otetaan käyttöön verkkosivulla lisäämällä haluttu RGraph-kirjasto HTML-sivun head-elementin sisään script-elementillä. Jokaiselle kaaviotyypille on oma kirjastonsa, joten kehittäjä saa valita mitä kaavioita haluaa ladata sivuilleen. Body-elementin sisällä canvas-elementti sijoitetaan sinne, missä kaavion halutaan sivulla näkyvän. Canvas-elementti vaatii attributeikseen tunnusteen sekä kaavion leveyden ja korkeuden. Kaavion tyyppi, otsikot, tiedot ja kaavion piirtämisen laukaisija (esimerkiksi kun dokumentti on ladattu) määritellään script-elementin sisällä JavaScript-koodina, ja tämän elementin voi sijoittaa minne tahansa HTML-dokumentissa. (RGraph Licensing 2012b.)

3 Verkkosovelluskehysten vertailu

Valitsin vertailuun viisi suosituinta PHP-verkkosovelluskehystä PHP Frameworks -sivuston Top 10 -listan perusteella. Sivuston käyttäjät ovat voineet äänestää suosikkejaan ja viisi suosituinta verkkosovelluskehystä ovat Yii Framework, CodeIgniter, CakePHP, Zend Framework ja Symfony. (PHP Frameworks 2012.)

Lähdin vertailemaan eri verkkosovelluskehysten ominaisuuksia, rakennetta ja syntaksia. Koska aika on rajallinen, en kovin syvällisesti voinut opetella jokaista verkkosovelluskehystä, vaan tyydyin tutustumaan jokaisen ominaisuuksiin ja dokumentointiin kunkin verkkosovelluskehysten kotisivujen kautta. Syntaksiin tutustuin käymällä läpi jokaisen verkkosovelluskehysten verkkosivuilta löytyvän oppaan ja toistamalla niiden sisällön palvelimelläni.

3.1 Yii Framework

Asennus

Yii Frameworkin asentaminen on yksinkertaista: käyttäjän tulee ladata Yii Frameworkin lähdekoodipaketti Yii Frameworkin virallisilta sivuilta ja purkaa se palvelimen juureen, minun tapauksessani kohteeseen `/var/www/yii`. Palvelimen ja Yii Frameworkin yhteensopivuus kannattaa varmistaa menemällä verkkoselaimella osoitteeseen `http://localhost/yii/requirements/index.php`. Sivun kertoo, vastaavatko palvelimelle asennetut ohjelmat ja liitännäiset Yii Frameworkin vaatimuksia. Kun näin on, voidaan luoda uusi projekti käyttämällä `yii`-komentorivityökalua. Ohjelma suoritetaan komentoriviltä antamalla ensin projektin nimi ja sen haluttu sijainti, esimerkiksi minun tapauksessani: `/var/www/yii/framework/yii projektin_nimi /var/www/yii/projekti`. Yii Framework ja juuri luotu projekti ovat nyt käytettävissä, ja tätä demonstroimassa on Yii Frameworkin esimerkiksi sivusto, joka esittelee joitakin Yii Frameworkin ominaisuuksia, kuten kirjautumislomaketta ja yhteydenottolomaketta. (Yii Software LLC 2012c; 2012d.)

Tietokanta

Jos käyttäjä haluaa liittää tietokannan projektiinsa, täytyy hänen muokata Yii Frameworkin asetustiedostoa, joka löytyy minun tapauksessani kohteesta: `/var/www/yiiprojekti/protected/config/main.php`. Kyseisestä tiedostosta käyttäjän pitää etsiä `connectionString` -muuttuja ja asettaa siihen käyttämänsä tietokantamooottori ja tietokannan sijainti, tai jos käyttäjä käyttää MySQL:ää, voi hän jättää muuttujan huomioimatta ja ottaa kommentoinnin pois muuttujan alapuolella olevasta `db`-taulukosta. `ConnectionString` -muuttujaan tulee käytettävän tietokannan osoite ja nimi. `Username` ja `password` -muuttujille annetaan samat tiedot kuin kirjautuessa sisään MySQL-palvelimelle, ja `charset` -muuttujaan syötetään tietokannan käyttämä merkistökoodaus. (Yii Software LLC 2012c.)

CRUD-operaatiot otetaan käyttöön gii-työkalun avulla. Ennen giin käyttöä, käyttäjän tulee asettaa edellä mainitussa `main.php`-tiedostossa giin salasana. Giitä käytetään navigoimalla verkkoselaimella osoitteeseen `http://localhost/yii-projekti/index.php?r=gii`. Sivun kysyy käyttäjän aiemmin asettamaa salasanaa ja oikean salasanan saatuaan ohjaa käyttäjän giin etusivulle. Tältä sivulta käyttäjä voi luoda nopeasti käsittelijöitä, CRUD-operaatioita, lomakkeita, malleja ja moduleja projektiin. Saadakseen CRUD-operaatiot käyttöön nykyisessä tietokannassa käyttäjän täytyy ensin luoda malli painamalla `Model Generator` -linkkiä ja antamalla `Table Name` -tekstikenttään halutun tietokannan taulun nimen ja `Model Class` -tekstikenttään nimen, jolla kyseiseen tauluun tullaan viittaamaan koodissa. Käyttäjän tulee vielä esikatsella generoitu malli painamalla `Preview`-nappia. Jos luotu mallitiedosto kelpaa, luodaan se painamalla `Generate`-nappia. Tämän jälkeen käyttäjän täytyy luoda CRUD-operaatiot painamalla `Crud Generator` -linkkiä ja antamalla äsken luodun mallin nimi `Model Class` -kenttään. `Controller ID` -kenttään annetaan haluttu nimi, jolla käsittelijää kutsutaan. Tällä samalla nimellä navigoidaan myös kyseisen mallin sivulle verkkoselaimella. CRUD-operaation luonti esikatsellaan ja hyväksytään kuten edellä mainitussa mallitiedoston luomisessa. Juuri luodut CRUD-sivut ovat nyt käytettävissä esimerkiksi osoitteessa `http://localhost/yiiprojekti/index.php?r=user`, jossa `r:n` jälkeinen merkkijono on käsittelijän nimi. Tältä sivulta käyttäjä voi tarkastella,

muokata ja poistaa kyseiseen tietokantatauluun tallennettuja tietueita tai luoda uusia. (Yii Software LLC 2012c.)

Mielestäni CRUD-operaatioiden luominen giin avulla on hyödyllistä ja aikaa säästävää vain, jos aikoo käyttää mallia yhden tietokantataulun käsittelyyn. Minun tapauksessani hyödynsin useaa taulua yhtä aikaa ja jouduin kirjoittamaan itse omat tietokantakyselyt mallitiedostoihin, joten automaattisesta CRUD-generoinnista ei minulle ollut juurikaan hyötyä.

Yii Frameworkissa voidaan muodostaa tietokantakyselyitä Query Builderin avulla olio-orientoituneesti ja tietoturvalisella tavalla, koska se automaattisesti asettaa lainausmerkit tietokantataulujen, rivien ja muuttujien nimien ympärille, mikä vähentää SQL-injektion riskiä. Tyypillinen Query Builderilla muodostettu kysely näyttää tältä:

```
$user = Yii::app()->db->createCommand()
    ->select('id, username, profile')
    ->from('tbl_user u')
    ->join('tbl_profile p', 'u.id=p.user_id')
    ->where('id=:id', array(':id'=>$id))
    ->queryRow();
```

Kyselyllä haetaan tbl_user ja tbl_profile -tauluista tunnus sekä käyttäjänimi ja profiili, joiden tunnus täsmää annetun muuttujan kanssa. Vastaava kysely näyttää PDO:lla toteutettuna tältä:

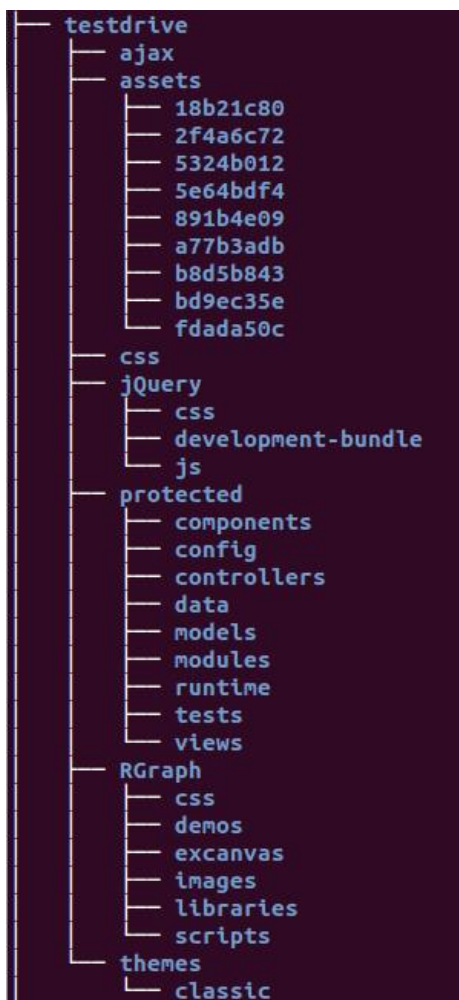
```
$query = $db->query("SELECT id, username, profile FROM
tbl_user AS u, tbl_profile AS P WHERE u.id = p.user_id AND
id = '$id'");
$user = $query->fetchAll(PDO::FETCH_ASSOC);
```

Minulle Query Builderin käyttö tuntui vaivalloiselta ja enemmän aikaa vievältä kuin PDO:n käyttäminen, joten päädyin käyttämään sovellukseni tietokantakyselyissä PDO:ta. Näen kuitenkin Query Builderille käyttöä laajempia tieto-

kantoja käsittelevissä sovelluksissa, joissa täytyy käsitellä enemmän kuin kolme taulua kerrallaan, sillä mitä enemmän tietokantatauluja ja kenttiä joutuu käsittelemään PDO:lla, sitä monimutkaisemmaksi koodi muuttuu. Tästä johtuen tietyn pisteen jälkeen Query Builder on helppolukuisempi kuin PDO, sillä Query Builderissä kyselyn osat ovat pilkottu omille riveilleen. (Yii Software LLC 2012e)

Rakenne

Ruutukaappauksessa (kuva 1) näkyvät Yii Framework -projektin automaattisesti luomat hakemistot. AJAX-, jQuery- ja RGraph-hakemistot liittyvät toteuttamaani sovellukseen, eivätkä ole osa Yii Frameworkia. Tärkeimmät kansiot Yii Framework -projektissa käyttäjälle ovat protected-kansiossa sijaitsevat controllers-, models- ja views-kansiot, jotka nimensä mukaisesti pitävät sisällään käsittelijä-, malli- ja näkymätiedostot. Giin CRUD-generaattorin luomat käsittelijätiedostot tallentuvat controllers-kansioon ja näkymätiedostot views-kansion Controller ID:n mukaan nimettyyn alikansioon, kun taas malligeneraattorin luoma mallitiedosto tallentuu models-kansioon. Sivuston ulkoasua pystyy muokkaamaan projektin juuressa sijaitsevan CSS-kansion CSS-tyylitiedostoilla ja views-kansion layouts-alikansiossa sijaitsevan main.php-tiedoston kautta. Kyseinen tiedosto toimii koko sivuston ulkoasuna, eikä vaikuta näkymätiedostoista tulevaan sisältöön millään tavalla.



Kuva 1. Yii Framework -projektin hakemistorakenne.

Käyttäjä voi lisätä uuden sivun projektiin luomalla PHP-tiedoston views-kansiossa haluttuun mallikansioon ja lisäämällä tiedoston nimen kyseisen mallin käsitelijätiedostoon ja accessRules-metodin haluttuun taulukkoon. Taulukkoja on neljä: ensimmäinen taulukko antaa kaikille oikeuden avata kyseessä olevan sivun, toinen taulukko antaa sisäänkirjautuneille oikeuden avata sivun, kolmas taulukko antaa pelkästään pääkäyttäjälle oikeuden avata sivun ja neljäs taulukko estää kaikilta pääsyn sivulle. Lopuksi käyttäjän pitää luoda samaan käsitelijätiedostoon julkinen metodi, jossa haluttu näkymätiedosto renderöidään render-funktiolla.

3.2 Zend Framework

Asennus

Zend Framework on asennettavissa Zend Frameworkin kotisivuilta löytyvästä asennuspaketista tai lataamalla Zend Frameworkin runkosovelluksen, joka käyttää Zend Framework 2 -MVC-kerrosta ja moduulijärjestelmiä. Tämä sovellus on tarkoitettu Zend Frameworkin aloitteleville käyttäjille ja koska Zend Frameworkin viralliset dokumentaatiot käyttävät tätä runkosovellusta esimerkeissään, käytän minäkin sitä vertailussani. (Zend Technologies Ltd 2012b; 2012e.)

Ladattu runkosovellus puretaan palvelimen juureen, mutta ennen varsinaista asennusta on kuitenkin varmistettava Apachen .htaccess-tiedostojen tuki tarkastamalla, että Apachen mod_rewrite-moduuli on asennettuna ja Apachen asetustiedostosta, httpd.confista, on AllowOverride -asetuksen arvoksi on annettu FileInfo. Htaccess-tiedostoilla korvataan hakemistokohtaisesti Apachen yleisiä asetuksia, esimerkkinä salasanasuojauksien asettaminen ja CGI-ohjelmien suorittamisen estäminen. (The Apache Foundation 2011.)

Asennus tapahtuu suorittamalla komentoriviltä komennot `php composer.phar self-update` ja `php composer.phar install`. Seuraavaksi tehdään virtuaali-isäntä luomalla projektin mukaan nimetty tiedosto hakemistoon `/etc/apache2/sites-enabled`. Juuri luotuun tiedostoon asetetaan palvelimen portti, haluttu palvelimen nimi, esimerkiksi `projekti.localhost`, projektin julkisen kansion sijainti, projektikansion etusivun tiedostonimi, `AllowOverride All`, jotta kaikki hakemistot voivat käyttää .htaccess-tiedostoja, `Order allow,deny`, joka määrittää että palvelin käsittelee ensin sallittujen ja sitten kiellettyjen hakemistojen säännöt sekä `Allow from all`, joka sallii yhteydenotot kaikilta asiakaskoneilta. Tämän jälkeen asetetaan `/etc/hosts`-tiedosto ohjaamaan äskeiseen tiedostoon määritetyllä palvelimen nimellä tehdyt yhteydenotot localhostin IP-osoitteeseen. Nyt Zend Framework on käytettävissä helposti `projekti.localhost`-osoitteessa. (Maxi-Pedia 2012; Zend Technologies Ltd 2012c; 2012d.)

Tietokanta

Zend Frameworkissa mallitiedostot luodaan projektin Model-hakemistoon ja ne vastaavat aina yhtä tietokantataulua. Ohjaavia objekteja käytetään tietokannasta lataamiseen ja tietokantaan tallentamiseen. Ohjaavan objektin luokkaan luodaan metodi, joka kopioi taulukkoon siirretyn datan objektin ominaisuuksiin. Tietokantaa käsittelevät metodit sijoitetaan eri mallitiedostoon, joka käyttää TableGateway-luokkaa, joka taas on vuorovaikutuksessa tietokantataulussa olevan tiedon kanssa. Mitään valmiita asetustiedostoja tai malleja ei luoda asennuksen yhteydessä, vaan kaikki tiedostot on luotava tyhjästä itse. (Zend Technologies Ltd 2012a.)

Jotta varmistetaan että samaa instanssia mallitiedoston luokasta käytetään aina, luodaan projektin juureen luotuun Module.php tiedostoon metodi, jossa käytetään ServiceManageria määrittämään kuinka kyseinen instanssi luodaan. Metodi palauttaa taulukon ”tehtaita”, jotka ModuleManager yhdistää ennen kuin ne lähetetään eteenpäin ServiceManagerille. Taulukkoon määritellään kaksi ”tehdasta”: ensimmäisessä välitetään TableGateway ohjaavan objektin luokkaan ja toisessa luodaan tämä kyseinen TableGateway ja käsketään sen käyttää haluttua mallitiedoston objektia. Seuraavaksi Zend Frameworkin config/autoload/global.php-asetustiedostoon määritellään tietokannan käyttämä ajuri, käytettävän tietokannan nimi ja osoite, sekä määritellään ServiceManager käyttämään Zend Frameworkin sovitinluokkaa, jotta se voisi kääntää tietokantakomennot halutulle tietokantamoottorille. (Zend Technologies Ltd 2012a.)

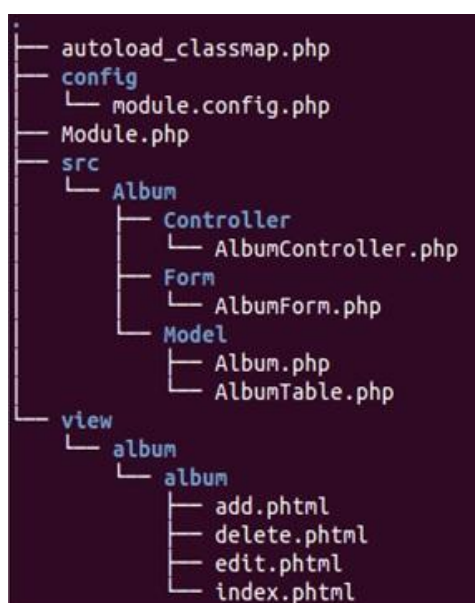
Zend Frameworkissa on valmiit tietokantafunktiot valitsemiselle, lisäämiselle, päivittämiselle ja poistamiselle. Funktioita käytetään kutsumalla Zend Frameworkin TableGateway-objektin metodeja. Esimerkiksi tietokannasta valitaan tietoa näin:

```
$rowset = $this->tableGateway->select(array('id' => 1));
```

(Zend Framework Ltd 2012h.)

Rakenne

Zend Frameworkissa ei ole valmista hakemistorakennetta projektille, vaan käyttäjän on itse luotava hakemistot käsin. Projektihakemisto luodaan module-hakemiston sisään ja sinne luodaan alihakemistot config, src ja view (kuva 2). Config-hakemistoon tulee nimensä mukaisesti projektin asetukset ja ne tallennetaan module.config.php-tiedostoon. Tiedosto palauttaa taulukon, jonka sisällä on kaksi taulukkoa: controllers ja view_manager. Controllers-taulukko sisältää taulukon, johon on listattu kaikki projektissa käytettävät käsittelijätiedostot ja niiden sijainnit. View_manager-taulukko sisältää taulukon, johon on asetettu näkymätiedostojen sijainti, jotta Zend Framework pystyisi löytämään ne. (Zend Framework Ltd 2012f.)



Kuva 2. Zend Framework -projektin hakemistorakenne.

Projektin juureen lisätään Module.php-tiedosto, joka sisältää Module-luokan, jonka sisällä on kaksi metodia. Näistä toinen palauttaa äsken luodun asetustiedoston ja toinen ClassMapAutoloaderille annetut luokat ja niihin liittyvien tiedostojen sijainnit sekä StandardAutoloaderille määrätyt luokat ja niiden sijainnit. Esimerkissä ClassMapAutoloaderia ei käytetty ja sen palauttama tiedosto pistettiin palauttamaan tyhjä taulukko. Lopuksi ModuleManagerille kerrotaan projektin olemassaolosta muokkaamalla Zend Frameworkin /config/application.con-

fig.php-tiedostoa lisäämällä modules-tilukseen uuden projektin nimi. (Zend Framework Ltd 2012f.)

Verkko-osoitteen reititys tiettyyn tiedostoon tapahtuu projektin /config/module.config.php-tiedostossa luomalla sinne taulukko reitittäjälle, joka sisältää reitin pohjan, esimerkiksi /projekti[:action][:id]. Hakasulkeiden sisällä oleva action merkitsee käsittelijätiedostoissa määriteltyjä toimintoja, kuten poistamista, lisäämistä ja niin edelleen. Id merkitsee muuttujaa, kuten esimerkiksi poistettavan uutisartikkelin tunnustetta. (Zend Technologies Ltd 2012a; 2012g)

Käsittelijätiedostot luodaan projektin Controller-hakemistoon ja toiminnot lisätään käsittelijäluokan sisään julkisina metodeina. Verkko-osoitteeseen lisätty toiminto kutsuu suoraan kyseistä toimintometodia luokasta ja suorittaa sen. Näkymätiedostot sijaitsevat projektin view-kansiossa kahden projektin nimellä varustetun hakemiston alla ja ne ovat muodoltaan .phtml-tiedostoja, joihin voidaan kirjoittaa tavallista PHP-koodia ja HTML:a sekaisin, kuten tavallisessakin .php-tiedostossa. Käsittelijätiedosto tuo tiedot mallin kautta tietokannasta näkymälle ViewModel-objektin avulla. Näkymästä tietoon voidaan viitata käymällä tietokantakyselyn tulokset läpi foreach-ohjauksella ja viittaamalla suoraan objektin ominaisuuksiin esimerkiksi \$asiakas->etunimi. (Zend Technologies Ltd 2012a; 2012g)

Zend Frameworkin dokumentaatio oli mielestäni puutteellinen. Ohjeita seuraamalla ei saanut toimivaa runkosovellusta, vaan jokaisen kappaleen päätteeksi jouduin lukemaan kappaleen kommentit, joissa tarjottiin ratkaisuja ohjeiden toimimattomaan koodiin. Myöskään koodia ei kommentoitu syvällisemmin juuri ollenkaan ja usein ohjeissa ei kerrottu, miksi luokan piti periä jokin Zend Frameworkin oma luokka.

3.3 CodeIgniter

CodeIgniterin **asennus** on helppoa: käyttäjän tarvitsee vain ladata CodeIgniterin asennuspaketti virallisilta CodeIgniterin kotisivuilta ja purkaa se palvelimen juureen. Tämän jälkeen käyttäjän pitää antaa palvelimensa perusverkko-osoite config.php-tiedoston \$config['base_url'] -muuttujaan sekä mahdollinen sa-

lausavain `$config['encryption_key']` -muuttujaan, mikäli projektissa halutaan käytettävän salausta tai istuntoja. Jos projektissa on tarkoitus käyttää tietokantaa, sen asetukset tulee syöttää `database.php`-tiedoston muuttujiin. (EllisLab Inc 2012e.)

Tietokanta

CodeIgniter ei sisällä Yii Frameworkin tapaista automaattista CRUD-operaatioiden luomista, vaan kaikki tietokantaa käsittelevät toiminnot pitää käyttäjän itse ohjelmoida mallitiedostoihin. CodeIgniterissä on kuitenkin valmiit Active Record-funktiot yleisimmille tietokantakomennoille, joten käyttäjän ei ole pakko kirjoittaa puhdasta PHP:n PDO:ta tietokantojen käsittelyyn. Active Record mahdollistaa saman tietokantakyselyn käytön eri tietokantaohjelmistoissa, koska kunkin tietokantaohjelmiston sovitin generoi kyselysyntaksin itseään varten. Esimerkki-tietokantakomento kirjoitettaisiin näin:

```
$this->db->select('id, username, profile');  
$this->db->from('tbl_user');  
$this->db->join('tbl_profile', 'tbl_profile.user_id =  
tbl_user.id');  
$this->db->where('tbl_user.id, $id');  
$query = $this->db->get();
```

Komento näyttää hyvin paljon samalta kuin Yii Frameworkin vastaava komento, mutta CodeIgniterissa ei tarvitse käyttää taulukkoa komennon rakentamiseen, vaan funktioita voidaan käyttää erillään. Myös taulukon käyttö on mahdollista joidenkin tiettyjen tietokantakomentojen rakentamiseen. Olen samaa mieltä CodeIgniteristä kuin Yii Frameworkista, eli näen CodeIgniterin tietokantakomennoille käyttöä isommissa tietokantakokonaisuuksissa, joissa PDO muuttuisi epäluettavaksi. (EllisLab Inc 2012a.)

Rakenne

Kuvassa 3 esitetään CodeIgniter-projektin hakemistorakenne. Kaikki kuvan hakemistot ovat CodeIgniterin omia. Toisin kuin Yii Frameworkissa, hakemisto, johon projektit tallennetaan, sijaitsee samalla hakemistotasolla kuin CodeIgniterin järjestelmätiedostot.



Kuva 3. CodeIgniter-projektin hakemistorakenne.

CodeIgniter on ainoa vertailussa mukana oleva verkkosovelluskehys, jonka mukana tulee käyttöohjeet joita voi selata yhteydettömässä tilassa. CodeIgniterissa käsittelijä-, malli- ja näkymätiedostot luodaan käsin toisin kuin Yii Frameworkissa, jossa kaikki voidaan halutessaan generoida giillä. CodeIgniterissa käsittelijä-

, malli- ja näkymätiedostoille on varattu omat kansionsa: controllers, models ja views -kansiot.

Ensimmäisenä käyttäjän tulee luoda käsittelijätiedosto controllers-kansioon. Käsittelijätiedosto voi olla nimeltään mitä tahansa, kunhan se on PHP-muodossa. Käsittelijätiedoston sisään luodaan luokka, jolla tulee olla sama nimi kuin käsittelijätiedostollakin ja sen täytyy periä CI_Controller-luokka, jotta uudet sivut voisivat käyttää sen metodeita ja muuttujia. Seuraavaksi käyttäjän tulee tehdä luokan sisään view-metodi, jonka tehtävänä on tulostaa haluttu sivu loppukäyttäjälle ja tehdä muita sivun tulostamista auttavia tehtäviä, kuten tarkastaa onko loppukäyttäjän haluamaa sivua olemassa palvelimella. Sivua ladataan seuraavalla komennolla: `$this->load->view('hakemisto/sivun_nimi', $data)`. Sivunlatauskomentoja voidaan käyttää käsittelijässä niin paljon kuin halutaan, esimerkiksi ladataan ensin sivustolla yleisesti käytetty valikkorakenne, joka on erillisenä PHP-tiedostona, minkä jälkeen ladataan sitten muu sivun sisältö. Tällä tavalla käyttäjä voi käyttää sivupohjia, eikä hänen tarvitse tehdä ulkoasua erikseen joka sivulle. `$data`-taulukkoon käyttäjä voi tallentaa mitä tahansa näkymätiedoston tarvitsemia muuttujia, kuten sivun otsikon tai päivämäärän. Jos esimerkiksi halutaan lähettää päivämäärä näkymätiedostolle, tallennetaan se ensin `$data`-taulukkoon komennolla `$data['date'] = date("d.m.Y")` ja näkymätiedostossa siihen viitataan vain käyttämällä muuttujaa `$date`. (EllisLab Inc 2012h; 2012i.)

Mallitiedosto luodaan CodeIgniterissa application/models-kansioon PHP-tiedostona. Tiedostoon luodaan luokka mallitiedoston nimellä ja se perii CI_Model-luokan. Luokan konstruktorissa ladataan tietokanta komennolla `$this->load->database()`. ja luodaan halutut tietokantaa käsittelevät metodit erillisinä metodeina konstruktorin perään. Jotta mallia voidaan hyödyntää käsittelijätiedostossa, pitää se ladata käsittelijän konstruktorissa komennolla `$this->load->model('malli')`, mutta ensin käsittelijän konstruktorissa pitää kutsua CI_Controller-luokan `__construct()`-metodia, jotta malli voidaan ladata. Tämän jälkeen näkymätiedostosta voidaan kutsua mallin funktioita esimerkiksi näin: `$this->malli->funktio()`. (EllisLab Inc 2012h.)

CodeIgniterissa voidaan reitittää sivuja tiettyyn verkko-osoitteeseen. Normaalisti ilman reititystä tiettyyn sivuun pääsee käsiksi menemällä verkkoselaimella esimerkiksi osoitteeseen `esimerkki.fi/luokka/metodi/id/`, mutta `/application/config/routes.php` -tiedoston kautta voidaan määrittää haluttu reitti sivuille. Reitti määritellään antamalla arvot `$route`-taulukkoon. `$route['default_controller']` -avaimen arvo, esimerkiksi `$route['default_controller'] = 'pages/view'`, määrittää minne selain ohjaa käyttäjän, jos verkko-osoitteen perässä ei ole argumenttia ja `$route['(:any)']`-avaimen arvo, esimerkiksi `$route['(:any)'] = 'pages/view/$1'`, yrittää etsiä käyttäjän antamaa argumenttia määritellystä kohteesta. (EllisLab Inc 2012h; 2012i.)

3.4 CakePHP

CakePHP **asennetaan** lataamalla virallinen asennuspaketti sen kotisivuilta ja purkamalla pakatun kansion sisältö suoraan palvelimen juureen. Tällä kertaa ei siis luoda verkkosovelluskehykselle omaa kansiota, johon paketin sisältö purettaisiin. Seuraavaksi käyttäjän on asetettava kirjoitusoikeudet hakemistolle `app/tmp`, jotta CakePHP voisi kirjoittaa sinne väliaikaistiedostoja. CakePHP:n asetustiedosto, `core.php`, sijaitsee `app/Config`-hakemistossa ja sieltä pitää muuttaa `Security.salt` ja `Security.cipherSeed` -muuttujien oletusarvoja. Näitä muuttujia käytetään tiivisteiden salttina ja merkkijonojen salaukseen sekä purkamiseen. CakePHP käyttää Apache-palvelinohjelman `mod_rewrite`-moduulia ohjaamaan verkko-osoitteen projektin sijaintiin palvelimella, mutta `mod_rewrite` ei toiminutkaan palvelimelläni, joten jouduin ottamaan kyseisen ominaisuuden pois käytöstä poistamalla kommentin riviltä `Configure::write('App.baseUrl', env('SCRIPT_NAME'))` ja `.htaccess`-tiedostot CakePHP:n juuri-, `app`- ja `app/webroot`-hakemistosta. Tietokanta saadaan käyttöön asettamalla tietokannan asetukset `app/Config/database.php`-tiedostoon. Käyttäjä voi asettaa monen eri tietokannan asetukset erillisiin taulukoihin `DATABASE_CONFIG`-luokan sisällä. (Cake Software Foundation Inc 2012e.)

Tietokanta

CakePHP sisältää valmiit funktiot tietokannan käsittelyyn. Tyypillisin tapa hakea tietoa tietokannasta on find-funktio, johon voidaan asettaa ehtoja funktion parametrina. Käytössä olevat parametrit ovat first, joka palauttaa ensimmäisen tuloksen, count taas palauttaa löydettyjen tuloksien määrän, all palauttaa kaikki löydetyt tulokset, list palauttaa indeksoidun taulukon, johon tulokset on tallennettu, threaded palauttaa sisäkkäisen taulukon ja neighbors palauttaa haettujen ehtojen täyttämien tuloksen molemmilta puoliilta yhden tuloksen. Myös omien hakuehtoien luominen find-funktiolle on mahdollista. Uuden hakuehdon nimi lisätään mallitiedostoon \$findMethods-muuttujaan ja sitten luodaan metodi, johon täsmennetään haku ehdot. CakePHP sisältää myös Magic Find Types -hakufunktiot findAllBy ja findBy. Nämä funktiot toimivat siten, että niiden perään kirjoitetaan halutun tietokantakentän nimi ja annetaan hakuehto ensimmäisenä parametrina, esimerkiksi findAllByEtunimi('Ville'). Esimerkkietokantakysely näyttää CakePHP:llä tältä:

```
$query = $this->User->find('all', array(
    'conditions' => array('User.id' = $id)
));
```

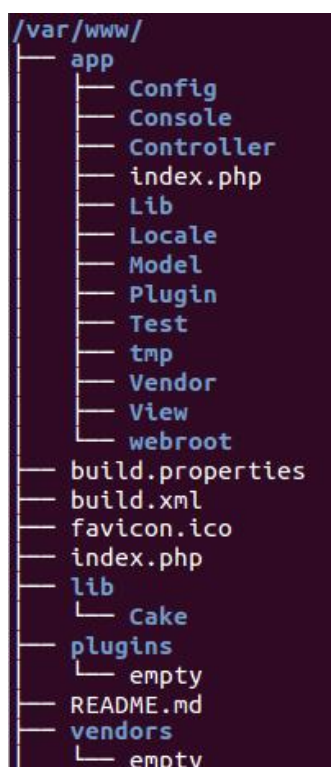
Tiedon hakeminen useasta tietokantamallista ei ole mahdollista suoraan käyttämällä find-funktioita, vaan käyttäjän on luotava mallitiedostoon suhde mallien välille. Eri suhdetyypit ovat yksi-yhteen, yksi-moneen, moni-yhteen ja moni-moneen. Suhde luodaan tekemällä julkinen suhdemuuttuja haluttuun mallitiedostoon ja antamalla sille taulukkona suhteen toisena osapuolena olevan mallin nimi ja halutut avaimet, jotka vaihtelevat eri suhdetyyppien välillä. (Cake Software Foundation Inc 2012a.)

Tiedon tallentaminen tietokantaan on tehty CakePHP:ssä helpoksi: käyttämällä käsittelijätiedostossa FormHelper-lomakeavustajaa tai mallin find-metodeja, eikä käyttäjän tarvitse huolehtia datan asettelusta. Tietoa tallennetaan tietokantaan kutsumalla dataa save-metodin sisällä, esimerkiksi \$this->Test->save(\$this->request->data)). Tiedon päivittäminen tapahtuu updateAll-metodin

kautta syöttämällä sille päivitettävät kentät ja halutut arvot yhtenä taulukkona, ja ehdot erillisenä taulukkona. Tiedon poistamiselle on omistettu kaksi funktiota: delete ja deleteAll. Delete-funktio poistaa tietueen sille parametrina syötetyn tunnisteiden perusteella ja deleteAll on muuten samanlainen funktio kuin delete, mutta sille voidaan antaa parametrina muitakin poistoehdoja kuin tunniste. (Cake Software Foundation Inc 2012f; 2012g.)

Rakenne

Kuten CodeIgniterissäkin, CakePHP:ssä projektikansio sijaitsee samalla tasolla kuin järjestelmätiedostot (kuva 4) ja käsittelijät, mallit ja näkymät kirjoitetaan käsin omiin kansioihinsa. CakePHP:ssä projektikansiota ei eroteta: käsittelijä luodaan app/Controller-hakemistoon samalla periaatteella kuin muissakin verkkosovelluskehyksissä, eli tehdään luokka, joka sisältää metodit jokaiselle näkymälle. Jos käyttäjä esimerkiksi antaa funktiolle nimen sivu(), on kyseisen funktion toiminnallisuus käytettävissä esimerkiksi osoitteessa www.esimerkki.com/cakephp/sivu. CakePHP:n nimeämiskäytännön mukaan käsittelijä pitää nimetä käyttämällä monikkoa halutusta nimestä ja lisäämällä perään sana Controller, esimerkiksi TestsController. Käsittelijään luodun luokan pitää periä ApplicationController-luokka, johon voidaan luoda koko sovelluksen kattavia metodeita ja samalla se perii CakePHP:n Controller-päälukon. Käsittelijän funktio lähettää dataa eteenpäin näkymälle set()-metodilla ja malliin voidaan viitata esimerkiksi komennolla `$this->Test`, olettaen että mallin nimi on Test. (Cake Software Foundation Inc 2012c.)



Kuva 4. CakePHP:n hakemistorakenne.

Näkymä luodaan app/View-hakemistossa hakemistoon, joka on nimetty käsittelijätiedoston mukaan. Esimerkiksi jos käsittelijätiedosto on nimeltään TestsController, annetaan hakemistolle nimeksi Tests. Näkymätiedosto on muista verkosovelluskehyksistä poiketen tiedostotyyppiltään .ctp (CakePHP template), eikä .php ja sille on annettava nimeksi käsittelijätiedostossa käytetty funktion nimi. Tiedostotyyppistä huolimatta näkymätiedoston kirjoittamiseen käytetään HTML:ää ja PHP:tä. Näkymätiedostossa voidaan viitata käsittelijän set()-metodilla käsiteltyyn muuttujaan samalla nimellä, kuin mikä sille annettiin saman metodin sisällä. (Cake Software Foundation Inc 2012c.)

Mallitiedosto luodaan app/Model-hakemistoon ja siihen luodun luokan on perittävä AppModel-luokka, joka toimii samalla periaatteella kuin käsittelijän ApplicationController-luokka. Koska malli perii Model-päälukon, se luo dynaamisesti malliobjektin, joten joissain tapauksissa mallitiedostoon ei tarvitse kirjoittaa muuta kuin itse malliluokan luominen. (Cake Software Foundation Inc 2012c; 2012h.)

3.5 Symfony

Symfony **asennetaan** lataamalla ja purkamalla sen asennuspaketti palvelimen juureen. Tämän jälkeen tarkistetaan vastaavatko palvelimen asetukset ja asennetut ohjelmat Symfonyn vaatimuksia suorittamalla ohjelma `app/check.php` komentoriviltä. Komentoriville tulostuu lista kaikista vaatimuksista ja niiden edessä on teksti OK, ERROR tai WARNING riippuen siitä, löytyykö ominaisuutta palvelimelta. ERROR-teksti tulee pakollisen ominaisuuden puuttuessa ja WARNING valinnaisen puuttuessa. Omalla palvelimelläni sain varoitukset lyhyemmän PHP-koodin aloitustunnisteen käytön mahdollistavan `short_open_tag` -asetuksen päällä olemisesta sekä PHP:n internationalisaatiolisäosan ja PHP:n APC-kiihdyttimen puuttumisesta. Ainoan virheen sain PHP:n asetustiedostosta puuttuneesta aikavyöhykeasetuksesta. Short-open-tag- ja aikavyöhykeongelmat korjasin muokkaamalla PHP:n `php.ini`-asetustiedostosta vaaditut rivit Symfonyn haluamiksi ja PHP:n lisäosan ja kiihdyttimen asensin suoraan Ubuntu Linuxin pakettivarastosta. (Sensio Labs International 2012d; 2012f.)

Kun palvelimelle asennetut ohjelmat ja asetukset vastaavat Symfonyn vaatimuksia, voidaan siirtyä itse asennukseen navigoimalla verkkoselaimella osoitteeseen `http://localhost/Symfony/web/config.php`. Tässä vaiheessa on hyvä varmistaa, että `app/config/parameters.ini` -asetustiedosto sekä `app/cache`- ja `app/logs` -hakemistot ovat verkkopalvelimen kirjoitettavissa ja tämän jälkeen luodaan tietokanta Symfonyn käyttöön. Asennussivulla kysytään Symfonylle tarkoitetun tietokannan tietoja, jotka käyttäjän pitää syöttää. Seuraavalla sivulla generoidaan tiiviste, jota käytetään muun muassa Cross-site request forgery -haavoittuvuutta varten. (Sensio Labs 2012d.)

Symfonyn asentaminen vaati paljon enemmän palvelinympäristön asetusten ja kirjoitusoikeuksien muokkaamista verrattuna muihin verkkosovelluskehysiin, Symfonyn varsinaiset asetukset tosin hoidettiin hyvin verkkokäyttöliittymän kautta. Symfony oli myös ainoa verkkosovelluskehys, joka ei toiminut suoraan palvelimelleni asennetuilla ohjelmilla.

Tietokanta

Symfony sisältää tietokannan käsittelyyn Doctrine ORM -kirjaston. Doctrine ORM:n tarkoitus on tarjota tehokkaat työkalut tietokannan käsittelyn helpottamiseksi. Doctrine ORM:n käyttö ei ole kuitenkaan pakollista, vaan käyttäjä voi käyttää halutessaan raakoja tietokantakyselyitä Doctrine DBAL -abstraktiokerroksen avulla. Symfonyssa projektit tunnetaan nimellä bundle, nippu. Aivan aluksi käyttäjän on luotava entiteettiluokka nipun Entity-hakemistoon, johon listataan tietokantataulun kentät muuttujina. Tämän jälkeen lisätään samaan luokkaan objektien kartoitustiedot kommentteina. Kommentoinnin lisäksi kartoitustiedot voidaan merkitä YAML- ja XML-merkintäkielillä, mutta ne pitää tehdä eri tiedostoon nipun /Resources/config/doctrine-hakemistossa. (Sensio Labs 2012b.)

Seuraavaksi käyttäjän täytyy generoida metodit muuttujien asettamiseen ja palauttamiseen käyttämällä komentorivillä komentoa `php app/console doctrine:generate:entities Organisaatio/TestiBundle/Entity/Objektin_nimi`. Samaa komentoa voidaan käyttää myös koko organisaation tai nipun entiteettien generoimiseen käyttämällä komennossa pelkkää organisaation nimeä tai organisaation ja nipun nimeä peräkkäin. Jos tietokannassa ei ole valmiina tauluja, voidaan ne generoida Doctrinen avulla komennolla `php app/console doctrine:schema:update -force`. Aiemmin luotu entiteetti pitää liittää nipun Default-Controller-käsittelijätiedostoon käyttämällä use-operaattoria, jonka jälkeen käyttäjä voi luoda tietokannan käsittelymetodit samaan käsittelijään. (Sensio Labs 2012b.)

Tiedon tallentaminen tapahtuu luomalla tietokantataululle objekti, johon asetetaan arvot aiemmin generoiduilla asettajametodeilla. Tämän jälkeen haetaan muuttujaan Doctrinen entiteetin hallintaobjektia, joka vastaa tiedon tallentamisesta tietokantaan ja tiedon hakemisesta tietokannasta. Seuraavaksi kerrotaan hallintaobjektille, mitä objektia sen tulee käsitellä kutsumalla persist-funktiota ja syöttämälle sille parametrina haluttu objekti. Lopuksi tieto tallennetaan tietokantaan flush-funktiolla. Tiedon tallentaminen näyttää siis tältä:

```
use Organisaatio\AsiakasBundle\Entity\Customer;
use Symfony\Component\HttpFoundation\Response;
```



```

public function createAction()
{
    $customer = new Customer();
    $customer->setName('Antti Asiakas');
    $customer->setAddress('Mannerheimintie 30');
    $customer->setTelephone('0501234567');

    $em = $this->getDoctrine()->getManager();
    //Haetaan entiteetin hallintaobjektia
    $em->persist($product) // Kerrotaan mitä objektia
    käsitellään
    $em->flush(); tallennetaan

    return new Response('Asiakas nro. `.`.$customer-
    >getId().` luotu.`'); // Palautetaan teksti tallennuksen on-
    nistumisesta
}

```

(Sensio Labs 2012b.)

Tiedon hakeminen tietokannasta on vieläkin yksinkertaisempaa. Käyttäjän on kutsuttava objektiin jotakin varastoa, joka hakee tietyn luokan entiteettejä. Varasto pitää sisällään erilaisia hakufunktioita, muun muassa "hae pääavaimella", "hae tietyn kentän tietyllä arvolla" ja "hae kaikki". Käyttäjän on liitettävä haluttu hakufunktio varaston hakemisen perään, esimerkiksi:

```

$customer = $this->getDoctrine()-
>getRepository('OrganisaatioAsiakasBundle:Customer')-
>find($id);

```

(Sensio Labs 2012b.)

Kyseinen komento hakee tietokannasta asiakkaan, jonka asiakasnumero täsmää annetun \$id-muuttujan kanssa ja tallentaa sen \$customer-objektiin. Tämän jälkeen käyttäjä voi vielä käsitellä haettua tietoa tai lähettää objektin eteenpäin sivupohjalle loppukäyttäjälle näytettäväksi.

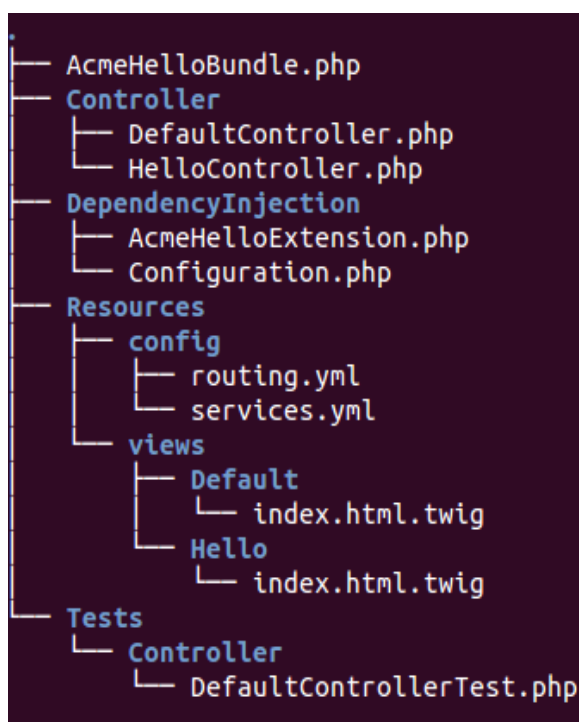
Rakenne

Uusi nippu luodaan käyttämällä Symfonyn hakemistossa komentoa `php app/console generate:bundle --namespace=organisaation_nimi/projektin_nimi --format=yml`, joka generoi rungon nipulle ja rekisteröi sen Symfonyn ytimen kanssa. Myös Symfony tukee reititystä. Reititystiedostoja on kaksi: kaikkien nipujen käytettävissä oleva reititystiedosto `/app/config/routing.yml` ja nipun yksilöllinen reititystiedosto `/src/organisaation_nimi/projektin_nimi/resources/config/routing.yml` (kuva 5). Kaikkien nipujen käytössä olevasta reititystiedostosta voidaan määrittää tuotavaksi jokaisen nipun oma reititystiedosto tai siihen voidaan asettaa nipun reititysasetukset suoraan.

Selkeyden varmistamiseksi käyttäisin ensimmäisenä mainittua reititystiedostoa vain jokaisen nipun reititystiedoston tuomiseen ja pitäisin varsinaisen reitityksen nipujen omissa reititystiedostoissa, jotta ne eivät menisi keskenään sekaisin. Käytännössä reititys tapahtuu määrittelemällä rakenne, pattern, jota verkko-osoitteen on vastattava ja luomalla defaults-taulukkoon kyseisellä verkko-osoitteella avautuva käsittelijätiedosto, sen luokka ja metodi. Esimerkiksi:

```
testi:
    pattern: /testi/{nimi}
    defaults: { _controller: OrganisaatioTestiBundle:Testi:index }
```

Esimerkissä oleva reititystiedosto siis määrittää, että menemällä verkko-selaimella sijaintiin `/testi/{nimi}`, käyttäjälle suoritetaan Organisaatio-nimisen organisaation Testi-nipun Testi-käsittelijätiedostosta metodi `index`, joka voisi vaikkapa renderöidä sivuston etusivun. Aaltosulkujen sisällä oleva nimi kuvaa muuttujaa, esimerkiksi jonkin tietyn uutisartikkelin tunnusta tai käyttäjätunnusta ja niin edelleen. Reititystiedostoja voidaan toteuttaa YAML- ja XML-merkintäkielillä tai PHP:llä. (Sensio Labs 2012a.)



Kuva 5. Symfonyprojektin hakemistorakenne.

Käsittelijätiedosto sijaitsee nipun Controller-hakemistossa ja on rakenteeltaan hyvin samanlainen muidenkin verkkosovelluskehysten käsittelijätiedostojen kanssa. Käsittelijätiedosto on siis PHP-tiedosto, joka sisältää käsittelijäluokan, johon on toteutettu metodeja jokaiselle sivun tapahtumalle, kuten sivun renderöinnille, käyttäjän sisään kirjautumiselle ja niin edelleen. Ainoastaan käsittelijän pääluokan periminen on toteutettu eri tavalla kuin muissa verkkosovelluskehyksissä. Aivan ensimmäisenä määritellään käsittelijälle nimiavaruus ja tuodaan käsittelijän pääluokka use-operaattorilla käytettäväksi. Use-operaattori myös lyhentää pääluokan nimen halutuksi: use Symfony\Bundle\FrameworkBundle\Controller\Controller-tiedostoon voidaan nyt viitata vain käyttämällä sanaa Controller. Pääluokka otetaan käyttöön käsittelijätiedostossa käyttämällä avainsanaa extends luokkaa määrittäessä, esimerkiksi: class TestiController extends Controller. (PHP Group 2012a; Sensio Labs 2012a.)

Symfonyssa on tuki sivupohjille, jotka toimivat kuten näkymätiedostot muissa verkkosovelluskehyksissä. Sivupohjat sijaitsevat nipun /Resources/views/nipun_nimi/-hakemistossa ja ovat tiedostomuodoltaan joko PHP- tai twig-tiedostoja. Ne renderöidään käsittelijällä palauttamalla render-funktio, johon on syö-

tetty käytettävän sivupohjan sijainti ja taulukossa sivulla käytetyt muuttujat. Esimerkiksi:

```
return
$this>render('OrganisaatioTestiBundle:Testi:index.html.twig
', array('otsikko => $otsikko));
```

Twig on SensioLabsin kehittämä sivupohjamoottori, joka on suunniteltu olemaan nopea, turvallinen sekä joustava. Twig sisältää muun muassa paranneltuja tapoja käyttää if-lauseita ja toistorakenteita, muuttujien käsittelyä suodattimilla, makroja sekä operaattoreita. Twigin tärkein ominaisuus on sivupohjan periminen: käyttäjä voi rakentaa valmiin sivupohjan, joka sisältää kaikki sivun peruselementit, ja määrittää niihin bloqueja, joiden avulla ne voidaan korvata lapsisivupohjasta käyttämällä kyseistä blockin nimeä. Esimerkiksi sivuston otsikon korvaaminen tapahtuisi näin:

Perussivupohja:

```
<title>{% block title %}{% endblock %}</title>
```

Lapsisivupohja korvaa perussivupohjan title-elementin:

```
{% block title %}Otsikko{% endblock %}
```

(Sensio Labs 2012a; 2012h; 2012i.)

4 Toteutus

Kehitin verkkosovellusta omalle työasemalleni asennetussa virtuaaliympäristössä, jonka käyttöjärjestelmänä toimi Ubuntu Linuxin versio 12.04. Työasemalla oli myös palvelinvalmiudet eli asennettuna oli Apache HTTP-palvelinohjelma, PHP:n versio 5.4 ja MySQL-tietokantapalvelimen versio 5.5. Tietokannan hallintaan käytin verkkoselaimen kautta käytettävää phpMyAdmin-ohjelmaa, jolla onnistuvat samat toiminnot kuin perinteisesti komentorivin kautta. Itse HTML5-, PHP-, RGraph- ja jQuery-koodit kirjoitin gedit-tekstinkäsittelyohjelmalla, josta löytyy hyvä tuki HTML-, PHP-, CSS- ja JavaScript-kielille.

4.1 Vaatimukset

Suunnittelimme Jaakko Vanhalan kanssa verkkosovelluksen vaatimusmäärittelyn ja päätimme mihin tilaan verkkosovellus on saatava marraskuuhun mennessä. Sovelluksen on tarkoitus olla käyttäjäystävällinen, vuorovaikutteinen ja avoin. Käyttäjäystävällisyyteen panostin toteuttamalla mahdollisimman selkeän ja raikkaan näköisen käyttöliittymän, jota on helppo käyttää. Vuorovaikutteisuus ja avoimuus ilmenevät kommentoitavilla ja eteenpäin jaettavilla HOPSeilla, kursseille ilmoittautumisina ja opastavana käyttöliittymänä. Muita vaadittuja ominaisuuksia ovat käyttäjän rekisteröinti ja kirjautumisjärjestelmän toteuttaminen, kurssien valinta HOPSiin ja niiden tietojen tarkastelu, kursseille ilmoittautuminen ja ilmoittautumisen hyväksyminen (opettaja suorittaa).

Sovellus vaatii kaksi käyttäjäryhmää: opiskelijat ja opettajat. Opettajat eroavat opiskelijoista siten, että opettajat voivat hyväksyä ilmoittautumisia kursseille, eivätkä he voi itse ilmoittautua niille. Lisäksi jokaiselle kurssille määritellään oma opettaja. Opiskelijat voivat luoda useita erilaisia HOPSeja ja halutessaan julkaista tietyn HOPSin. HOPSin julkaiseminen tarkoittaa käytännössä sitä, että verkkosovellus generoi sille satunnaisen kirjaimista ja numeroista koostuvan verkko-osoitteen. Tällä ratkaisulla varmistan, ettei kukaan ulkopuolinen voi arvata HOPSin verkko-osoitetta esimerkiksi käyttäjätunnuksen tai käyttäjän tunnisteen perusteella. Näin vain henkilöt, joille linkki on jaettu, pääsevät tarkastelemaan HOPSia.

Vaaditut kurssin tiedot ovat opintojakson nimi, toteutusajankohta, ilmoittautumisajankohta, opettajan nimi ja sähköpostiosoite, kurssin lisätiedot ja opintopistemäärä. Kurssin valitsemiseksi pitää lisätä kurssin käyttäjän HOPSiin ja nostaa tuntien määrää kaaviossa kyseisen kurssin toteutusajan kohdalle. Kuormituksen seurantakaaviossa vaaka-akselille on sijoitettu opiskelukuukaudet ja pystyakselille arvioitu tuntimäärä, joka opiskelijan tulisi käyttää päivässä opiskeluun suorittaakseen kurssin optimaalisesti. Tässä on oletettu, että opiskelija työskentelee kuutena päivänä viikossa aktiivisesti.

Jokaisen julkaistun HOPSin sivulla täytyy sijaita käyttäjien kommentit kyseiselle HOPSille sekä lomake kommenttien lähettämiseen. Näissä kommentteissa opettajat ja opiskelijat voivat keskustella kyseisen HOPSin sisällöstä. Verkkosovellus tehostaa opiskelijan ohjausta ilman tarvetta sähköpostiviestittelylle tai tapaamisille paikanpäällä.

4.2 Tietokanta

Tietokannaksi valitsin sovellukselleni MySQL:n, koska minulla on siitä eniten kokemusta. Tietokantani toimii InnoDB-tietokantamoottorilla ja merkistöstandardina käytän UTF-8:aa. Yhteensä tietokannassa on 22 tietokantataulua, joista itse tein viisi: comment, course, courselist, hops ja registration. Loput tauluista ovat käyttäjänhallintaan käytettävän Yii-user-management-lisäosan tekemiä tauluja.

Comment-tauluun tallennetaan käyttäjien lähettämiä viestejä eri HOPSeihin. Taulu sisältää kentät kommentin tunnisteelle, käyttäjätunnisteelle, HOPSin tunnisteelle, kommentille sekä kommentin lähetysajankohdalle. Course-tauluun on tallennettu kaikki kurssit. Sen kentät ovat kurssin tunniste ja nimi, laajuus opin-
topisteinä, aloitus- ja lopetuspäivämäärät, ilmoittautumisen alkamis- ja loppu-
mispäivämäärät, kurssin kuvaus, opettajan tunniste, kyseisen kurssin Moodle-
osoite ja Moodle-tilan avain. Courselist-taulua käytin yhdistääkseni course-tau-
lun kurssit hops-taulun kanssa. Tässä taulussa on vain kolme kenttää: tunniste,
HOPSin tunniste ja kurssin tunniste. Hops-tauluun tallennetaan HOPSien tun-
niste, satunnainen kymmenmerkkinen merkkijono julkista HOPSia varten,
HOPSin nimi, HOPSin luoneen opiskelijan tunniste ja luomispäivämäärä. Regi-
stration-tauluun tallennetaan käyttäjän ilmoittautumiset eri kurseille. Sen kent-
tiä ovat tunniste, ilmoittautumisen tila (hyväksyty, käsittelemätön, hylätty), arvo-
sana (0-5, -1, jos kurssi on kesken), kurssin tunniste ja käyttäjän tunniste.

Yii-user-management-lisäosan käyttö loi monta tälle sovellukselle turhaa tieto-
kantataulua, mutta lisäosan oikean toiminnan varmistamiseksi en lähtenyt muut-
tamaan tai poistamaan tauluja. Käyttämäni Yii-user-managementin taulut ovat
profile, role, user ja user-role. Profile-taulu on relaatiossa user-taulun kanssa ja
sinne tallennetaan muun muassa profiiliin tunniste, käyttäjän tunniste, luontipäi-

vämäärä, profiilin yksityisyys, käyttäjän etu- ja sukunimi ja sähköpostiosoite. Role-aulussa ovat tallennettuna palvelussa käytettävät eri roolit. User-auluun tallennetaan käyttäjätunniste, käyttäjätunnus, salasana salattuna, käyttäjän luontipäivämäärä, viimeisin vierailu sovelluksessa ja viimeisimmän tehdyn toiminnon aika. User-role-aulussa role-aulun roolit yhdistetään user-aulun käyttäjien kanssa ja sen kentät ovat käyttäjätunniste ja roolitunniste.

4.3 Käyttäjähallinta

Käyttäjähallintaan käytin sovelluksessani kolmannen osapuolen luomaa Yii-user-management Yii Framework -lisäosaa. Sen ominaisuuksia ovat käyttäjä- ja roolihallinta, käyttöoikeudet eri sivuille, käyttäjäprofiilit, käyttäjäryhmät, käyttäjän profiilikuvat, käyttäjän rekisteröinti, salasanan palautus, ystävyudet käyttäjien välillä ja postituskomponentti.

Käyttämäni YUM-versio on 0.8rc6, joka on siis vielä ”julkaisukandidaattias-teella”. Olisin käyttänyt viimeisintä julkaisuversiota eli 0.7:aa, mutta se ei toiminnut nykyisen Yii Framework -version kanssa. Uusinkaan versio ei kuitenkaan toiminut moitteetta, sillä käyttäjän rekisteröiminen ei onnistunut lainkaan ja minun piti kirjoittaa sille oma rekisteröintilomake ja sen toiminnallisuus. Myöskään YUMin luomia tietokantatauluja ei ole millään tavalla linkitetty toisiinsa. Jos käyttäjän poistaa palvelusta, jää hänen profiilinsa, roolinsa, käyttäjäryhmänsä ja käyttöoikeutensa vielä tietokantaan. Rekisteröintiongelmia lukuun ottamatta YUM säästi paljon aikaani muun muassa käyttäjän roolinhallinnallaan ja käyttäjäprofiileilla. (Yii Software LLC 2012f.)

Toteutin sovellukseen kaksi käyttäjäryhmää: opiskelijat ja opettajat. Opiskelijoilla on oikeus tarkastella kursseja ja ilmoittautua niille sekä luoda omia HOPSeja ja jakaa ja muokata niitä. Opettajilla on oikeus hyväksyä, hylätä ja poistaa ilmoittautumisia sekä antaa arvosanoja opintosuorituksille. Molemmat käyttäjäryhmät voivat kommentoida julkisia HOPSeja.

4.4 Tietoturva

Tietoturvaan kiinnitin erityistä huomiota, koska kyseessä on sivusto, joka sisältää opiskelijoiden henkilökohtaisia ja koulutukseen liittyviä tietoja. Kaikki käyttäjätunnukset on suojattu käyttäjän valitsemalla salasanalla ja ne tallennetaan tietokantaan salattuna. Valitsin käyttäjien salasanojen minimipituudeksi 5 merkkiä, etteivät käyttäjät käyttäisi liian helposti murrettavia muutaman merkin salasanoja. Samalla kuitenkin yritin säilyttää käyttäjäystävällisyyden pitämällä salasanan minimipituuden lyhyenä. Käyttäjän salasana tiivistetään PHP:n crypt-funktiolla, jossa käytetään Blowfish-salausmenetelmää ja erillistä salttia. Salttia käytän salasanoissa siksi, ettei salasanoihin päästäisi käsiksi etsimällä salasananatiivisteitä ns. rainbow table -tiedostosta, johon on esilaskettu tiivisteiden tarkistussummat.

Crypt-funktio on toinen PHP:n suositelluista salasanan salaukseen käytettävistä funktioista ja toinen on hash, joka tukee isompaa määrää salausmenetelmiä kuin crypt-funktio, mutta koska se sisältyy PHP:n erilliseen lisäosaan, ei ole varmaa, onko palvelimella hash-funktiota saatavilla. Tästä syystä on varmempaa valita salaukseen käytettäväksi crypt-funktio. Muita funktioita, joilla voisi tiivistää salasanoja, ovat md5 ja sha1 -funktiot, mutta ne ovat suunniteltu tiivistämään merkkijonon nopeasti ja tehokkaasti, joten nykytietokoneilla niiden murtaminen on helppoa. (PHP Group 2012b.)

Käyttäjät voivat julkaista palvelussa henkilökohtaisia opetussuunnitelmiaan esimerkiksi näyttääkseen niitä opinto-ohjaajalleen. Jotta kuka tahansa ei pääsisi käsiksi julkaistuihin opetussuunnitelmiin, toteutin HOPSin julkaisun siten, että sovellus arpoo satunnaisen kymmenmerkkisen merkkijonon julkaistun HOPSin sijainniksi esimerkiksi tähän tapaan: <http://oksj.fi/index.php?r=hops/public-hops&url=euot1wxdk>. Näin HOPSia pääsevät tarkastelemaan vain ne, joille HOPSin julkaissut opiskelija on linkin antanut.

Tietokantaa käsittelin PDO-rajapinnan avulla, joka on turvallisempi kuin sitä edeltävä rajapinta MySQL. MySQL-rajapinta kärsii SQL-injektioista, joissa käyttäjän syöte lisätään SQL-kyselyyn muuttamattomana. Käyttäjä pystyy syötämällä SQL-kieltä SQL-kyselyn perään esimerkiksi poistamaan tietokantatau-

lun DROP TABLE -komennolla tai tehdä salasananatarkistuksesta kommentoidun jos kehittäjä ei ole muistanut käyttää mysql_real_escape_string-funktiota poistamaan käyttäjän syötteestä erikoismerkkejä. PDO-rajapinta tukee valmisteltuja lauseita, joissa tietokannalle kerrotaan tarkkaan millainen tietokantakysely suoritetaan, ennen kuin kyselyyn liitetään yhtään parametria. Näin tietokanta saa tietää tarkalleen mitä muuttujia tulee mihinkin, eikä käyttäjä pääse huijaamaan syöttämällä muuttujaan SQL-kieltä. Estääkseni käyttäjää syöttämästä HTML- ja PHP-kieltä, käytin htmlspecialchars-funktiota, joka muuttaa syötteessä olevat erikoismerkit nimiviittauksiksi ja siten sovellus ei laske niitä suoritettavaksi koodiksi. (PHP Group 2012c; Petersen 2012; Laaksonen 2011.)

5 Tulokset

Toiminnallisen osuuden lopputulokseksi syntyi opiskelijoiden toimintojen osalta toimiva verkkosovellus, joka soveltuu jatkokehitykseen esimerkiksi jossakin tulevassa opinnäytetyössä tai jollakin projektikurssilla. Opettajien ja pääkäyttäjän toiminnot jäivät hieman vajavaisiksi ja vaativat jatkokehitystä, josta kerron myöhemmin Testaus ja jatkokehitys -osiossa.

Tein myös useita UML-kaavioita jatkokehittäjiä varten: Bouml-UML-työkalulla toteutin käyttötapauskaavion (liite 1). Se kuvaa sovelluksen toiminnot ja käyttäjät sekä luokkakaavion tietokannasta (liite 2). Siitä näkyy jokainen tietokannan taulu ja niiden väliset assosiaatiot ja sarakkeet. Lisäksi tein WebSequenceDiagrams-verkkosivuston avulla sekvenssikaavioita joistakin sovelluksen toiminnoista, jotka kuvaavat olioiden vuorovaikutusta eri toimintojen sisällä (liite 3). Sovellukseen on käyttöohjeet, mutta perinteisen erillisen dokumentin sijaan sisällytän ne verkkosovelluksen käyttöliittymään.

5.1 Kurssit

Painamalla sivuston ylälaidan valikosta Kurssit-linkkiä, käyttäjä pääsee tarkastelemaan koko koulutusohjelman kurssitarjontaa, suoritettuja kurssejaan sekä

tekemään ilmoittautumisia (kuva 6). Sivun näkymän toteutin jQueryn haitarivalikolla, johon asetin kolme otsikkoa: Kaikki kurssit, Ilmoittautumiset ja Suoritetut kurssit. Kaikki kurssit -otsikon alle tulostuvat kaikki kurssit, joille opiskelija voi ilmoittautua. Jokainen kurssi tulostetaan omalle rivilleen ja niistä kerrotaan laajuus ja ajankohta. Kurssin nimeä klikkaamalla avautuu lisätietoja kurssista. Käyttäjä voi ilmoittautua kurssille painamalla kurssin kohdalle tulostettua ilmoittautumisnappia. Nappi tulostuu sillä ehdolla, että käyttäjä ei ole jo ilmoittautunut kyseiselle kurssille, ja painalluksen jälkeen se katoaa kurssin vierestä, ilmoitus ilmoittautumisen onnistumisesta tulostetaan käyttäjälle ja ilmoittautuminen siirtyy kurssin vastuopettajalle käsiteltäväksi.

The screenshot shows a web interface for 'OKSJ HOPS Kurssit'. At the top right, there is a user ID '0901455' and a 'Kirjaudu ulos' button. The main heading is 'Kurssilistaus'. A dropdown menu is open, showing 'Kaikki kurssit'. Below this, a table lists courses with columns for 'Nimi ja tunnus', 'Laajuus', and 'Ajankohta'. Each row has an 'Ilmoittaudu' button. The first course is 'BD2001E - HTML and XHTML' with 3 credits and dates from 01.09.2012 to 31.12.2012. Below the table, there is additional information for the first course: 'HTML:n ja XHTML:n perusteet', 'Ajankohta: 01.09.2012 - 31.12.2012', 'Ilmoittautumisaika: 01.05.2012 - 07.09.2012', and 'Vastuopettaja: Nalle Uppo'.

Nimi ja tunnus	Laajuus	Ajankohta	
BD2001E - HTML and XHTML	3 op	01.09.2012 - 31.12.2012	Ilmoittaudu
HTML:n ja XHTML:n perusteet			
Ajankohta: 01.09.2012 - 31.12.2012			
Ilmoittautumisaika: 01.05.2012 - 07.09.2012			
Vastuopettaja: Nalle Uppo			
BD2003 - Linux	3 op	07.10.2012 - 31.12.2012	
BD2203 - PHP-ohjelmointi	3 op	01.03.2012 - 31.05.2012	Ilmoittaudu
BD2205 - Kuvankäsittely ja käyttö	3 op	01.03.2012 - 31.05.2012	Ilmoittaudu
www-dokumenteissa			

Kuva 6. Sovelluksen kurssilistaus, näkyvissä Kaikki kurssit -valikko.

Ilmoittautumiset-otsikon alle tulostetaan kaikki kurssit, joille käyttäjä on ilmoittautunut. Ilmoittautumisen tila ilmoitetaan värikoodilla kurssin tiedoissa: keltainen väri tarkoittaa, että ilmoittautumista ei ole vielä käsitelty, punainen että ilmoittautuminen on hylätty ja vihreä että ilmoittautuminen on hyväksytty. Tässäkin näkymässä kurssista saa lisätietoja klikkaamalla sen nimeä ja tunnusta.

Suoritetut kurssit -otsikon alle tulostetaan kaikki kurssit, joista on annettu arvosana. Kurseista tulostetaan samat tiedot kuin edellisissäkin osioissa, mutta lisäksi tulostetaan vielä kurssin arvosana. Sivun alaosaan tulostetaan suoritet-

tujen kurssien määrä ja niiden keskiarvo. Kuten muissakin osioissa kurssin nimeä klikkaamalla avautuu lisätietoja kurssista.

5.2 HOPS

Henkilökohtaisen opintosuunnitelman näkymän toteutin samalla haitarivalikolla, jota käytin kurssinäkymässä, ja johon asetin kaksi otsikkoa: Tallennetut HOPSit ja Luo HOPS. Tallennetut HOPSit -otsikon alla on listattuna käyttäjän tallentamat opintosuunnitelmat. Opintosuunnitelman nimen viereen tulostetaan viimeisin muokauspäivämäärä, julkisen linkin luontinappi ja opintosuunnitelman poistonappi (kuva 7). Opintosuunnitelman nimeä klikkaamalla käyttäjä pääsee opintosuunnitelman yksityiseen versioon, joka on muuten samanlainen julkisen version kanssa, mutta se sisältää linkin opintosuunnitelman muokkaamiseen. Jos käyttäjä julkaisee opintosuunnitelman, sivun ylälaitaan tulostetaan keltaiseen infolaatikkoon viesti julkaisun onnistumisesta ja tekstilaatikko, joka sisältää julkisen linkin.

The screenshot shows the OKSJ HOPS Kurssit interface. At the top, there is a navigation bar with 'OKSJ HOPS Kurssit' and a user ID '0901455 Kirjaudu ulos'. Below this, the main heading is 'HOPS'. A dropdown menu is open, showing 'Tallennetut HOPSit'. The list contains two entries:

Nimi	Muokattu viimeksi	Julkinen linkki	Poista	Poista linkki
Testi	20:22 11.10.2012	Julkinen linkki	Poista	Poista linkki
Hops	13:46 10.10.2012	Julkinen linkki	Poista	Poista linkki

Below the list is a button for 'Luo HOPS'. At the bottom of the page, there is a footer with the OKSJ logo, a Creative Commons license notice: 'Tämän teoksen käyttöoikeutta koskee Creative Commons Nimeä 3.0 Muokkaamaton -lisenssi.', and 'Powered by Yii Framework'.

Kuva 7. Sovelluksen Tallennetut HOPSit -näkyvä.

Opintosuunnitelman julkistamisen jälkeen opintosuunnitelman nimen viereen tulostetaan linkki julkiseen opintosuunnitelmaan ja nappi julkisen linkin poistamista varten. ”Luo HOPS” -otsikon alla käyttäjä voi luoda oman HOPSinsa antamalla HOPSin nimen sille varattuun tekstikenttään ja valitsemalla alla olevasta listasta haluamansa kurssit (kuva 8). Nimen antaminen ja ainakin yhden kurssin valitseminen on pakollista ja näiden puuttuessa sovellus antaa virheilmoituksen.

HOPS

Tallennetut HOPSit

Luo HOPS

HOPSin nimi:

Nimi ja tunnus	Laajuus	Ajankohta	Valinta
BD2001E - HTML and XHTML	3 op	01.09.2012 - 31.12.2012	<input type="checkbox"/>
BD2003 - Linux	3 op	07.10.2012 - 31.12.2012	<input type="checkbox"/>
BD2203 - PHP-ohjelmointi	3 op	01.03.2012 - 31.05.2012	<input type="checkbox"/>
BD2205 - Kuvankäsittely ja käyttö	3 op	01.03.2012 - 31.05.2012	<input type="checkbox"/>
www-dokumenteissa			
BD2206 - Verkkomultimedia - Flash	4 op	01.03.2012 - 31.05.2012	<input type="checkbox"/>
BD2240 - ASP.NET 3.5 -ohjelmointi	6 op	01.08.2012 - 31.12.2012	<input type="checkbox"/>
BD2302 - Introduction to C++	3 op	01.01.2012 - 31.05.2013	<input type="checkbox"/>
BD2304 - Advanced C++	3 op	01.01.2012 - 31.05.2012	<input type="checkbox"/>
UC1001 - Minä korkeakouluopiskelijana	5 op	01.09.2012 - 31.12.2012	<input checked="" type="checkbox"/>
UC1002 - Tutkin ja Kehitän	5 op	01.01.2012 - 31.12.2012	<input type="checkbox"/>

Tallenna

Kuva 8. Sovelluksen HOPSin luontilomake.

Yksityisen opintosuunnitelman sivulla tunnistetaan käyttäjän käyttäjätunniste ja verrataan sitä opintosuunnitelman luojan käyttäjätunnisteeseen. Jos tunnisteet eivät täsmää, käyttäjälle annetaan viesti, jossa ilmoitetaan, että HOPS on yksityinen ja estetään HOPSin näyttäminen käyttäjälle. Sivulla tulostetaan kaikki käyttäjän HOPSiinsa valitsemat kurssit, niiden ajankohdat ja laajuudet. Kurssia klikkaamalla kurssin lisätiedot liukuvat sen nimen alle. Näitä lisätietoja ovat kurssin kuvaus, ilmoittautumisaika ja vastuuolettajan nimi sekä sähköpostiosoite. Sähköpostiosoitetta klikkaamalla avautuu käyttäjän käyttöjärjestelmän oletussähköpostiohjelman viesti-ikkuna, jonka vastaanottaja ja otsikko -kentät

on valmiiksi täytetty valitun kurssin perusteella. Sivun alaosassa sijaitsee kaksi kaaviota, joista ensimmäisestä käy ilmi mihin kohtaan vuotta HOPSiin valitut kurssit sijoittuvat ja toisesta, kuinka monta tuntia päivässä opiskelija joutuu keskimäärin työskentelemään suorittaakseen valitsemansa kurssit. Julkinen HOPS on muuten samanlainen kuin yksityinen, mutta sivun alalaidassa on kommenttilomake, josta voidaan antaa kommentteja HOPSille (kuva 9). Julkiseen HOPSiin pääsee käsiksi kuka tahansa, jolle julkisen HOPSin verkko-osoite on annettu.

OKSJ HOPS Kurssit
0901455
Kirjaudu ulos

HOPS - Hops

Kurssin nimi ja tunnus	Laajuus	Ajankohta
BD2001E - HTML and XHTML	3 op	01.09.2012 - 31.12.2012
BD2003 - Linux	3 op	07.10.2012 - 31.12.2012
BD2302 - Introduction to C++	3 op	01.01.2012 - 31.05.2013
BD2304 - Advanced C++	3 op	01.01.2012 - 31.05.2012

Yhteensä: 12 op

Opintojaksot

Kurssi	Tammi	Helmi	Maalis	Huhti	Touko	Kesä	Heinä	Elo	Syys	Loka	Marras	Joulu
BD2001E	1	1	1	1	1	1	1	1	1	1	1	1
BD2003	1	1	1	1	1	1	1	1	1	1	1	1
BD2302	1	1	1	1	1	1	1	1	1	1	1	1
BD2304	1	1	1	1	1	1	1	1	1	1	1	1

Kuukausi	Tuntimäärä
Tammi	1
Helmi	1
Maalis	1
Huhti	1
Touko	1
Kesä	0
Heinä	0
Elo	0
Syys	1
Loka	2
Marras	2
Joulu	2

(cc) by Jussi Alanen, Jaakko Vanhala

Kommentit

Viestisi:

Lähetä

Ville Eronen - 26.10.2012 19:18
Täytekommentti on täyttävä

Ville Eronen - 15.10.2012 12:33
Testaan

Kuva 9. Julkinen HOPS.

Kurssien ja kaavioiden välissä sijaitsevasta Muokkaa-linkistä käyttäjä pääsee muokkaamaan juuri sillä hetkellä avoinna olevaa HOPSia. Muokkauslomake on hyvin samankaltainen HOPSin luontilomakkeen kanssa: vain HOPSin nimen kenttä puuttuu. Käyttäjän alun perin HOPSiinsa valitsevat kurssit näkyvät ruudussa valmiiksi valittuina.

5.3 Opettajan näkymä

Rooliltaan opettajaksi määritellyt käyttäjät näkevät sivustolla erilaisen valikon, jossa HOPS- ja Kurssit -otsikoiden tilalla on Ilmoittautumiset-otsikko. Ilmoittautumiset-sivu on jaettu jQueryn haitarivalikolla kolmeen osaan: käsittelyä odottavat ilmoittautumiset, hyväksytyt ilmoittautumiset ja hylätyt ilmoittautumiset. Opettajalle näkyvät vain ne ilmoittautumiset, joissa on ilmoitauduttu kyseisen opettajan kurssille. Käsittelyä odottavat ilmoittautumiset -osiossa näkyvät kaikki kurssit, joille on ilmoittautumisia, mutta joita ei ole hyväksytty tai hylätty. Kurssin nimeä klikkaamalla paljastuvat kaikki ilmoittautumiset kyseiselle kurssille. Ilmoittautumisesta käyvät ilmi opiskelijan opiskelijanumero, nimi sekä kurssi, jolle ilmoitaudutaan. Ilmoittautumisen vieressä on kaksi nappia, joilla opettaja voi hyväksyä tai hylätä ilmoittautumisen. Käsittelemättömien ilmoittautumisten määrä näytetään opettajalle joka sivulla keltaisessa infolaatikossa, jotta ilmoittautumiset eivät jäisi huomioimatta.

Hyväksytyt ilmoittautumiset -osiossa näkyvät kaikki kurssit, joille on hyväksytyjä ilmoittautumisia. Nämä ilmoittautumiset paljastuvat kurssin nimeä klikkaamalla. Ilmoittautumisesta tulostetaan samat tiedot kuin edellisessä osiossa, mutta vieressä on nyt kolme nappia ja alavetovalikko. Poista-napilla poistetaan ilmoittautuminen, Hylkää-napilla hylätään ilmoittautuminen ja Tallenna-napilla tallennetaan alavetovalikosta valittu arvosana (kuva 10). Kurssi merkitään suoritetuksi, kun arvosana on annettu.

OKSJ Ilmoittautumiset
ope
Kirjaudu ulos

▶ Käsittelyä odottavat ilmoittautumiset

▾ Hyväksytyt ilmoittautumiset


[HTML and XHTML](#)
[Linux](#)
[PHP-ohjelmointi](#)
[Kuvankäsittely ja käyttö www-dokumenteissa](#)
[Verkkomultimedia - Flash](#)
[ASP.NET 3.5 -ohjelmointi](#)

Opiskelijanumero	Nimi				
0102234	taavi saarelainen	1			Tallenna Poista Hylkää
1101074	Henri Viitanen	2			Tallenna Poista Hylkää
0902345	jaakko 2	4			Tallenna Poista Hylkää

[Introduction to C++](#)
[Advanced C++](#)

▶ Hylätyt ilmoittautumiset

OKSJ



Tämän teoksen käyttöoikeutta koskee [Creative Commons Nimeä 3.0 Muokkaamaton -lisenssi](#).
 Powered by [Yii Framework](#).

Kuva 10. Opettajan Hyväksytyt ilmoittautumiset -näkökulma.

Hylätyt ilmoittautumiset -osiossa näkyvät kaikki kurssit joille on ilmoittautumisia, jotka opettaja on hylännyt. Kurssin nimeä klikkaamalla paljastuvat kaikki ilmoittautumiset kyseiselle kurssille. Ilmoittautumisesta tulostetaan samat tiedot kuin kahdessa edellisessä osiossa ja vieressä on kaksi nappia: Poista ja Hyväksy. Poista-napilla opettaja voi poistaa ilmoittautumisen ja Hyväksy-napilla siirtää ilmoittautumisen hyväksytyjen joukkoon.

Toteutin ilmoittautumisten hyväksymiset ja hylkäämiset sekä arvosanan tallentamiset AJAXin dynaamisen XMLHttpRequest-objektin avulla, joka mahdollistaa asynkronisen kommunikaation selaimen ja palvelimen välillä. Käytännössä tämä tarkoittaa sitä, että koko sivua ei tarvitse ladata uudestaan arvosanaa tai ilmoittautumisen tilaa tallentaessa, vaan tieto talletetaan tietokantaan ja tallennettu tieto näkyy sivulla heti, ilman sivun uudelleenlataamista. Päädyin tähän ratkaisuun, koska normaalisti sivu päivittyisi ja hyppäisi takaisin haitarikäyttöliit-

tymän ensimmäiseen osioon, minkä takia käyttäjälle aiheutuisi ylimääräisiä hiiren painalluksia. (Refsnes Data 2012.)

Pyrin luomaan sovelluksen käyttöliittymästä mahdollisimman yksiselitteisen, mutta tiettyihin kohtiin laitoin lisätietoa, joka aukeaa uuteen ikkunaan painamalla kysymysmerkkiä sivulla. Toteutin lisäinfon esittämisen jQueryn dialogi-ikkunalla, jonka laitoin liukumaan auki kysymysmerkinapin viereen. Dialogi-ikkuna on siirrettävissä hiirellä minne vain selainikkunassa ja se pysyy auki, vaikka käyttäjä vierittäisi sivua tai klikkaisi haitarivalikossa toisen osion auki. Dialogi-ikkuna sulkeutuu, kun käyttäjä painaa raksia ikkunan yläkulmasta, menee toiselle sivulle tai päivittää nykyisen sivun.

6 Testaus ja jatkokehitystarpeet

Järjestin testaustilaisuuden 12.10., jonka kesto oli neljä tuntia ja tiedotin siitä koulutusohjelmani epävirallisella IRC-keskustelukanavalla. Uuden tunnuksen luoneita käyttäjiä oli testin aikana 13 kappaletta, ja he saivat kokeilla vapaasti opiskelijoiden ominaisuuksia sovelluksessa. Testitilaisuuden aikana ei löytynyt opiskelijoiden puolelta yhtään virhettä tai väärin toimivaa ominaisuutta, mutta opettajien puolelta löytyi ongelmia ilmoittautumisia käsittelevissä AJAX-toiminnoissa. Kun esimerkiksi, opettaja yritti hyväksyä opiskelijan ilmoittautumista, niin joissakin tapauksissa hyväksyminen meni läpi normaalisti, mutta joissakin tietyn käyttäjätunnisteen ylittävissä käyttäjissä hyväksyminen ei toiminut ollenkaan. Tämä ongelma koski kaikkia opettajien ilmoittautumisia käsitteleviä ominaisuuksia. Se johtui virheestä käsiteltävien opiskelijoiden käyttäjätunnisteiden käsitteilyssä, joka johti siihen, että AJAX-toiminnot eivät saaneet oikeita tunnisteita käyttöönsä. Kyseinen ongelma on korjattu, ja sovelluksessa ei ole tällä hetkellä tiedossa yhtään ongelmaa.

Jatkokehitystarpeina sovellukselleni näen ensimmäisenä prioriteettina mobiilikäyttöliittymän suunnittelun ja toteutuksen. Vaikka nykyisessä tilassa sovellukseni toimii mobiilialustoilla, voisi sitä virtaviivaistaa mobiilikäyttöä ajatellen vaikkapa jQuery Mobile -käyttöliittymäjärjestelmää käyttämällä. Myös RGraphil-

la toteutettuja kaavioita tulisi miettiä uudelleen mobiilialustoja varten. Ajanpuutteen vuoksi keskityin pääasiassa opiskelijoille tarkoitettujen toimintojen suunnitteluun ja toteuttamiseen, joten opettajien ja pääkäyttäjän käyttöliittymiä ja toimintoja tulisi kehittää eteenpäin. Myös käyttäjänhallintaan käytetyn YUM-lisäosan hyötyjä ja haittoja tulisi tarkastella ja miettiä, jatketaanko sen käyttöä vai valitaanko jokin toinen lisäosa. Kolmas vaihtoehto on kirjoittaa koko käyttäjänhallinta itse.

7 Loppusanat

Ongelmia opinnäytetyön aikana kohtasin niin teorian, kuin toteutuksenkin puolella. Teoriapuolella minua harmittamaan jäi Zend Frameworkin puutteellinen ja hyvin pintaraapaisulta vaikuttava dokumentaatio, joka haittasi vertailuni toteuttamista ja heikensi sen laatua. Lisäksi ongelmia toi englanninkielisen ammattisanasto, jolle ei aina ollut virallisia suomenkielisiä vastineita. Toteutuksen puolella Yii Frameworkin käyttäjänhallintalisäosa Yii-User-Management aiheutti vaikeuksia: toimimaton käyttäjän rekisteröiminen ja muutenkin virheitä vilisevä lähdekoodi sekä YUMin tietokantataulujen relaatioiden puute hankaloittivat työtäni. Minulla oli myös jo aiemmin mainittuja ongelmia AJAXin kanssa, mutta sain niiden ratkaisemisesta uutta itseluottamusta sovelluksen toteuttamiseen.

Erityisen tyytyväinen olen aikataulutuksen onnistumiseen ja siihen, että työ valmistui vaaditussa ajassa. Sain arvokasta kokemusta eri verkkosovelluskehityksestä, etenkin Yii Frameworkista, sekä verkkosovelluskehityksestä yleensä ja pitkäkestoisesta projektityöstä. Lisäksi opin uusina asioina jQuery-JavaScript-kirjaston käytön sekä dynaamisten ja vuorovaikutteisten toimintojen toteuttamisen AJAXilla. Kaiken kaikkiaan onnistuin opinnäytetyöprosessissa ja sovelluksen toteutuksessa mielestäni hyvin ja kykenin täyttämään asetetut tavoitteet.

Lähteet

- Apache Foundation. 2011. Apache Tutorial: .htaccess files.
<http://httpd.apache.org/docs/2.0/howto/htaccess.html>. 28.10.2012.
- Cake Software Foundation Inc. 2012a. Associations: Linking Models Together.
<http://book.cakephp.org/2.0/en/models/associations-linking-models-together.html>. 9.10.2012.
- Cake Software Foundation Inc. 2012b. CakePHP: the rapid development php framework. <http://cakephp.org/>. 5.6.2012.
- Cake Software Foundation Inc. 2012c. Getting Started.
<http://book.cakephp.org/2.0/en/getting-started.html>. 9.10.2012.
- Cake Software Foundation Inc. 2012d. Helpers.
<http://book.cakephp.org/2.0/en/views/helpers.html>. 5.6.2012.
- Cake Software Foundation Inc. 2012e. Installation.
<http://book.cakephp.org/2.0/en/installation.html>. 9.10.2012.
- Cake Software Foundation Inc. 2012f. Retrieving Your Data.
<http://book.cakephp.org/2.0/en/models/retrieving-your-data.html>. 9.10.2012.
- Cake Software Foundation Inc. 2012g. Saving Your Data.
<http://book.cakephp.org/2.0/en/models/saving-your-data.html>. 9.10.2012.
- Cake Software Foundation Inc. 2012h. Views.
<http://book.cakephp.org/2.0/en/views.html>. 9.10.2012.
- Cake Software Foundation Inc. 2012i. What is CakePHP? Why Use it?.
<http://book.cakephp.org/2.0/en/cakephp-overview/what-is-cakephp-why-use-it.html>. 5.6.2012.
- Custom Application Development Portal. 2010. What Is Rapid Application Development?. <http://www.customapplicationdevelopment.org/rapid-application-development/what-is-rapid-application-development.html>. 5.6.2012.
- DocForge. 2011. Web application framework.
http://docforge.com/wiki/Web_application_framework. 25.5.2012.
- DocForge. 2010. Web application/Security.
http://docforge.com/wiki/Web_application/Security. 18.6.2012.
- EllisLab Inc. 2012a. Active Record Class.
http://codeigniter.com/user_guide/database/active_record.html. 9.10.2012.
- EllisLab Inc. 2012b. CodeIgniter at a Glance.
http://codeigniter.com/user_guide/overview/at_a_glance.html. 4.6.2012.
- EllisLab Inc. 2012c. CodeIgniter Features : CodeIgniter User Guide.
http://codeigniter.com/user_guide/overview/features.html. 4.6.2012.
- EllisLab Inc. 2012d. Company History. <http://ellislab.com/company/history>. 4.6.2012.
- EllisLab Inc. 2012e. Installation Instructions.
http://codeigniter.com/user_guide/installation/index.html. 9.10.2012.
- EllisLab Inc. 2012f. Models : CodeIgniter User Guide.
http://codeigniter.com/user_guide/general/models.html. 4.6.2012.
- EllisLab Inc. 2012g. Open Software License ("OSL") v 3.0.
http://codeigniter.com/nightly_user_guide/license.html. 4.6.2012.

- EllisLab Inc. 2012h. Tutorial – News Section.
http://codeigniter.com/user_guide/tutorial/news_section.html.
 9.10.2012.
- EllisLab Inc. 2012i. Tutorial – Static Pages.
http://codeigniter.com/user_guide/tutorial/static_pages.html.
 9.10.2012.
- jQuery Foundation. 2012a. About. <http://jquery.org/about>. 21.6.2012.
- jQuery Foundation. 2012b. Downloading jQuery.
http://docs.jquery.com/Downloading_jQuery. 21.6.2012.
- jQuery Foundation. 2012c. jQuery. <http://jquery.com/>. 21.6.2012.
- jQuery Foundation. 2012d. jQuery UI. <http://jqueryui.com/>. 25.6.2012.
- jQuery Foundation. 2012e. jQuery UI – Configure Your Download.
<http://jqueryui.com/download/>. 25.6.2012.
- jQuery Foundation. 2012f. jQuery UI – Demos & Documentation.
<http://jqueryui.com/demos/>. 25.6.2012.
- jQuery Foundation. 2012g. License. <http://jquery.org/license>. 21.6.2012.
- jQuery Foundation. 2012h. Tutorials: How jQuery Works.
http://docs.jquery.com/How_jQuery_Works. 22.6.2012.
- Laaksonen. 2011. PHP-ohjelmointi: Osa 13 – Tietoturva.
http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=php_13#sq
 linjektio. 9.10.2012.
- Maxi-Pedia. 2012. Order allow, deny. <http://www.maxi-pedia.com/Order+allow+deny>. 28.10.2012.
- Open Source Initiative. 2009. Open Source Initiative OSI - The MIT License (MIT):Licensing. <http://www.opensource.org/licenses/mit-license.html>. 5.6.2012.
- Petersen. 2012. What is PDO and why should I use it?
<http://www.evanpetersen.com/item/what-is-pdo-and-why-should-i-use-it.html>. 9.10.2012.
- PHP Frameworks. 2012. Top 10 Ranking PHP Frameworks.
<http://www.phpframeworks.com/top-10-php-frameworks/>. 14.6.2012.
- PHP Group. 2012a. Namespaces overview.
<http://www.php.net/manual/en/language.namespaces.rationale.php>.
 28.10.2012.
- PHP Group. 2012b. Safe Password Hashing.
<http://www.php.net/manual/en/faq.passwords.php>. 7.9.2012.
- PHP Group. 2012c. SQL Injection.
<http://php.net/manual/en/security.database.sql-injection.php>.
 9.10.2012.
- Refsnes Data. 2012. AJAX Introduction.
http://www.w3schools.com/ajax/ajax_intro.asp. 9.10.2012.
- RGraph Licensing. 2012a. About RGraph.
<http://www.rgraph.net/about.html>. 27.6.2012.
- RGraph Licensing. 2012b. Documentation and examples.
<http://www.rgraph.net/docs/index.html>. 27.6.2012.
- RGraph Licensing. 2012c. Examples of the supported types of charts.
<http://www.rgraph.net/examples/index.html>. 27.6.2012.
- RGraph Licensing. 2012d. Licensing FAQs and information.
<http://www.rgraph.net/docs/licensing.html>. 27.6.2012.
- RGraph Licensing. 2012e. RGraph: HTML5 & JavaScript charts for your website. <http://www.rgraph.net/>. 27.6.2012.

- Rose India. 2008. MVC Architecture. <http://www.roseindia.net/struts/mvc-architecture.shtml>. 16.6.2012.
- Sensio Labs. 2012a. Creating Pages in Symfony2. http://symfony.com/doc/current/book/page_creation.html. 28.10.2012.
- Sensio Labs. 2012b. Databases and Doctrine. <http://symfony.com/doc/current/book/doctrine.html>. 28.10.2012.
- Sensio Labs. 2012c. HTTP fundamentals. http://symfony.com/doc/current/book/http_fundamentals.html. 7.6.2012.
- Sensio Labs. 2012d. Installing and Configuring Symfony. <http://book.cakephp.org/2.0/en/views.html>. 9.10.2012.
- Sensio Labs. 2012e. Offices. http://sensiolabs.com/en/a_propos/nos_implantations.html. 7.6.2012.
- Sensio Labs. 2012f. Requirements for running Symfony2. <http://book.cakephp.org/2.0/en/views.html>. 9.10.2012.
- Sensio Labs. 2012g. Symfony at a Glance. <http://symfony.com/symfony-at-a-glance>. 7.6.2012.
- Sensio Labs. 2012h. Twig for Template Designers. <http://twig.sensiolabs.org/doc/templates.html#functions>. 28.10.2012.
- Sensio Labs. 2012i. Twig – The flexible, fast, and secure template engine for PHP. <http://twig.sensiolabs.org/>. 28.10.2012.
- Sensio Labs. 2012j. Why should I use a Framework? <http://symfony.com/why-use-a-framework>. 7.6.2012.
- World Wide Web Consortium. 2012. The canvas element. <http://www.w3.org/TR/html5/the-canvas-element.html#the-canvas-element>. 27.6.2012.
- Yii Software LLC. 2012a. About Yii. <http://www.yiiframework.com/about/>. 25.5.2012.
- Yii Software LLC. 2012b. Features. <http://www.yiiframework.com/features/>. 28.5.2012.
- Yii Software LLC. 2012c. Getting Started: Creating Your First Yii Application. <http://www.yiiframework.com/doc/guide/1.1/en/quickstart.first-app>. 5.6.2012.
- Yii Software LLC. 2012d. Installation. <http://www.yiiframework.com/doc/guide/1.1/en/quickstart.installation>. 9.10.2012.
- Yii Software LLC. 2012e. Query Builder. <http://www.yiiframework.com/doc/guide/1.1/en/database.query-builder>. 9.10.2012.
- Yii Software LLC. 2012f. Yii-user-management. <http://www.yiiframework.com/extension/yii-user-management/>. 9.10.2012.
- Zend Technologies Ltd. 2012a. Database and models. <http://framework.zend.com/manual/2.0/en/user-guide/database-and-models.html>. 28.10.2012.
- Zend Technologies Ltd. 2012b. Get the Skeleton Application. <http://framework.zend.com/downloads/skeleton-app>. 28.10.2012.
- Zend Technologies Ltd. 2012c. Getting Started: A skeleton application. <http://framework.zend.com/manual/2.0/en/user-guide/skeleton-application.html>. 28.10.2012.

- Zend Technologies Ltd. 2012d. Getting Started with Zend Framework 2.
<http://framework.zend.com/manual/2.0/en/user-guide/overview.html>.
28.10.2012.
- Zend Technologies Ltd. 2012e. Latest Releases.
<http://framework.zend.com/downloads/latest>. 28.10.2012.
- Zend Technologies Ltd. 2012f. Modules.
<http://framework.zend.com/manual/2.0/en/user-guide/modules.html>.
28.10.2012.
- Zend Technologies Ltd. 2012g. Routing and Controllers.
<http://framework.zend.com/manual/2.0/en/user-guide/routing-and-controllers.html>. 28.10.2012.
- Zend Technologies Ltd. 2012h. Zend\Db\Sql.
<http://framework.zend.com/manual/2.0/en/modules/zend.db.sql.html>.
28.10.2012.
- Zend Technologies Ltd. 2012i. Zend Framework: About.
<http://framework.zend.com/about/overview>. 29.5.2012.
- Zend Technologies Ltd. 2012j. Zend Framework Components.
<http://framework.zend.com/about/components>. 19.6.2012.
- Zend Technologies Ltd. 2012k. Zend Framework Downloads.
<http://framework.zend.com/download/current/>. 30.5.2012.
- Zend Technologies Ltd. 2012l. Zend Server Datasheet.
<http://static.zend.com/topics/Zend-Server-DS-1011-EN-c.pdf>.
30.5.2012.
- Zend Technologies Ltd. 2012m. Zend Server Editions.
<http://www.zend.com/en/products/server/editions>. 30.5.2012.
- Zend Technologies Ltd. 2012n. Zend Server - PHP Web Application Server.
<http://www.zend.com/en/products/server/>. 30.5.2012.

Verkkosovelluksen käyttötapauskaavio ja kirjallinen kuvaus



Tunniste	Kirjautuminen
Toimija	Opettaja, opiskelija tai pääkäyttäjä
Tavoite	Toimija kirjautuu sisään
Laukaisija	Toimijan halu kirjautua sisään
Esiehdot	Toimija on rekisteröitynyt palveluun
Suoritus	<ol style="list-style-type: none"> 1. Toimija klikkaa Kirjaudu sisään -linkkiä verkkosivun oikeasta yläreunasta 2. Toimija kirjoittaa käyttäjätunnuksensa ja salasanasensa avautuvaan dialogi-ikkunaan 3. Toimija painaa Kirjaudu-nappia tai näppäimistöä enteriä
Poikkeukset	Toimija ei täytä kaikkia kenttiä. Toimija antaa väärän käyttäjätunnuksen tai salasanan. Toimija ei ole rekisteröitynyt
Lopputulos	Toimija on kirjautunut sisään
Huomioitavaa	Kaikki kentät on täytettävä

Tunniste	Ulos kirjautuminen
Toimija	Opettaja, opiskelija tai pääkäyttäjä
Tavoite	Toimija kirjautuu ulos järjestelmästä
Laukaisija	Toimija haluaa kirjautua ulos
Esiehdot	Toimija on kirjautunut sisään
Suoritus	<ol style="list-style-type: none"> 1. Toimija klikkaa sivun oikeasta yläreunasta Kirjaudu ulos -linkkiä
Poikkeukset	Toimija ei ole kirjautunut sisään
Lopputulos	Toimija on kirjautunut ulos
Huomioitavaa	-

Tunniste	Rekisteröinti
Toimija	Opettaja tai opiskelija
Tavoite	Toimija luo itselleen käyttäjätunnuksen
Laukaisija	Toimijan halu päästä käyttämään verkkosovellusta
Esiehdot	-
Suoritus	<ol style="list-style-type: none"> 1. Toimija klikkaa sivun oikeassa yläreunassa olevaa Rekisteröidy-

	linkkiä 2. Toimija täyttää kaikki kentät 3. Toimija painaa Tallenna-nappia tai enteriä näppäimistöltä
Poikkeukset	Toimija on kirjautunut jo sisään toisella tunnuksella. Toimija ei täytä kaikkia kenttiä. Toimija antaa liian lyhyen salasanan.
Lopputulos	Toimija on luonut itselleen käyttäjätunnuksen
Huomioitavaa	Salasanan on oltava viisi merkkiä pitkä

Tunniste	Kurssien selaaminen
Toimija	Opiskelija
Tavoite	Opiskelija selaa kursseja
Laukaisija	Opiskelijan halu selata kursseja
Esiehdot	Opiskelijalla on käyttäjätunnus, opiskelija on kirjautunut sisään
Suoritus	1. Opiskelija painaa sivun valikosta Kurssit-linkkiä 2. Opiskelija valitsee haitarivalikosta haluamansa kategorian
Poikkeukset	Opiskelija ei ole kirjautunut sisään
Lopputulos	Opiskelija on selannut kursseja
Huomioitavaa	Kategorioita ovat Kaikki kurssit, Ilmoittautumiset ja Suoritetut kurssit

Tunniste	Kurssille ilmoittautuminen
Toimija	Opiskelija
Tavoite	Opiskelija ilmoittautuu kurssille
Laukaisija	Opiskelijan halu ilmoittautua kurssille
Esiehdot	Opiskelijalla on käyttäjätunnus, opiskelija on kirjautunut sisään
Suoritus	1. Opiskelija painaa Kurssit-linkkiä 2. Opiskelija painaa Ilmoittaudu-nappia haluamansa kurssin kohdalla
Poikkeukset	Opiskelija ei ole kirjautunut sisään
Lopputulos	Opiskelija on ilmoittautunut kurssille

Huomioitavaa	Jos opiskelija on jo ilmoittautunut kursseille, Ilmoittaudu-nappia ei enää ole näkyvissä kyseiselle kurssille
---------------------	---

Tunniste	HOPSin luominen
Toimija	Opiskelija
Tavoite	Opiskelija luo HOPSin
Laukaisija	Opiskelijan halu luoda HOPS
Esiehdot	Opiskelijalla on käyttäjätunnus ja hän on kirjautunut sisään
Suoritus	<ol style="list-style-type: none"> 1. Opiskelija painaa HOPS-linkkiä 2. Opiskelija painaa haitarivalikosta Luo HOPS -otsikkoa 3. Opiskelija antaa HOPSille nimen tekstikenttään 4. Opiskelija valitsee haluamansa kurssit HOPSiin 5. Opiskelija painaa Tallenna-nappia
Poikkeukset	Opiskelija ei ole kirjautunut sisään, HOPSilla ei voi olla samaa nimeä kuin aiemmin tallennetulla HOPSilla
Lopputulos	Opiskelija on luonut HOPSin
Huomioitavaa	HOPSin nimi-kenttä on pakollinen ja opiskelijan on valittava ainakin yksi kurssi HOPSiinsa. Jos opiskelijalla on hyväksytyjä ilmoittautumisia kursseille ne näkyvät kirjoitussuojattuina ja valittuina valintaruutuina

Tunniste	HOPSin muokkaaminen
Toimija	Opiskelija
Tavoite	Opiskelija muokkaa HOPSia
Laukaisija	Opiskelijan halu muokata HOPSia
Esiehdot	Opiskelijalla on käyttäjätunnus, opiskelija on kirjautunut sisään ja opiskelijalla on tallennettu HOPS
Suoritus	<ol style="list-style-type: none"> 1. Opiskelija painaa HOPS-linkkiä 2. Opiskelija valitsee Tallennetut

	<p>HOPSit -otsikon alta haluamansa HOPSin klikkaamalla sen nimeä</p> <ol style="list-style-type: none"> 3. Opiskelija painaa Muokkaa-linkkiä 4. Opiskelija tekee haluamansa muutokset kurssivalintoihin painamalla valintaruutuja kurssien kohdalta 5. Opiskelija painaa Tallenna-nappia
Poikkeukset	Opiskelija ei ole kirjautunut sisään, Opiskelijalla ei ole luotua HOPSia
Lopputulos	Opiskelija on muokannut HOPSia
Huomioitavaa	Opiskelijan on valittava ainakin yksi kurssi HOPSiinsa. Jos opiskelijalla on hyväksytyjä ilmoittautumisia kursseille ne näkyvät kirjoitussuojattuina ja valittuina valintaruutuina

Tunniste	HOPSin julkaiseminen
Toimija	Opiskelija
Tavoite	Opiskelija julkaisee HOPSin
Laukaisija	Opiskelijan halu julkaista HOPS
Esiehdot	Opiskelijalla on käyttäjätunnus, opiskelija on kirjautunut sisään, opiskelijalla on tallennettu HOPS
Suoritus	<ol style="list-style-type: none"> 1. Opiskelija painaa HOPS-linkkiä 2. Opiskelija painaa Julkaise-nappia haluamansa HOPSin kohdalta 3. Opiskelija ottaa HOPSin julkisen linkin talteen ilmestyvästä tekstikentästä tai Julkinen linkki-linkistä 4. Opiskelija antaa linkin HOPSiinsa haluamilleen henkilöille
Poikkeukset	Opiskelija ei ole kirjautunut sisään, opiskelijalla ei ole tallennettua HOPSia, HOPS on jo julkinen
Lopputulos	Opiskelija on julkaissut HOPSin
Huomioitavaa	-

Tunniste	Roolien hallinta
Toimija	Pääkäyttäjä

Tavoite	Pääkäyttäjä hallitsee rooleja
Laukaisija	Halu muuttaa opiskelija opettajan rooliin tai toisinpäin
Esiehdot	Pääkäyttäjä on kirjautunut sisään
Suoritus	<ol style="list-style-type: none"> 1. Pääkäyttäjä painaa etusivulta Hallitse rooleja -linkkiä 2. Pääkäyttäjä valitsee halutun käyttäjän kohdalta alavetovalikosta halutun roolin 3. Pääkäyttäjä painaa halutun käyttäjän kohdalla tallenna
Poikkeukset	Pääkäyttäjä ei ole kirjautunut sisään
Lopputulos	Pääkäyttäjä on hallinnut rooleja
Huomioitavaa	-

Tunniste	Kurssien lisääminen
Toimija	Pääkäyttäjä
Tavoite	Pääkäyttäjä lisää kursseja
Laukaisija	Halu lisätä kursseja
Esiehdot	Pääkäyttäjä on kirjautunut sisään
Suoritus	<ol style="list-style-type: none"> 1. Pääkäyttäjä painaa etusivulta Kurssien lisääminen -linkkiä 2. Pääkäyttäjä täyttää vaaditut kentät 3. Pääkäyttäjä painaa Tallenna-nappia
Poikkeukset	Pääkäyttäjä ei ole kirjautunut sisään
Lopputulos	Pääkäyttäjä on lisännyt kurssin
Huomioitavaa	Tunnus, laajuus, kuvaus, vastuopettaja ja nimi -kentät ovat vaadittuja

Tunniste	HOPSin kommentointi
Toimija	Opiskelija tai Opettaja
Tavoite	Toimija kommentoi HOPSia
Laukaisija	Toimijan halu kommentoida hopsia
Esiehdot	Toimija on kirjautunut sisään, toimijalla on julkinen linkki kommentoitavaan HOPSiin
Suoritus	<ol style="list-style-type: none"> 1. Toimija menee omaan julkiseen HOPSiinsa painamalla HOPSin

	<p>kohdalta Julkinen linkki -linkkiä tai jonkun muun HOPSiin saamallaan julkisella linkillä</p> <p>2. Toimija kirjoittaa viestikenttään viestin</p> <p>3. Toimija painaa Lähetä-nappia</p>
Poikkeukset	Toimija ei ole kirjautunut sisään, toimijalla on toimimaton HOPSiin linkki
Lopputulos	Toimija on kommentoinut HOPSiä
Huomioitavaa	Viesti-kenttä on pakko täyttää

Tunniste	Ilmoittautumisten selaaminen
Toimija	Opettaja
Tavoite	Opettaja selaa ilmoittautumisia
Laukaisija	Opettajan halu selata ilmoittautumisia
Esiehdot	Opettaja on kirjautunut sisään
Suoritus	<ol style="list-style-type: none"> Opettaja painaa ilmoittautumiset-linkkiä valikosta Opettaja valitsee haitarivalikosta haluamansa kategorian
Poikkeukset	Opettaja ei ole kirjautunut sisään
Lopputulos	Opettaja on selannut ilmoittautumisia
Huomioitavaa	-

Tunniste	Ilmoittautumisen hylkääminen
Toimija	Opettaja
Tavoite	Opettaja hylkää ilmoittautumisen
Laukaisija	Opettajan halu hylätä ilmoittautuminen
Esiehdot	Opettaja on kirjautunut sisään
Suoritus	<ol style="list-style-type: none"> Opettaja painaa ilmoittautumiset-linkkiä valikosta Opettaja valitsee haitarivalikosta kategorian Opettaja klikkaa haluamansa kurssin nimeä Opettaja painaa Hylkää-nappia haluamansa opiskelijan kohdalta
Poikkeukset	Opettaja ei ole kirjautunut sisään, järjestelmässä ei ole yhtään käsittelemättä tai hyväksytyä ilmoittautumista
Lopputulos	Opettaja on hylännyt ilmoittautumisen

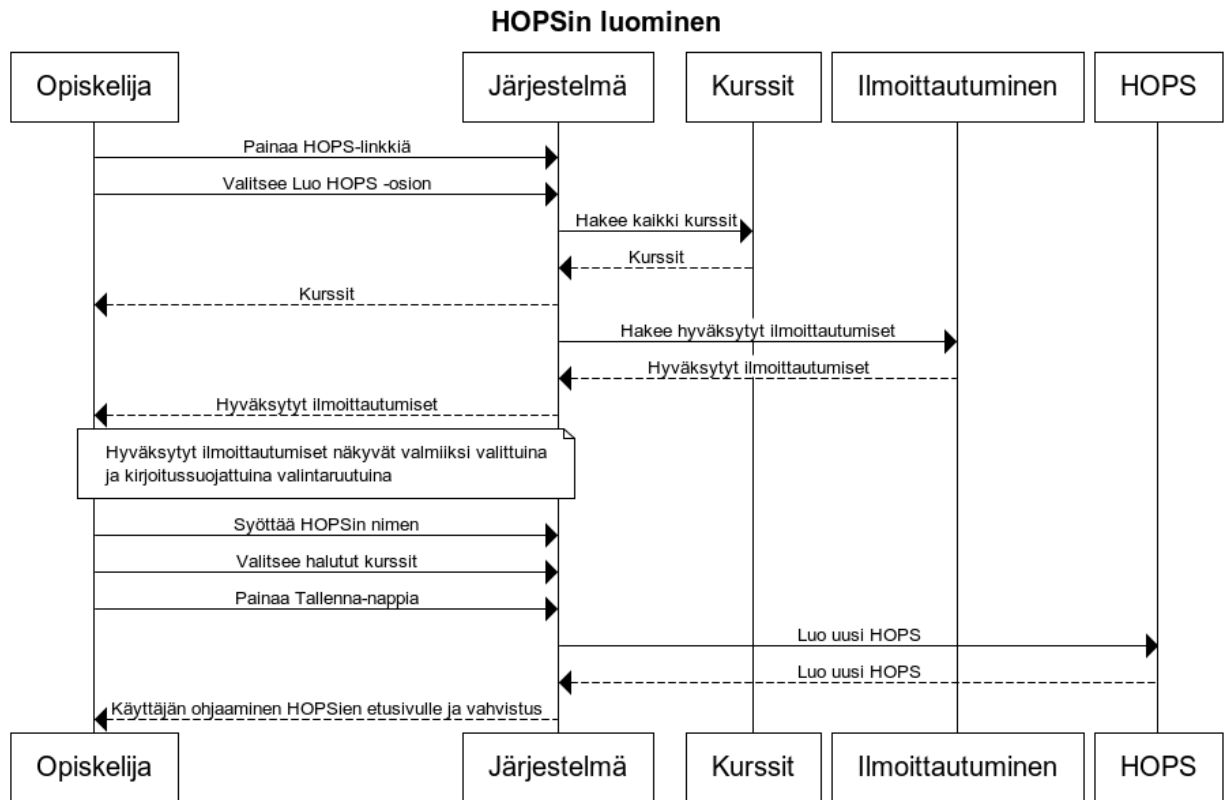
Huomioitavaa	-
---------------------	---

Tunniste	Ilmoittautumisen hyväksyminen
Toimija	Opettaja
Tavoite	Opettaja hyväksyy ilmoittautumisen
Laukaisija	Opettajan halu hyväksyä ilmoittautuminen
Esiehdot	Opettaja on kirjautunut sisään
Suoritus	<ol style="list-style-type: none"> 1. Opettaja painaa ilmoittautumisetlinkkiä valikosta 2. Opettaja valitsee haitarivalikosta kategorian 3. Opettaja klikkaa haluamansa kurssin nimeä 4. Opettaja painaa Hyväksy-nappia haluamansa opiskelijan kohdalta
Poikkeukset	Opettaja ei ole kirjautunut sisään, järjestelmässä ei ole yhtään käsittelemättä tai hylättyä ilmoittautumista
Lopputulos	Opettaja on hyväksynyt ilmoittautumisen
Huomioitavaa	-

Tunniste	Ilmoittautumisen poistaminen
Toimija	Opettaja
Tavoite	Opettaja poistaa ilmoittautumisen
Laukaisija	Opettajan halua poistaa ilmoittautuminen
Esiehdot	Opettaja on kirjautunut sisään
Suoritus	<ol style="list-style-type: none"> 1. Opettaja painaa ilmoittautumisetlinkkiä valikosta 2. Opettaja valitsee haitarivalikosta Hyväksytyt kurssit tai Hylätyt kurssit -kategorian 3. Opettaja klikkaa haluamansa kurssin nimeä 4. Opettaja painaa Poista-nappia haluamansa opiskelijan kohdalta
Poikkeukset	Opettaja ei ole kirjautunut sisään, järjestelmässä ei ole yhtään hyväksyttyä tai hylättyä ilmoittautumista
Lopputulos	Opettaja on hylännyt ilmoittautumisen
Huomioitavaa	-

Tunniste	Arvosanan antaminen
Toimija	Opettaja
Tavoite	Opettaja antaa arvosanan
Laukaisija	Opettajan halu antaa arvosana
Esiehdot	Opettaja on kirjautunut sisään
Suoritus	<ol style="list-style-type: none">1. Opettaja painaa ilmoittautumisetlinkkiä valikosta2. Opettaja valitsee haitarivalikosta Hyväksytyt kurssit -kategorian3. Opettaja klikkaa haluamansa kurssin nimeä4. Opettaja valitsee alasvetovalikosta haluamansa arvosanan haluamansa opiskelijan kohdalta
Poikkeukset	Opettaja ei ole kirjautunut järjestelmään, järjestelmässä ei ole yhtään hyväksyttyä ilmoittautumista
Lopputulos	Opettaja on antanut arvosanan opiskelijalle
Huomioitavaa	-

Sekvenssikaavioita verkkosovelluksen toiminnoista



Kurssille ilmoittautuminen

